DESIGN AND IMPLEMENTATION OF INTELLIGENT CONTROLLERS ON NONLINEAR SYSTEMS

A Thesis Submitted
In Partial Fulfilment of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY

by

ABHISHEK CHAUDHARY
(2k19/PhDEE/504)

Under the Supervision of
Prof. Bharat Bhushan

Department of Electrical Engineering

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi – 110042, India

February, 2025

CANDIDATE’S DECLARATION

I Abhishek Chaudhary, hereby certify that the work which is being presented in the thesis entitled “**Design and Implementation of Intelligent Controllers on Nonlinear Systems”** in partial fulfilment of the requirements for the award of the degree of Doctor of Philosophy, submitted in the Department of Electrical Engineering, Delhi Technological University is an authentic record of my own work carried out during the period from 2020 to 2024 under the supervision of Prof. Bharat Bhushan.

The matter presented in the thesis has not been submitted by me for the award of any other degree of this or any other Institute.

Abhishek Chaudhary
(2K19/PhD/EE/504)

This is to certify that the student has incorporated all the corrections suggested by the examiners in the thesis and the statement made by the candidate is correct to the best of our knowledge.

Signature of Supervisor

CERTIFICATE BY THE SUPERVISOR

Certified that Abhishek Chaudhary (2k19/PhDEE/504) has carried out his research work presented in this thesis entitled “**Design and Implementation of Intelligent Controllers on Nonlinear Systems**” for the award of **Doctor of Philosophy** from Department of Electrical Engineering, Delhi Technological University, Delhi under my supervision. The thesis embodies results of original work and studies are carried out by the student himself and the contents of the thesis do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Prof. Bharat Bhushan
Department of Electrical Engineering
Delhi Technological University
Delhi-110042, India

Date:

Design and Implementation of Intelligent Controllers on Nonlinear Systems

Abhishek Chaudhary

ABSTRACT

The operation of robotic systems to execute complex tasks within dynamic environments represents a critical and challenging area of modern control systems engineering. As robotics, artificial intelligence and autonomous systems continue to advance, their applications across various sectors of society multiply, offering a wide array of opportunities and introducing significant risks. These opportunities and risks are particularly pronounced in areas such as path tracking, speed control and balance control—factors that are deeply influenced by the inherent complexity and unpredictability of real-world environments. To ensure that robotic systems can operate autonomously without succumbing to collisions, disturbances, or operational failures, it becomes imperative to monitor and optimize the parameters governing both mechanical and electronic components, thereby enhancing system reliability and performance.

This research delves into the integration of intelligent control approaches and sophisticated optimization algorithms aimed at achieving advanced path tracking, robust balance control, continuous system monitoring and overall robustness, all while relying solely on on-board computing resources. The focus of this study is on the development and implementation of control strategies that are both adaptive and resilient, capable of handling the uncertainties and nonlinearities that typify real-world environments. These strategies are specifically tailored for two-degree-of-freedom (2-DOF) operations within benchmark systems such as the ball balancer and helicopter systems, which serve as practical examples of the challenges faced by modern control systems.

The design of these controllers begins with the application of feedback linearization techniques in conjunction with classical control methodologies. This foundational approach is subsequently enhanced by incorporating intelligent controllers designed to improve robustness and adaptability in the face of unpredictable disturbances. The research provides a comprehensive overview of the dynamic equations that govern these systems, offering essential insights for control system designers who seek to understand the physical behaviour underlying the mathematical models. This theoretical foundation is followed by a detailed exposition of the mathematical techniques employed to augment the basic control laws, emphasizing the robust methodologies that make the system more resilient to external and internal perturbations.

One of the primary challenges identified in current control practices is the limited capacity of conventional controllers to handle nonlinearities and uncertainties inherent in dynamic environments. To address this, the research proposes an intelligent control approach for both the ball balancer and helicopter systems. This approach utilizes a fuzzy-proportional-integral-derivative (Fuzzy-PID) controller, which is adept at managing the position control of the ball balancer and the trajectory tracking of the helicopter. The fuzzy logic component of the controller enhances its ability to deal with uncertainties, while the PID elements ensure precise and responsive control. To further optimize the performance of this controller, the research introduces a novel teaching-learning-based optimization (TLBO) algorithm. This algorithm improves upon existing methods by addressing transparency issues in the literature, thereby providing a more reliable and effective optimization process.

Moreover, the study develops a hybrid optimization algorithm (HGPCTLBO) designed to optimize the parameters of a hybrid controller under conditions of random uncertainty. This hybrid controller combines classical control techniques with intelligent methodologies, resulting in a system that is both robust and flexible. The optimization process is tailored to the unique demands of both the helicopter and ball balancer systems, ensuring that the control parameters are finely tuned to achieve optimal performance under varying operational conditions.

The optimization of the constraint parameters for both the classical controller and the hybrid classical-intelligent controller is carried out using a novel hybrid Giza pyramid construction teaching-learning-based optimization (GPC-TLBO) algorithm. This algorithm is specifically designed to handle the complex constraints and performance criteria associated with the two benchmark systems. The effectiveness of the developed control techniques is rigorously validated through comprehensive simulation studies, as well as real-time analysis conducted on the actual systems. These validation processes demonstrate the superior performance and reliability of the proposed methodologies, highlighting their potential for widespread application in advanced robotic systems.

In conclusion, this research makes significant contributions to the field of control systems by offering novel optimization approaches to handling the complexities and uncertainties of dynamic environments. The integration of intelligent control techniques with advanced optimization algorithms represents a promising direction for future developments in robotics and autonomous systems, paving the way for more resilient and capable robotic systems that can operate effectively in a wide range of challenging environments.

ACKNOWLEDGEMENT

First and foremost, I wish to express my profound gratitude to God Almighty for His endless blessings and guidance throughout this journey. It is His grace that has provided me with the strength and wisdom necessary to achieve this milestone and for that, I am forever thankful.

I extend my sincere thanks to the reviewers who dedicated their time and expertise to thoroughly evaluate my work. Their invaluable feedback and constructive criticism have been instrumental in refining and completing this thesis. Their commitment to academic excellence has greatly contributed to the quality of this research and I am deeply appreciative of their efforts.

I am deeply grateful to my family for their unwavering love, support, and belief in me. To my parents, I owe everything to the values you instilled in me from a young age. Your sacrifices, guidance and encouragement have been the bedrock upon which my achievements are built. Your enduring support has been my greatest source of strength and I am forever indebted to you.

To my younger brother, your belief in my abilities, coupled with your encouragement, has driven me to push through even the most challenging moments. I am grateful for the bond we share and for being the pillar of support in my life throughout this journey of ups and down.

To my beloved wife, your love, patience and understanding have been my greatest source of comfort and motivation. Your unwavering belief in me, even when the road was difficult, has given me the courage to pursue my dreams. You have been my rock, providing me with the strength to continue and I am profoundly grateful for your sacrifices, especially during the times when I was fully immersed in my work. This achievement is as much yours as it is mine and I am blessed to have you by my side.

To my daughter, I dedicate this work to you as well. It was your time that I utilized to write this thesis, but watching you grow up in my arms has been the greatest joy and blessing I could ask for. You are my inspiration and I hope that one day you will find the same passion and determination in your pursuits as I have in mine.

I am also immensely thankful to my friends, whose technical expertise, moral support, and encouragement have been invaluable throughout this journey. Their willingness to engage in discussions, provide feedback and offer encouragement during difficult times has been crucial to the completion of this work. Special thanks to my colleagues, who are more family than friends now, who stood by me, cheering me on and providing a listening ear when I needed it most.

This thesis is a testament to the collective support and guidance of these remarkable individuals. Without their contributions, this journey would not have been possible. I am deeply grateful to each and every one of you.

Thank you.

Abhishek Chaudhary

TABLE OF CONTENT

CANDIDATE’S DECLARATION ____ i

CERTIFICATE BY THE SUPERVISOR ____ ii

ABSTRACT ____ iii

ACKNOWLEDGEMENT ____ vi

TABLE OF CONTENT ____ viii

LIST OF TABLES ____ xi

LIST OF FIGURES ____ xii

Chapter 1. Introduction

- 1.1 Overview ____ 1
- 1.2 Background ____ 1
  - 1.2.1 Optimization algorithms ____ 1
  - 1.2.2 Nonlinear systems ____ 2
    - 1.2.2.a 2DoF ball balancer system ____ 3
    - 1.2.2.b 2DoF helicopter system ____ 4
  - 1.2.3 Nonlinear control ____ 4
  - 1.2.4 Underactuated nonlinear systems ____ 5
- 1.3 Existing challenges ____ 9
  - 1.3.1 Challenges in optimization ____ 7
  - 1.3.2 Challenges in nonlinear control system ____ 7
- 1.4 Motivation ____ 8
- 1.5 Objectives ____ 10
- 1.6 Methodology ____ 11
- 1.7 Outline ____ 12

Chapter 2. Literature Review

- 2.1 Optimization algorithms ____ 13
  - 2.1.1 Heuristic optimization techniques ____ 15
  - 2.1.2 Metaheuristic optimization techniques ____ 15
- 2.2 Classical controller ____ 21
- 2.3 Computational techniques for classical controller tuning ____ 23
- 2.4 Control mechanisms used for 2DoF ball balancer system ____ 28
- 2.5 Control mechanisms used for 2DoF helicopter system ____ 29
- 2.6 Identified research gaps ____ 32

Chapter 3. Two Degree of Freedom (2DoF) Systems

- 3.1 Overview ____ 33
- 3.2 Modelling of 2DoF ball balancer System ____ 33
  - 3.2.1 Transfer function ____ 35
  - 3.2.2 Mathematical equations for plate movement ____ 35

3.2.3 Servo angle calculations 36
3.3 Problem formulation for 2DoF ball balancer system 37
3.4 Modelling of 2DoF helicopter System 38
3.4.1 Inter-relation between pitch and yaw axis 38
3.5 Problem formulation for 2DoF helicopter model 40
3.5.1 Calculation of ABCD parameters 42
3.6 Conclusion 43

Chapter 4. Optimized Classical Control of 2DoF Systems

4.1 Classical PID controller for 2DoF systems 44
4.1.1 Defining cost functions 47
4.2 TLBO algorithm based control mechanism 48
4.2.1 TLBO algorithm 48
4.2.2 TLBO algorithm based control of 2DoF ball balancer 51
4.2.2.a Time response analysis 53
4.2.2.b Cost function analysis 54
4.2.3 TLBO algorithm based control of 2DoF helicopter system 55
4.2.3.a Time response analysis 55
4.2.3.b Cost function analysis 57
4.3 GPC algorithm based control mechanism 58
4.3.1 GPC algorithm 58
4.3.2 GPC algorithm based control of 2DoF ball balancer system 61
4.3.2.a Time response analysis 63
4.3.2.b Cost function analysis 63
4.3.3 GPC algorithm based control of 2DoF helicopter system 64
4.3.3.a Time response analysis 65
4.3.3.b Cost function analysis 67
4.4 Conclusion 67

Chapter 5. Improved TLBO Algorithm Based Control of 2DoF Systems

5.1 Improved TLBO (iTLBO) 68
5.1.1 Updating process for teaching factor 68
5.1.2 Updated data-driven solution 69
5.2 iTLBO on traditional benchmark functions 71
5.3 CEC functions used for validating iTLBO 74
5.4 Describing function and control scheme implementation 76
5.5 iTLBO based control of 2DoF ball balancer system 78
5.5.1 Numerical Simulation 79
5.5.2 Real-time Experiment 80
5.6 iTLBO based control of 2DoF helicopter system 81
5.6.1 Convergence analysis 81
5.6.2 Numerical Simulation 82
5.6.3 Real-time Experiment 83
5.7 Conclusion 85

Chapter 6. Hybrid Intelligent-Classical Control for 2DoF Systems

6.1 Limitations of classical control strategy and need of hybrid controller 86
6.2 Optimized FPID control mechanism 87

6.3 Hybrid FPID based control of 2DoF ball balancer system 89
    6.3.1 Simulated results and comparison 89
    6.3.2 Real-time results 91
6.4 Hybrid FPID based control of 2DoF helicopter system 93
    6.4.1 Simulated results and comparison 94
    6.4.2 Real-time results 95
6.5 Conclusion 97

Chapter 7. HGPCTLBO: A Hybrid Algorithm, Based Optimized Control of 2DoF Systems

7.1 Idea of HGPCTLBO 99
    7.1.1 Command phase 100
    7.1.2 Co-operate phase 101
7.2 Algorithm flow of improved GPC: HGPCTLBO 101
7.3 HGPCTLBO-PID based control of 2DoF ball balancer system 102
    7.3.1 Numerical analysis with objective functions 103
    7.3.2 Real-time results with stability analysis 107
7.4 HGPCTLBO-PID based control of 2DoF helicopter system 108
    7.4.1 Numerical analysis with objective functions 109
    7.4.2 Real-time results with stability analysis 113
7.5 Conclusion 115

Chapter 8. Conclusion and Future Scope of Work

8.1 Introduction 116
8.2 Contributions of the work 116
8.3 Suggestions for future scope of work 118
8.4 Societal impact of the work 119

APPENDIX 120

REFERENCES 128

CURRICULUM VITAE 149

LIST OF PUBLICATIONS 150

LIST OF TABLES

Table 4.1: Comparison of time response of servo angle results 53
Table 4.2: Objective function values 54
Table 4.3: Comparison of simulation results obtained after time response analysis of pitch angle 57
Table 4.4: Comparison of simulation results obtained after time response analysis of yaw angle 57
Table 4.5: Values of different error derivatives after applying TLBO-PID based control mechanism 58
Table 4.6: Comparison of time response servo angle results 63
Table 4.7: Objective function values 63
Table 4.8: Comparison of simulation results obtained after time response analysis of pitch angle 65
Table 4.9: Comparison of simulation results obtained after time response analysis of yaw angle 65
Table 4.10: Values of different error derivatives after applying TLBO-PID based control mechanism 67
Table 5.1: Optimization results on benchmark function 1 72
Table 5.2: Optimization results on benchmark function 2 72
Table 5.3: Optimization results on benchmark function 1 73
Table 5.4: Optimization results on benchmark function 4 74
Table 5.5: Mean solution comparison of iTLBO algorithm with optimization algorithms available in literature after implementing on CEC benchmark functions 76
Table 5.6: Time response parameters and RMSE values 80
Table 5.7: Time response parameters obtained for pitch axis for various controllers during simulation 83
Table 5.8: Time response parameters obtained for yaw axis for various controllers during simulation 83
Table 6.1: Rule set used for fuzzy logic control action 88
Table 6.2: Time response parameters 90
Table 6.3: Root Mean Square Error 90
Table 6.4: Integral time absolute error values 91
Table 6.5: Time response parameters of graphs obtained in Fig. 6.11 & Fig. 6.12 97
Table 7.1: Time domain specifications comparison of all optimization algorithms used above 105
Table 7.2: Objective function values for experiments conducted 105
Table 7.3: Response observed from yaw axis for various optimization methods 111
Table 7.4: Response observed from pitch axis for various optimization methods 112
Table 7.5: Minimized error values after applying various optimization algorithms 112

LIST OF FIGURES

Fig. 1.1 Operational procedure of an optimization algorithm 2
Fig. 1.2 2-degree of freedom (2DoF) Ball Balancer System 4
Fig. 1.3 2-degree of freedom (2DoF) Helicopter System 4
Fig. 2.1 General steps followed by optimization techniques 16
Fig. 3.1 Real time working laboratory setup of 2DoF ball balancer system 34
Fig. 3.2 Working Structure between different components of 2DoF ball balancer system 34
Fig. 3.3 Schematic diagram of a two degree of freedom ball balancer system. 34
Fig. 3.4 2DOF ball balancer transfer function for (a) X-axis servo angle (b) Y-axis servo angle and (c) one dimensional ball balancer (1DBB) open loop transfer function 35
Fig. 3.5 Free body diagram of 2DOF ball balancer system 36
Fig. 3.6 Block diagram with closed loop control scheme of a two degree of freedom ball balancer system 37
Fig. 3.7 Free body diagram of a 2DoF helicopter model 38
Fig. 3.8 Cross-coupling effect of pitch rotor and yaw rotor in 2DoF helicopter model 38
Fig. 3.9 Real time experimental (a) Body and (b) setup of 2 DOF helicopter system under external disturbances 39
Fig. 4.1 Close loop control system with PID constraints 44
Fig. 4.2 Block diagram of PID control scheme of 2DoF ball balancer system 45
Fig. 4.3 Block diagram of cross-coupled controller for 2DoF helicopter system 46
Fig. 4.4 Flow chart for TLBO algorithm 50
Fig. 4.5 Block diagram representing TLBO based tuning of PID controller of a 2DOF ball balancer system 51
Fig. 4.6 Simulated response of default PD controller showing output variations of servo load angle and ball position of a 2DoF ball balancer system 52
Fig. 4.7 Simulated response of TLBO-PID controller showing output variations of servo load angle and ball position of a 2DoF ball balancer system 52
Fig. 4.8 Bar chart showing the values of different objective functions after using classical approach and TLBO optimization algorithm 54
Fig. 4.9 Block diagram representing TLBO based tuning of PID controller of a 2DOF helicopter. 55
Fig. 4.10 Simulated response of LQR based controller showing output variations of yaw angle trajectory, pitch angle trajectory and voltage 56

Fig. 4.11	Simulated response of TLBO-PID control mechanism showing output of yaw angle trajectory, pitch angle trajectory and voltage variations	56
Fig. 4.12	Stone block position on the ramp with all the forces	59
Fig. 4.13	Detailed flow-chart for GPC	60
Fig. 4.14	Block diagram representing GPC based tuning of PID controller of a 2DOF ball balancer system	61
Fig. 4.15	Simulated response of PD controller showing output variations of servo load angle and ball position	62
Fig. 4.16	Simulated response of GPC-PID control mechanism showing output of servo load angle and ball position	62
Fig. 4.17	Comparison of ISE, IAE & ITAE response of GPC algorithm with ZN tuned PID	64
Fig. 4.18	Block diagram representing TLBO based tuning of PID controller of a 2DOF helicopter	64
Fig. 4.19	Simulated response of LQR based controller showing output variations of yaw angle trajectory, pitch angle trajectory and voltage	66
Fig. 4.20	Simulated response of GPC-PID control mechanism showing output of yaw angle trajectory, pitch angle trajectory and voltage variations	66
Fig. 5.1	Detailed flow chart illustrating the iTLBO algorithm	70
Fig. 5.2	Complete block diagram to represent the angle and position control of a 2DOF ball balancer system using iTLBO optimization method while tuning a PID controller	78
Fig. 5.3	Simulation response of (a) x-axis of ball balancer system after using the PID controller (b) servo angle variations (c) voltage experienced by servo motor.	79
Fig. 5.4	Simulation response of (a) x-axis of ball balancer system after using iTLBO optimized PID controller approach (b) servo angle variations (c) voltage experienced by servo motor	79
Fig. 5.5	Real time responses of (a) ball position on x-axis (b) servo angle (c) input voltage experienced by servo motor of ball balancer system using PID controller	80
Fig. 5.6	Improved real time responses of (a) ball position on x-axis (b) servo angle (c) input voltage experienced by servo motor of ball balancer system using proposed iTLBO-PID controller	80
Fig. 5.7	Block diagram representing iTLBO based tuning of PID controller of a 2DOF helicopter	81
Fig.5.8	Absolute Error vs Time graph obtained by (a) LQR controller [322] (b) TLBO algorithm [63] (c) iTLBO algorithm	82
Fig. 5.9	Pitch angle trajectory with the square reference signal during external disturbance (a) default Quanser [322] LQR controller (b) TLBO algorithm (c) proposed iTLBO algorithm	84
Fig. 5.10	Yaw angle trajectory with the square reference signal during external disturbance using (a) default Quanser [322] LQR controller (b) TLBO algorithm (c) iTLBO algorithm.	84

Fig. 6.1 Hybrid Fuzzy+Proportional+Integral+Derivative (FPID) control mechanism 87
Fig. 6.2 Fuzzy logic membership function 88
Fig. 6.3 Control mechanism to handle the balancing of 2DoF ball balancer system 89
Fig. 6.4 Simulation response of variations experienced by x-axis and servo angle deflection of ball balancer system using the PID controller 90
Fig. 6.5 Simulation response of variations experienced by x-axis and servo angle deflection of ball balancer system using proposed approach 90
Fig. 6.6 Integral time absolute error observed while using the PID controller and proposed hybrid FPID-iTLBO controller 91
Fig. 6.7 Response of PID controller during real time experiment showing (a) ball position (b) servo angle (c) input voltage signal received by servo motor 92
Fig. 6.8 Response of FPID-iTLBO control mechanism during real time experiment showing (a) ball position (b) servo angle (c) input voltage signal received by servo motor 92
Fig. 6.9 Controller mechanism used to handle pitch and yaw angles of 2DoF helicopter system 93
Fig. 6.10 Error convergence graph of (a) pitch angle with LQR controller (b) pitch angle with FPID-iTLBO controller (c) yaw angle on LQR controller [1] (d) yaw angle with FPID-iTLBO controller 95
Fig. 6.11 Flight trajectory of pitch angle under the external disturbance using (a) LQR controller and (b) FPID-iTLBO control scheme 96
Fig. 6.12 Flight trajectory of yaw angle under the external disturbance using (a) LQR controller and (b) FPID-iTLBO control scheme 96
Fig. 7.1 Flow chart of HGPCTLBO algorithm 102
Fig. 7.2 Block diagram showing the tuning mechanism used for constraints of PID controller 103
Fig. 7.3 The ball position w.r.t. reference step signal for classical PID controller 104
Fig. 7.4 The ball position w.r.t. reference step signal for GPC-PID controller 104
Fig. 7.5 The ball position w.r.t. reference step signal for TLBO-PID controller 104
Fig. 7.6 The ball position w.r.t. reference step signal for proposed HGPCTLBO-PID controller 104
Fig. 7.7 The simulation outcomes of the ball balancer model using the traditional PID controller illustrate (from left to right): (a) the ball's tracking response on the square plate, (b) the variations in servo angle during operation, and (c) the feedback voltage supplied to the servo motor during closed-loop feedback operation 106
Fig. 7.8 The simulation outcomes of the ball balancer model using the HGPCTLBO-PID controller depict (from left to right): (a) the ball's tracking response on the square plate, (b) the variations in servo angle during operation, and (c) the feedback voltage supplied to the servo motor during closed-loop feedback operation 106

Fig. 7.9 Comparison of ISE, IAE & ITAE response of different objective functions while using classical PID, GPC-PID, TLBO-PID & HGPCTLBO-PID controllers 106
Fig. 7.10 Real time responses of (a) ball position (b) servo angle (c) voltage experienced by servo motor of ball balancer system using PID controller 107
Fig. 7.11 Real time responses of (a) ball position (b) servo angle and (c) voltage experienced by servo motor of ball balancer model using HGPCTLBO –PID controller 107
Fig. 7.12 Control mechanism used for handling the pitch and yaw angles of 2DoF helicopter system 108
Fig. 7.13 Fuzzy-HGPCTLBO-PID controller mechanism used to handle flight angles of 2DoF helicopter 109
Fig. 7.14 Yaw angle vs time response when using TLBO optimization algorithm 110
Fig. 7.15 Yaw angle vs time response when using GPC optimization algorithm 110
Fig. 7.16 Yaw angle vs time response when using HGPCTLBO optimization algorithm 110
Fig. 7.17 Pitch angle vs time response when using TLBO optimization algorithm 111
Fig. 7.18 Pitch angle vs time response when using GPC optimization algorithm 111
Fig. 7.19 Pitch angle vs time response when using HGPCTLBO optimization algorithm 111
Fig. 7.20 Integral of Square error obtained after using GPC, TLBO & HGPCTLBO algorithms 113
Fig. 7.21 Integral of absolute error obtained after using GPC, TLBO & HGPCTLBO algorithms 113
Fig. 7.22 Integral of time square error obtained after using GPC, TLBO & HGPCTLBO algorithms 113
Fig. 7.23 The evaluation of the pitch axis trajectory of a real-time helicopter model under the influence of external disturbances caused by high-speed fans subsequent to the optimization algorithms including (a) GPC (b) TLBO and (c) the hybridized-GPC-TLBO 114
Fig. 7.24 The evaluation of the yaw axis trajectory of a real-time helicopter model under the influence of external disturbances caused by high-speed fans subsequent to the optimization algorithms including (a) GPC (b) TLBO and (c) the hybridized-GPC-TLBO 114

Chapter 1. Introduction

1.1 Overview

The challenge of approximating underactuated systems through the development of automatic decision-making and nonlinear control methods is a prevalent issue in various problems [1]. Technological advancements have had a significant impact on the adoption of innovative analytical approaches to address nonlinear challenges [2]. In many instances involving nonlinear control systems, quantifying the state may prove to be a complex task, posing challenges in addressing intricate control engineering problems. Another challenge lies in the insufficient comprehension of critical variables, as the states of the system can profoundly influence the nature of the control design phase, ultimately determining the potential for outstanding performance. This complexity has prompted researchers to explore different approaches to address this challenge. The diversity and intricacy of these systems have driven researchers to examine the performance of various controllers, primarily focused on achieving self-balancing control and steady-state operation. Traditionally, the rigorous analysis and control design for these systems have been motivated by a range of scientific, industrial, and military applications. Simultaneously, the theoretically challenging nature of analysing the behaviour of nonlinear dynamical systems has captivated mathematicians, leading to a comprehensive study of control systems. Comprehending the control requirements of a system demands a profound understanding of the system itself; however, the complexities of nonlinearities frequently make it extremely challenging to develop controls that ensure the best performance of a system [3]. Consequently, the collaborative efforts of engineers and scientists have given rise to the development of theories such as linear control, optimal control, adaptive control, and nonlinear control. The parameter tuning of nonlinear systems, a challenging feature of control theory, stands out as one of its most formidable aspects. Optimization strategies have been employed in the past to surmount this challenge, proving particularly advantageous when dealing with complicated systems.

Over the recent decades, the domains of aerospace and robotics have consistently stood out as compelling sources of inspiration driving the thorough examination and control of nonlinear systems. The progress made by researchers in these fields has reciprocally influenced and elevated each other. Given these advances, this chapter outlines the different surfaces of nonlinear control theory pertaining to nonlinear systems.

1.2 Background

1.2.1 Optimization algorithms

Optimization involves the examination of selecting the most favourable option from a limited array of choices. It is often considered a crucial quantitative element within a dynamic system where decisions are necessary to improve one or more assessments under specific conditions [4]. Each optimization problem includes decision variables, specific objective (fitness) functions and a set of constraints. An analysis of optimization algorithms in the literature indicates the absence of a systematic classification. Figure 1.1 illustrates the operation of an optimization algorithm.

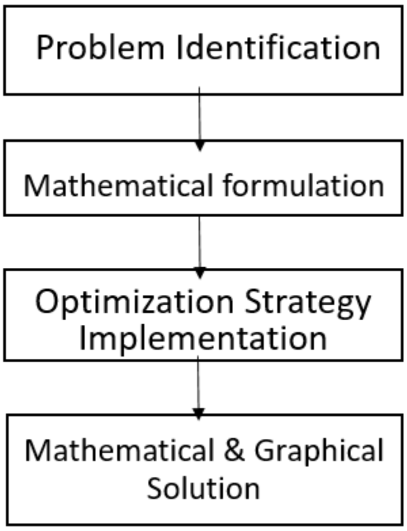

Problem Identification
Mathematical formulation
Optimization Strategy Implementation
Mathematical & Graphical Solution

Fig. 1.1 Operational procedure of an optimization algorithm

In metaheuristic algorithms, the prefix "meta-" conveys the idea of going "beyond" or operating at a "higher level," surpassing the capabilities of basic heuristics. Metaheuristics prove to be effective in generating acceptable solutions for complex problems through trial and error within a reasonable timeframe. While the terms 'heuristics' and ‘metaheuristics' are sometimes used interchangeably [5], a recent trend distinguishes any stochastic algorithms involving randomization and global exploration as metaheuristics, however there is no certainty in discovering the best solution, and the workings of an algorithm remain unpredictable. The objective is to devise an efficient and practical algorithm that operates effectively in the majority of cases and produces high-quality results [6].

Metaheuristic algorithms are characterized by two main features: intensification and diversification. The intensification phase, or exploitation, seeks and identifies the best candidates or solutions based on the current best approaches. The diversification phase, or exploration, ensures efficient traversal of the search space. Maintaining a delicate balance between these components significantly impacts the overall efficiency of an algorithm. Insufficient exploration coupled with excessive exploitation may lead the system to be trapped in a local optimum, making it extremely challenging, if not impossible, to find the global optimum. The "No Free Lunch theorems" in optimization assert that if algorithm A outperforms algorithm B for specific optimization functions, then B will outperform A for all other functions. This implies that when averaged over all potential function spaces, algorithms A and B perform equally well. In essence, there is no universally superior algorithm [7].

1.2.2 Nonlinear systems

To describe the characteristics of intricate nonlinear systems, the transition from linear systems to nonlinear systems in control methodologies has been illuminated. Time-invariant linear control systems have been well-established, allowing for the use of state or output feedback to control, observe, stabilize, and track the system. However, introducing additional constraints or specifications to the system description can render the control design quite challenging. Coping with mildly nonlinear systems commonly involves frequency domain analysis or an approach to input-output stability. The concept of feedback connectivity between a linear system and a nonlinearity has been expanded in the literature to encompass feedback between an LTI system and a gain-limited uncertainty. Consequently, integral quadratic constraints and robust stability theory were formulated [8]. While effective in the presence of linear, uncertain linear and slightly nonlinear systems, these techniques are not applicable to fully nonlinear systems. Fully nonlinear systems exhibit nonlinear temporal evolution and lack any fundamental linear components. To be more specific, a modification referred to as a saturation-type recurrent neural network, utilizing sigmoidal nonlinearity, exhibits no fundamental distinctions from a control system of Linear Time-Invariant (LTI) type or a somewhat nonlinear system in terms of controllability and observability.

Despite the nonlinear rule governing the system's time-evolution, its linearity as an output does not simplify the system analysis. Requirements for controllability (specifically in the discrete-time context) and observability of dynamic neural networks, serving as examples of highly nonlinear systems, were proposed in [9]. The previous work employed a somewhat intricate time-domain analysis technique. While frequency domain analysis exclusively addresses systems with linear state evolution over time, nonlinear systems, in a comprehensive local theory encompassing disturbance decoupling, tracking, stability, observability, and controllability [10]. Differential geometry and Lie theory, widely adopted in the literature, were the primary tools used to address these control challenges. Although these techniques proved effective for local analyses, they often fail in global studies involving affine control-based nonlinear systems. The theory of input-to-state stability [11] encompasses both absolute stability and robust stability theories for highly nonlinear systems. Control Lyapunov functions (CLFs) are pivotal instruments in this theory for robustness analysis against disturbances. Furthermore, after applying a specific change of coordinate variables to the nonlinear dynamics, the converted system or one or more of its components may be a nonlinear system that is not affine in terms of control.

1.2.2.a 2DoF ball balancer system

The 2 DOF Ball Balancer utilizes two Rotary Servo Base Units as its foundation. Through this experiment, researchers have the opportunity to apply the principles learned in the one-dimensional Ball and Beam experiment to the X-Y planar scenario. A complex electromechanical system with under-actuation, multivariate elements, and nonlinearity can be effectively represented using ball balancer systems. This system involves the simultaneous operation of two servo motors to control the position of a ball in a 2-Degree of Freedom (DOF) Ball Balancer. Being inherently nonlinear, this system serves as a challenging benchmark in various control applications and approaches. Users have the opportunity to experiment with different control methods to guide the ball to a specific location on a table. The system features a horizontal plate with slants in both directions, enabling the ball to roll freely across the surface. This setup dynamically demonstrates nonlinear kinematics and control theory. Due to its inherent characteristics of nonlinearity, instability and under-actuation, the ball balancer system is frequently employed to assess and evaluate control algorithms and technologies (Fig. 1.2).

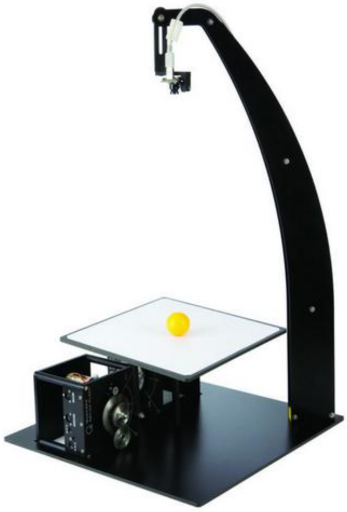


Fig. 1.2 Two-degree of freedom (2DoF) Ball Balancer System

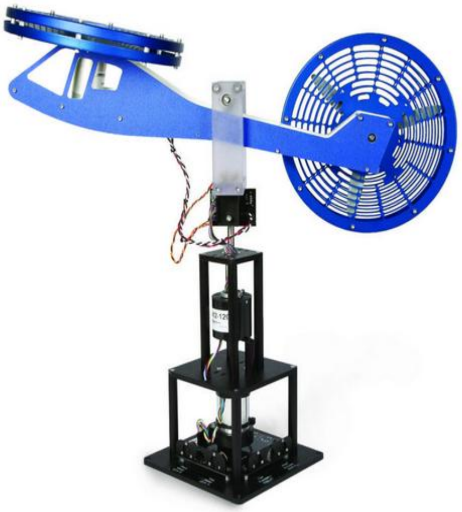


Fig. 1.3 Two-degree of freedom (2DoF) Helicopter System

1.2.2.b 2DoF helicopter system

A 2 Degree of Freedom (2 DoF) helicopter system is a setup in which the helicopter's motion is defined by two separate and manageable degrees of freedom. This usually includes the capability to regulate the helicopter's movement along two distinct axes or directions. In this context, "DoF" represents the count of independent parameters essential for describing the system's motion. In the instance of a 2 DoF helicopter system, there exist two such parameters that can be adjusted or controlled to impact the helicopter's behaviour. These systems involve the simultaneous operation of two servo motors to control the balancing of a 2-Degree of Freedom (DOF) helicopter. Being inherently nonlinear, this system serves as a challenging benchmark in various control applications and approaches. Users have the opportunity to experiment with different control methods to balance the body at different angles in free space. This setup dynamically demonstrates nonlinear kinematics and control theory. Due to its inherent characteristics of nonlinearity, instability, and under-actuation, the helicopter system is frequently employed to assess and evaluate control algorithms and technologies (Fig. 1.3)

1.2.3 Nonlinear control

Practical systems inherently exhibit nonlinearity, particularly over a broader spectrum of operating conditions, even though certain systems are expected to exhibit linear behaviour near a specific operational point and at slower speeds under specific conditions. Nonlinear models are employed to accurately represent a diverse array of physical phenomena, including but not limited to gravitational and electrostatic attraction, Coulomb friction, V-I characteristics of electrical systems, and the drag on a moving vehicle [12]. The popularity of nonlinear control is on the rise due to advancements in linear systems, a deeper exploration of nonlinearities, the need to address parametric uncertainty, and the flexibility it offers in system architecture. Approaches that account for dynamic forces such as sensory and Coriolis forces, which vary in speed, surpass basic techniques [13].

Consequently, linear control principles impose limitations on the speed at which a desired accuracy can be achieved. Utilizing a straightforward nonlinear controller allows for adaptation to nonlinear forces, enabling high speeds across an extensive range of motion. The nature of real-world systems defies linear approximation due to inherent nonlinearities such as hysteresis, dead zones, saturation, and backlash. Nonlinear approaches excel in compensating for these nonlinearities, offering unparalleled efficacy. Model parameter uncertainty is a common characteristic in real-world systems due to abrupt or gradual shifts in parameter values. The resilience or adaptability of a nonlinear controller proves instrumental in managing the implications of model uncertainty [14]. This thesis is thoroughly considering the two nonlinear systems i.e. ball balancer system and helicopter system to implement state-of-art control strategies for achieving higher stability in these underactuated systems.

1.2.4 Underactuated nonlinear systems

Underactuated mechanical systems are characterized by having fewer control inputs than configuration variables, and they find applications across various domains such as Robotics, Aerospace Systems, Marine Systems, Flexible Systems, Mobile Systems, and Locomotive Systems. The property of being "under actuated" in these systems can be attributed to four main reasons: i) inherent dynamics of the system (e.g., aircraft, spacecraft, helicopters, underwater vehicles, locomotive systems without wheels), ii) intentional design for cost reduction or practical purposes (e.g., satellites with two thrusters and flexible-link robots), iii) actuator failure (e.g., in a surface vessel or aircraft), and iv) deliberately imposed to create complex low-order nonlinear systems for gaining insights into the control of high-order underactuated systems (e.g., the Beam-and-Ball system, Acrobot, Cart-Pole system, TORA system , Rotating Pendulum). The predominant strategies for controlling underactuated systems, specifically those resembling inverted-pendulum dynamics, typically involve initiating a swing-up motion from the pendulum's downward position and subsequently transitioning to a balancing controller.

This balancing controller is commonly crafted using techniques such as linearization or gain scheduling to ensure equilibrium [15]. Illustrations of these approaches encompass swing-up control utilizing energy-based methods for the cart-pole system, triple-link inverted pendulum, and rotating pendulum [16]. For the Acrobot, the balancing controller employing spline functions is outlined in [17]. The beam-and-ball system, due to its intricate nature, has attracted considerable attention from researchers with varied interests. Diverse methodologies have been utilized, including approximate feedback linearization by Hauser et al. [18], small nested saturations for the stabilization of feedforward cascade nonlinear systems by Teel, and stabilization through output feedback, as detailed in [19]. The vertical take-off and landing (VTOL) aircraft serves as an exemplary underactuated system extensively employed as a test-bed for various trajectory tracking methods and configuration stabilization techniques. This encompasses trajectory tracking for slightly non-minimum phase systems [20] and control methods based on hybrid/switching approaches [21].

Exponential stabilization of underactuated examples of underwater vehicles and surface vessels was accomplished in [22] and [23] through appropriate coordinate transformations and the analysis of a time-varying linear system. A similar outcome for the attitude control of an underactuated spacecraft is detailed in [24]. The role of second-order non-holonomic constraints in necessitating the use of discontinuous stabilizing feedback laws for underactuated system stabilization is discussed in [25], primarily based on the well-known condition on the stability of nonlinear systems using time-invariant continuously differentiable state feedbacks [26]. Moreover, the accessibility of classes of underactuated mechanical systems has recently been addressed in [27]. This is grounded in a framework applied to the analysis of the controllability of non-holonomic systems [28] and a controllability theorem [29]. An example illustrating discontinuous stabilizing feedback for a system with an internal unactuated degree of freedom is presented in [30]. Adaptive control [31] and sliding mode control techniques [32] have also found application in underactuated mechanical systems, even though for limited scenarios.

Flexible-link robots, being a significant class of underactuated systems suitable for space applications due to their lightweight and rapid execution of commands, are discussed. The Euler-Bernoulli model for a flexible arm is portrayed as an infinite-dimensional system [33]. A truncated modal analysis is employed to derive a finite-dimensional state-space model for flexible robots [33], [34]. Trajectory tracking for flexible robots is intricate, and standard measurements like the angle of rotation or the position of the tip often result in poor performance and non-minimum phase zero dynamics. In [35], a proposed non-collocated minimum-phase output based on an analysis of the initial infinite-dimensional model is presented, followed by the design of a finite-order compensator for trajectory tracking.

A nonlinear non-collocated minimum-phase output for a flexible one-link robot arm is obtained from finite-order Euler-Lagrangian equations of the system. The method of controlled Lagrangians, preserving the Lagrangian structure of a mechanical system through the application of a control input, has been employed for local stabilization of the cart-pole system and the rotating pendulum to an equilibrium manifold [36]. However, this method has yet to successfully stabilize the rotating pendulum or more general underactuated systems to an equilibrium point. Hybrid and switching-based control approaches are gaining traction in the control of underactuated mechanical systems [36] and bipedal locomotion of walking robots. To conclude, aside from linearization-based techniques, the control of underactuated mechanical systems has predominantly focused on stabilization. This encompasses specific examples of cascade nonlinear systems and the utilization of energy-based methods combined with supervisory-based switching control. The current state of research in underactuated system control is far from achieving the goal of developing control design methods effective for broad classes of high-order underactuated systems in robotics and aerospace applications.

1.3 Existing challenges

In this subsection, the major challenges faced by researchers during the implementation of control strategies on nonlinear systems are discussed. These challenges are further addressed in subsequent chapters of this thesis.

1.3.1 Challenges in optimization

Optimization algorithms face numerous significant challenges, mirroring the intricate and varied nature of real-world issues. Some of the primary obstacles with validated reasoning are mentioned here and handled further in this thesis:

Enhancing Efficiency and Scalability:

Challenge: The efficiency and scalability of optimization algorithms are pivotal, particularly when addressing extensive problems or intricate systems. Reasoning: With the expansion in size and complexity of optimization problems, algorithms must adeptly manage an increasing number of variables and constraints while upholding efficiency.

Achieving Convergence to Global Optima:

Challenge: Persistently ensuring convergence to a global optimum instead of a local one poses a challenge. Reasoning: Many optimization algorithms tend to converge towards local optima, necessitating the development of methods that consistently identify the global optimum.

Enhancing Robustness in Noisy Environments:

Challenge: Numerous optimization problems in the real world are characterized by noisy or uncertain environments. Reasoning: Algorithms need to exhibit robustness to handle variations, uncertainties, and noise in both the objective function and constraints.

The optimization process involves seeking the most effective solution to a problem. Consequently, a major challenge for metaheuristics lies in effectively addressing this task. Despite numerous suggestions for metaheuristics, only a select few consistently achieve the necessary success rate. Population-based metaheuristics are commonly favoured for their ability to adapt to large-scale optimization challenges. As mentioned earlier, metaheuristics are algorithms tailored to specific problems. Therefore, the key question revolves around determining the optimal algorithm parameter specifications based on the characteristics and size of the problem's search space. Moreover, the selection of an appropriate metaheuristic algorithm is a complex endeavour. Recent advancements aim to enhance and broaden the applicability of metaheuristic methods in order to overcome these challenges.

1.3.2 Challenges in nonlinear control system

Nonlinear control systems present numerous difficulties owing to the inherent complexity of their behaviour. This tends to the existence of many challenges while dealing with the stability of nonlinear systems. Some of the primary challenges associated with managing nonlinear control systems with validated reasoning are mentioned here:

Complex Dynamics:

Challenge: Nonlinear systems frequently display complex and intricate dynamics. Reasoning: The complexity introduced by nonlinearities makes comprehending and forecasting the behaviour of these systems challenging, leading to complexities in both system analysis and design.

Uncertainty and Parameter Variability:

Challenge: Nonlinear control systems are vulnerable to uncertainties and variations in system parameters. Reasoning: Variations in parameters or uncertainties within the system model can induce unpredictable behaviour, necessitating the development of robust control strategies.

Stability and Bifurcations:

Challenge: Ensuring stability in nonlinear control systems proves to be an intricate task. Reasoning: Nonlinear systems may undergo bifurcations, resulting in sudden changes in system behaviour. Analysing and controlling these bifurcations represent challenging aspects of nonlinear control.

Controller Design:

Challenge: Formulating effective controllers for nonlinear systems is a challenging endeavour. Reasoning: The nonlinear nature of these systems often demands sophisticated control strategies, making the design of controllers that ensure stability and desired performance a complex process.

Real-Time Implementation:

Challenge: Executing nonlinear control algorithms in real-time can be formidable. Reasoning: The computational intricacies associated with certain nonlinear control strategies may present challenges for real-time implementation, particularly in applications with stringent time constraints.

Addressing these challenges requires a fusion of adequate mathematical modelling, control theory, and computational techniques to devise potent control strategies for nonlinear systems. Ongoing research endeavours contribute to the continual advancement of nonlinear control methodologies.

1.4Motivation

Based on the aforementioned observations, it is recognized that, beyond methods based on linearization, the control of underactuated mechanical systems has predominantly focused on stabilization. This involves specific instances of cascade nonlinear systems and the utilization of energy-based techniques in conjunction with supervisory-based switching control. The portrayal of highly nonlinear systems can be depicted as a progression from linear systems to nonlinear systems in control systems.

Questions pertaining to controllability, observability, stabilization, and tracking for this system using state or output feedback have been comprehensively understood for an extended period.

Addressing these issues is not as straightforward as the control problems for the original linear system. Minor deviations from the standard problem of stabilizing a linear time-invariant (LTI) control system, coupled with additional constraints, render the system complex. Moreover, the control design and analysis for systems with various nonlinearities have given rise to absolute stability theory. The current state of research in nonlinear systems control falls significantly short of our objective to develop effective control design methods applicable to diverse robotic applications. These motivation to current work can be outlined as follows:

- In control applications, the controller's reaction to real time like situations in the system's behaviour is not deeply discussed.

- Traditional control methods lack the capability to adapt and learn the system's behaviour across various operations, leading to frequent adjustments whenever there is a change in system operation, which demands a certain optimization technique.

- Optimization algorithms are more or less dependent on parametric uncertainties. For the development of more effective algorithm in the long run, each step should be based on thoughtful planning, innovative features, and intelligent approaches, which demands improvement in already available optimization algorithms.

- Researchers achieved a stable and satisfactory nonlinear system response by employing either a standard PID controller or it’s more flexible variants, which needs the initial constraint values to be inserted manually every time. This needs to be automated in addition with a more sophisticated optimization algorithms.

- As Einstein famously stated, "Everything should be made as simple as possible, but not simpler." In practical applications, a robust algorithm with a simpler architecture is favoured for ease of implementation while maintaining efficiency for real-world applications. Which means, for practical applications we should be focused on traditional controllers with much enhanced performance while cascading with intelligent techniques.

The limitations present in traditional control approaches, particularly when dealing with nonlinear and underactuated systems, served as the incentive for formulating various control strategies in this thesis. To elaborate briefly, challenges faced during the execution of famous algorithms like Teaching Learning Based Optimization (TLBO) algorithm and Giza Pyramid Construction (GPC) algorithm is addressed in detail. These algorithms aim to address uncertainties, enhance system sensitivity to various disturbances, and achieve effective control actions. This approach was developed in an offline mode using fuzzy controller, offering advantages in terms of accurate condition monitoring and better stability to nonlinear systems. All these control techniques are driven by the motivation to overcome the limitations identified in the earlier conventional control approaches.

1.5Objectives

This research is centred on the control of two-degree-of-freedom (2DoF) benchmark mechanical systems, driven by the widespread applications of underactuated systems and the intriguing theoretical challenges they pose. The research aims to achieve control and stability in the benchmark systems based on the following defined objectives:

- Develop mathematical models for the two-degree-of-freedom (2DoF) ball balancer system and 2DoF helicopter system.

- Design and implement classical and intelligent control techniques such as proportional-derivative, proportional-integral-derivative, linear quadratic control, fuzzy inference system and cascaded control for the 2DoF ball balancer system and 2DoF helicopter system using MATLAB/Simulink.

- To implement real-time control of the 2DoF ball balancer system and 2DoF helicopter system for the defined intelligent and classical controllers.

- Use the optimization algorithm to achieve more efficient and flexible response of 2DoF ball balancer system and 2DoF helicopter system while optimizing the classical and intelligent controllers.

- To improve the functioning of Teaching Learning Based Optimization (TLBO) algorithm by introducing a new improvement factor to the algorithm and validate the algorithm using CEC benchmark functions. Use the improved algorithm to control the 2DoF ball balancer and helicopter system.

- Develop a hybrid metaheuristic algorithm by improving Giza Pyramid Construction (GPC) algorithm while introducing Co-operate phase and Command phase to the algorithm. Then validate the developed algorithm using benchmark functions. Use the developed algorithm to control the 2DoF ball balancer system and 2DoF helicopter system.

- Formulate hybrid controllers such as fuzzy-proportional integral derivative controllers, and tune the constrained parameters of this cascaded controller using the developed hybrid metaheuristic algorithm to develop control mechanism for 2DoF ball balancer system and 2DoF helicopter system.

1.6 Methodology

This research makes a significant contribution by introducing optimization techniques to address the positioning, tracking, and balancing control challenges encountered in benchmark robotic systems. The approach involves the initial development of mathematical models for both systems. Additionally, an analysis of conventional control methods related to system modelling, operation and control is conducted to identify problems and formulate novel control strategies. Real time working results and Simulation graphs are carried out on 2DoF ball balancer system and 2DoF helicopter system using MATLAB 2015, which is powered by Intel(R) Core (TM) i5-4210U CPU processor @ 2.4 GHz and 1.70 GHz with 4 GB of RAM. The study proposes following methodology to tackle the control issues arising from the nonlinear, underactuated, and uncertain dynamics of the system:

- Optimization algorithms are used to find the best settings for control parameters to reduce various error functions. Optimization helps tackle down the problems like multiple local minimums and complicated interactions among two-degree-of-freedom (2DoF) nonlinear system variables. The flexibility of optimization algorithms, like TLBO and GPC, allows for real-time adjustments to adapt to changes or uncertainties in the system. Their ability to handle constraints ensures that the solutions are practical and feasible.

- The need to improve optimization algorithms comes from the growing complexity and variety of real-world problems. As technological, scientific and industrial challenges become more complex, traditional optimization methods struggle to efficiently navigate these intricate and dynamic solution spaces. Enhancements in optimization algorithms allow for the development and integration of innovative approaches to tackle specific issues, adapt to changing conditions, and boost overall efficiency. By adding an improvement factor to the TLBO algorithm and introducing cooperate phase and command phase to the GPC algorithm, these algorithms become better at handling multiple objective functions.

- These improvements are validated using CEC functions and optimizing PID controllers for robotic systems. This ensures that the enhanced optimization algorithms remain effective and versatile in solving practical problems, such as controlling a 2 DoF helicopter system and a 2 DoF ball balancer system.

- Using an improved optimization algorithm to tune the constrained parameters of a classical PID controller involves calculating initial parameters manually, which can be time-consuming and complex. This issue is addressed by using a fuzzy PID cascaded controller, which helps generate the initial constrained parameters for the classical controller.

- The importance of using a fuzzy PID controller in nonlinear robotic systems lies in its ability to handle the complex and unpredictable dynamics of these systems. Unlike linear systems, nonlinear robotic systems exhibit diverse and nonlinear behaviours that are difficult to model accurately. The fuzzy PID controller effectively manages these uncertainties by using a fuzzy logic-based approach, allowing for a more adaptable and responsive control strategy. This controller can dynamically adjust its parameters based on real-time feedback and environmental changes, enhancing the system's resilience and overall performance.

1.7 Outline of this thesis

Chapter 1 (this chapter) is about introduction of controlling the nonlinear and underactuated systems using classical and intelligent methods, with some of the basics of nonlinear control theory.

Chapter 2 provides a detailed literature on various optimization strategies, classical controller, fuzzy systems, and other recent control techniques developed to handle the uncertainties in nonlinear systems. This includes cascaded and combination form of controllers, hybrid form of optimization algorithms in respective control system and related domains. Detailed literature review on different control mechanism used to handle the balancing, tracking and positioning of 2DoF helicopter system and 2DoF ball balancer system is done to identify the hybrid optimization algorithms and cascaded classical-intelligent control mechanism to control these robotic systems.

Chapter 3 is about providing a detailed mathematical modelling of two benchmark nonlinear systems i.e. 2DoF ball balancer and 2DoF helicopter system using classical control mechanism. This modelling is providing proper problem formulation in these two benchmark nonlinear systems.

Chapter 4 develops the control mechanism for 2DoF ball balancer and 2DoF helicopter system while optimizing the proportional integral derivative constrained parameters by using teaching learning based optimization algorithm. This optimization technique handles the parameter uncertainties of these nonlinear systems efficiently.

Chapter 5 is about providing the real work like situations in laboratory itself in the presence of external disturbance. Henceforth, improved teaching learning based optimization technique is used to control the variations due to external disturbance in 2DoF helicopter and 2DoF ball balancer system.

Chapter 6 develops the automated formation of initial constrained parameters for classical controller by using cascaded fuzzy-PID controller to handle the trajectory tracking, positioning and balancing in 2DoF helicopter system and 2DoF ball balancer system.

Chapter 7 develops new optimization technique to control the 2DoF ball balancer system and 2DoF helicopter system. In here, command phase and co-operate phase are introduced to Giza pyramid construction algorithm and hence developing a novel approach to tune the constrained parameters of fuzzy-proportional-integral-derivative controller while balancing and tracking the above mentioned two robotic systems.

Chapter 8 provides a summary of the methodologies employed in this study and additionally outlines potential avenues for future research.

Chapter 2. Literature Review

Several control strategies and optimization techniques have been suggested in the literature for nonlinear systems. Optimization algorithm is a prevalent and substantial pattern with a wide range of applications. In virtually every aspect of the engineering and manufacturing sector, efforts are made to optimize processes, aiming to either minimize costs or maximize outcomes, profits, productivity, and overall efficiency. The dynamic behaviour evolving over time in nearly all real-world systems contributes to their complexity, prompting researchers to increasingly focus on nonlinear systems due to their uncertainties and unpredictable nature. In the literature, numerous controllers have been utilized and optimized to handle nonlinearities and enhance system performance. The crucial domain of controlling autonomous robots to carry out intricate tasks in dynamic environments has been a focal point in the field of control system. Many of these control surfaces are directly associated with position control, path planning, trajectory tracking, and the balancing control of vehicles. This chapter offers a summary of the fundamental and recent advancements in theories and methodologies aimed at achieving control aspects in benchmark robotic systems. Trajectory tracking and attitude control, examined through a two-degree-of-freedom (2DoF) helicopter model, specifically the twin-rotor multi-input multi-output system, is discussed deeply in this chapter. Subsequently, the chapter explores position tracking and balancing control in a two-degree-of-freedom (2DoF) ball balancer system also. This chapter provides a summary of the fundamental theories and the latest advancements in collecting control elements for benchmark nonlinear systems, which is done by using several control techniques and optimization algorithms. A detailed literature review of following topics is done in this chapter:-

- Optimization Algorithms
- Optimization of classical controller
- Control mechanisms used for 2DoF ball balancer system
- Control mechanisms used for 2DoF helicopter system

Let us go through the literature survey of all these topics one by one.

2.1 Optimization algorithms

Optimization is a critical practice with a wide range of applications. In engineering and manufacturing, there is a constant effort to optimize various elements, such as reducing costs or improving outcomes, productivity, and efficiency. Optimization challenges are present in many aspects of our lives, both obvious and hidden. Any practical system, or even a part of it, can be seen as an optimization system with one or more embedded optimization issues. The main goal of optimization algorithms is to maximize efficiency by continually seeking more precise and adaptable solutions for a given problem [37, 38]. This means that problems must be reframed within the optimization context, which involves creating a "search space" for potential solutions and evaluating their performance against set criteria [39, 40]. Optimization helps address problems characterized by significant nonlinearity, complexity, and large solution spaces. As problems become more complex, there is a need for methods that can provide high-quality solutions within a reasonable timeframe using available resources. Metaheuristic algorithms are a prime example, having become a vital part of all optimization processes. Achieving an optimal solution is the final stage in the optimization process. This process includes mathematically characterizing the system, identifying constraints, defining system attributes, and formulating an objective function.

In a broader context, the optimization process can be classified into two categories: (a) exact methods and (b) approximation methods [41]. The exact approach strives for the optimal solution, ensuring the best possible answer, while the approximation method aims to provide a high-quality solution within a reasonable timeframe, though not necessarily achieving optimality. Examples of exact optimization techniques include the branch and bound approach and dynamic programming, whereas approximate methods encompass cut & plane, local search, scatter search, genetic algorithms, among others. Additionally, there are two types of approximate procedures: approximation algorithms and heuristic techniques. The former guarantees a proven arrangement quality and runtime limitations, while the latter focuses on obtaining a practical excellent arrangement within a reasonable time. Heuristic-based algorithms are highly specific to the problem at hand. Metaheuristics, as a type of algorithm, embody basic heuristics similarly to a governing system but without specifying a particular problem or domain, making them applicable to any optimization task. The term "metaheuristics" was coined by Glover [5], emphasizing their logical improvement to achieve an acceptable arrangement within a sufficiently short computing time. The simplicity and ease of implementation, lack of requirement for slope data, avoidance of local optima, and versatility across a range of problems involving different controls make meta-heuristic optimization valuable in addressing various ongoing challenges.

Heuristic optimization methods were first proposed in the early '70s [42]. While these methods don't assure the optimal solution, they are capable of finding solutions that are close to optimal and feasible within a reasonable timeframe. The concept of heuristics was initially introduced [43] by G. Poyla in 1947, with its actual development gaining momentum after 1960. Essentially, heuristics are designed to provide improved computational performance most of the time, albeit at the expense of reduced accuracy compared to traditional optimization algorithms. Since heuristics are tailored to specific problems, they leverage domain-specific knowledge and are well-defined primarily for basic problems [44]. To be more precise, heuristics are approaches that use readily available but loosely structured information to guide problem-solving in both human beings and machines. Classically, the optimization algorithms are divided into two types – heuristic optimization algorithms and metaheuristic optimization algorithms. Heuristic optimization techniques and metaheuristic optimization techniques are both strategies employed to discover approximate solutions for complicated optimization problems. Nonetheless, there are distinctions between the two in terms of their scope, generality, and the extent of problem-specific knowledge involved. Key points of these algorithms are discussed in further sub-sections of this work.

2.1.1 Heuristic optimization techniques

- Tailored for Specific Problems: Heuristics are primarily crafted for particular problem types, utilizing domain-specific knowledge to steer the search for solutions.
- Limited Transferability: They may not easily adapt to different problem domains without substantial modification.
- Generality: Reliance on Specific Knowledge: Heuristics often depend on problem-specific knowledge and may be customized to exploit the structure of a given optimization problem.
- Localized Exploration: Many heuristics involve strategies focused on local exploration, iteratively refining the solution space by enhancing the current solution.
- Optimality: No Assurance of Optimality: Heuristic methods do not ensure the discovery of the optimal solution. Instead, they concentrate on swiftly identifying good, feasible solutions.

Most famous example for heuristic optimization algorithm is Hill Climbing. Hill Climbing is a localized search algorithm that progressively advances toward an elevated point in the solution space through iterative steps, effecting incremental enhancements. Termination occurs when no superior neighbouring solution is identified. But, this algorithm fails drastically when it comes to multiple optima solution kind of problems. The 2DoF robotic systems are such kind of nonlinear problem only. So we need better understanding of metaheuristic optimization techniques in order to handle the 2DoF robotic systems.

2.1.2 Metaheuristic optimization techniques

- Universal Problem-Solving: Metaheuristics are conceived as high-level, universal strategies applicable to a diverse array of optimization problems without heavy reliance on problem-specific knowledge.
- Wide Applicability: They are frequently suitable for various domains and problems with minimal adaptation.
- Generality: Independence from Specific Problems: Metaheuristics are not tied to specific problems and do not necessitate detailed knowledge about the specific characteristics of the optimization problem.
- Global Exploration: Metaheuristics often employ global search strategies, traversing the entire solution space to escape local optima and discover superior solutions.
- Optimality: No Guarantee of Optimality: Similar to heuristics, metaheuristics do not assure the identification of the optimal solution; instead, they focus on delivering high-quality solutions within a reasonable timeframe.

This exploration delves into fundamental terminology within the realm of metaheuristic computing to present the concept of metaheuristic computing with utmost clarity. Definition 1 – “heuristic is defined as a problem-solving reasoning methodology that facilitates the derivation of a solution through trial-and-error and/or the application of rules of thumb". Definition 2 – "computing, in a narrow sense, is described as the application of computers to solve a specific problem through imperative instructions. In a broader sense, it constitutes a process where a system transforms given information or instructions into expected intelligent behaviours". Synthesizing these definitions, the concept of metaheuristic computing can be articulated as follows – as per Definition 3 – "Metaheuristic computing is an adaptive and/or autonomous methodology for computing that employs general heuristic rules, algorithms, and processes to address a category of computational problems"[45].

In essence, heuristics are more tailored to specific problems, demanding a profound understanding of the particular optimization problem. On the other hand, metaheuristics are crafted as general problem-solving frameworks, applicable across a broad spectrum of problems without relying on intricate problem-specific knowledge. Metaheuristics often exhibit a higher level of abstraction, enabling broader applicability but potentially sacrificing some problem-specific optimization capabilities found in heuristics. Both approaches share the common objective of effectively discovering satisfactory solutions to complex optimization problems. There are numerous examples of metaheuristic optimization techniques in literature including Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Teaching Learning Based Optimization (TLBO) algorithm, Giza Pyramid Construction (GPC) optimization algorithm.

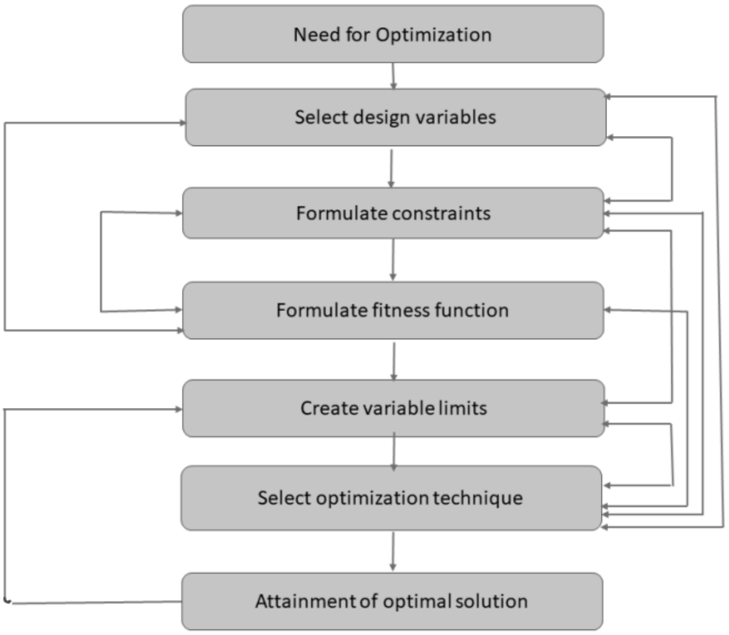


Fig. 2.1 General steps followed by optimization techniques

Metaheuristic optimization algorithms employ diverse strategies, each guided by unique principles in their quest for optimal or near-optimal solutions to intricate problems. Step wise steps followed by these optimization techniques to conquer a problem is explained in fig. 2.1. Genetic Algorithms (GA), drawing inspiration from biological evolution, adhere to the principles of natural selection and genetics, utilizing mechanisms such as crossover and mutation to progressively evolve a population of potential solutions [46]. Ant Colony Optimization (ACO), influenced by ant foraging behaviour, depends on pheromone trails to direct artificial ants in collaboratively constructing solutions through a decentralized approach [47]. Particle Swarm Optimization (PSO) takes cues from social behaviour, with particles adjusting positions based on individual experiences and global optimal solutions [48]. Each metaheuristic algorithm adheres to its distinct guiding principles, reflecting the array of inspirations shaping their approaches to optimization.

Most famous optimization algorithms are of four types – "Evolutionary based", "Trajectory based", "Nature Inspired" and "Ancient Inspired". Some of the recognized algorithms in the realm of evolutionary strategies include Genetic Algorithm (GA), Evolution Strategy (ES), Evolution Programming (EP), Differential Evolution (DE) and Bacteria Foraging Optimization (BFO) among others. Additionally, well-known swarm intelligence-based algorithms comprise Particle Swarm Optimization (PSO), Shuffled Frog Leaping (SFL), Ant Colony Optimization (ACO), Artificial Bee Colony (ABC), Fire Fly (FF) algorithm, and more. Apart from these, various algorithms like Iterated Local Search (ILS) and Guided Local Search (GLS) are based on trajectories followed in parent algorithm. Whereas the Ancient inspired algorithm is Giza Pyramid Construction (GPC) algorithm [49].

All evolutionary, trajectory and swarm intelligence-based algorithms are probabilistic and share common controlling parameters like population size, number of generations, elite size, etc. However, individual algorithms also require specific control parameters. For instance, GA utilizes mutation probability, crossover probability, and selection operator; PSO employs inertia weight, social and cognitive parameters; ABC depends on the number of onlooker bees, employed bees, scout bees, and limit; and HS algorithm considers harmony memory consideration rate, pitch adjusting rate, and the number of improvisations. Similarly, other algorithms like ES, EP, DE, BFO, AIA, SFL, ACO, etc., necessitate tuning their respective algorithm-specific parameters. The precise tuning of these parameters is crucial as it significantly impacts the performance of these algorithms. If not properly tuned, algorithm-specific parameters may increase computational effort or result in local optimal solutions. Recognizing this challenge, Rao et al. introduced the teaching-learning-based optimization (TLBO) algorithm [50], which stands out for not requiring any algorithm-specific parameters. TLBO, hinging on basic parameters such as population size and generation count, has garnered broad acknowledgment within the optimization research community [51]. Since its introduction, the Teaching-Learning-Based Optimization (TLBO) algorithm has seen a growing adoption across diverse fields of engineering and science. A thorough endeavour has been undertaken to gather data from numerous publications concerning the TLBO algorithm, recognizing the potential existence of further contributions in conferences and journals. Initially conceived to address constrained mechanical design optimization tasks, the Teaching-Learning-Based Optimization (TLBO) algorithm has expanded its scope into various applications, highlighting its adaptability and effectiveness. Its capabilities have been showcased in multi-objective design optimization tasks, notably in refining a robotic gripper and resolving clustering issues [52].

Expanding its applications, TLBO played a crucial role in optimizing the shape and size of truss structures while integrating dynamic frequency constraints [53]. A significant advancement came in the form of a cooperative co-evolutionary TLBO algorithm, which introduced a refined exploration approach tailored for addressing large-scale global optimization challenges [54]. Demonstrating its versatility, the algorithm made strides in Bioinformatics, where the multi-objective Teaching-Learning-Based Optimization (MO-TLBO) method effectively tackled the complex Motif Discovery Problem (MDP) [55]. Noteworthy enhancements to the algorithm included the integration of a swarm-based niching technique, a novel mutation strategy in the learner stage, and modifications to the teacher and learner phases, all contributing to its continuous refinement [56]. Applied to scenarios such as optimal capacitor placement in distribution networks and stochastic models for energy management, the algorithm consistently demonstrated its prowess [57]. Further enhancements, such as the Elitist Teaching Learning Oppositional Based (ETLOBA) algorithm [58], were introduced to augment TLBO's accuracy, as evidenced by comparative analyses with algorithms like HS, Improved Bees Algorithm (IBA), and ABC. The TLBO algorithm, akin to a literary masterpiece, continually unfolds its narrative across various optimization landscapes, showcasing its mettle against formidable counterparts such as GA, ABC, PSO, HS, DE, and Hybrid-PSO through rigorous experimentation on diverse benchmark problems [59]. Utilized the TLBO algorithm to optimize the design of planar steel frames. Compared the outcomes of TLBO with those obtained using GA, ACO, Harmony Search (HS), and the enhanced Ant Colony Optimization (IACO) [60].

The TLBO method was employed to select distinctive oligonucleotide primers [61]. Digital hearing aids based on IIR were fashioned using the TLBO algorithm [62]. The TLBO algorithm was utilized for the efficient design of grillage systems adhering to LRFD-AISC standards [63]. Additionally, it was applied to optimize truss structures, with the method's validity demonstrated through four design examples [64]. A new application involved suggesting the TLBO algorithm to determine the optimal placement and size of Distributed Generation (DG) units within distribution systems [65]. To address the challenge of multi-objective short-term optimal hydro-thermal scheduling, an algorithm called hybrid differential evolution and TLBO (hDE-TLBO) was introduced [66]. Additionally, a hybrid self-evolving algorithm was recommended for solving a nonlinear optimal power flow problem, specifically aiming to minimize the fuel costs associated with thermal units [67].

The effectiveness of the TLBO algorithm was demonstrated in addressing constrained mechanical design problems [68]. An improved version, referred to as the Ameliorated TLBO algorithm, was proposed to elevate solution quality and hasten the convergence speed of TLBO [69]. In an effort to accelerate convergence speed and enhance solution quality, a quasi-opposition-based learning concept was integrated into the original TLBO algorithm [70]. Another algorithm, the multi-objective Teaching Learning Algorithm based on Decomposition (MOTLA/D), was put to the test on three systems, with results compared to other multi-objective algorithms based on decomposition [71]. A modification involving a self-adaptive mechanism was introduced in the phase of the TLBO algorithm, aiming to achieve a balance between exploration and exploitation capabilities [72]. TLBO algorithm found practical application in optimizing process parameters for three machining processes, including the advanced abrasive water jet machining process, as well as the conventional processes of grinding and milling [73]. Conducted the optimization of parameters for a multi-pass turning operation through the application of the TLBO algorithm. Examined two distinct scenarios that had been previously explored by different researchers using various optimization methods like SA, GA, ACO, PSO, and others [74]. Implemented [75] a customized version of the Teaching–Learning-Based Optimization (TLBO) algorithm to address the short-term Hydro-Thermal Scheduling problem, incorporating considerations for valve point effects and prohibited discharge constraints. Applied the modified TLBO to tackle the same scheduling problem, taking into account the intricate and nonlinear relationships among the variables. Employed TLBO in classification tasks using neural networks within the realm of data mining. Introduced the Orthogonal Design based TLBO (OTLBO) algorithm [76], integrating it with statistical optimal methods to generate optimal offspring. Experimented with the Weighted TLBO (WTLBO) algorithm [77] on various benchmark optimization problems. Utilized TLBO for parameter identification challenges in designing a digital Infinite Impulse Response (IIR) filter [78]. Proposed an Improved TLBO algorithm with a Memetic method (ITLBO-M), enhancing global exploring ability through memetic methods and improving local search ability through one-to-one teaching [79]. Applied a modified TLBO for the design of a Proportional-Integral (PI) controller-based Power System Stabilizer (PSS) [80]. Introduced an Improved Harmony Search Based Teaching Learning (HSTL) optimization algorithm [81], aiming for a balance between convergence speed and population diversity. Critically examined a note challenging the performance supremacy of the TLBO algorithm, presenting a re-examination of views and experimental results in an objective manner [82]. [83] Surveyed the operational mechanism of TLBO for real-parameter optimization problems and categorized its real-world applications. Proposed a TLBO algorithm for multi-objective optimization problems (MOPs), incorporating non-dominated sorting and crowding distance computation concepts [84]. [85] Provided details on integrated maintenance scheduling for secure operation, formulating it as a complex optimization problem affecting unit commitment and economic dispatch schedules. Applied TLBO in hybrid fuzzy wavelet neural network for heart disease diagnosis [86]. Analysed TLBO algorithm performance on combinatorial optimization problems [87], specifically flow shop (FSSP) and job shop scheduling problems (JSSP). Employed TLBO for solving Economic Load Dispatch (ELD) problems with varying linear and non-linear constraints [88]. Used TLBO for solving optimal power flow problems, demonstrating effectiveness on IEEE 30-bus and IEEE 118-bus test systems [89]. Explored TLBO efficiency in solving benchmark problems related to truss structures with discrete design variables, comparing results with existing literature [90]. Applied modified TLBO and Double Differential Evolution (DDE) algorithm for solving Optimal Reactive Power Dispatch (ORPD) problems [91]. Proposed a multi-objective TLBO algorithm for solving the Motif Discovery Problem (MDP) [92], comparing results with other algorithms. Modified teacher and learner phases of TLBO for optimal location determination of AVRs in the distribution system [93]. Suggested an improved TLBO algorithm adapted to the enhanced framework of Particle Swarm Optimization (PSO) known as Bidirectional Teaching and Peer Learning PSO

(BTPLPSO) [94]. Proposed a method to quantify carbon emissions in turning operations using TLBO, including a multi-objective TLBO algorithm considering carbon emissions and operation time minimization [95]. Introduced a harmonic elimination technique for reducing harmonics in voltage source inverters using a modified TLBO algorithm [96]. [97]Presented an efficient method to extract all five parameters of a solar cell from a single illuminated current–voltage characteristic using TLBO. Integrated differential operators into TLBO with a Latin hypercube sampling technique to improve the initial population's flatness during the strip coiling process [98]. Used TLBO for optimal selection of design and manufacturing tolerances, comparing results with GA, NSGA-II, and MOPSO [99]. Considered mathematical models of casting processes for TLBO-based parameter optimization and proposed a multi-objective improved TLBO algorithm for unconstrained and constrained multi-objective function optimization. Evaluated performance on CEC 2009 competition test problems [100 – 101]. [102] Applied QTLBO to solve the unit commitment problem, comparing results with other algorithms. Incorporated oppositional-based learning into basic TLBO for solving combined heat and power dispatch problems [103]. Investigated the performance of a new TLBO variant for global function optimization, comparing with advanced PSO, DE, and ABC variants on CEC 2005 benchmark functions [104]. Proposed a modified TLBO algorithm with self-adaptive wavelet mutation strategy and fuzzy clustering technique [105]. Introduced a hybrid TLBO-DE algorithm [106] for chaotic time series prediction and demonstrated its effectiveness on typical chaotic nonlinear time series prediction problems. Formulated a multi-objective optimization problem and used a modified TLBO algorithm for coordinating TCSC [107], SVC, and power angle difference damping characteristics. [108] presented a Simplified Teaching–Learning-Based Optimization (STLBO) algorithm for solving the challenging NP-hard combinatorial optimization problem of Disassembly Sequence Planning (DSP). Introduced a hybrid TLBO [109] combining TLBO for solution evolution and variable neighbourhood search for fast solution improvement in Permutation Flow Shop Scheduling. Proposed a compact TLBO [110] algorithm to reduce memory requirements, utilizing adaptive statistic description to replace the process of a population of solutions. Modified TLBO with a feedback phase [111], DE algorithm operations, and chaotic perturbation mechanism to improve performance. Introduced the area copying operator of the producer–scrounger model into TLBO to decrease computation cost and enhance global performance for global optimization problems [112]. Applied TLBO techniques in forecasting a financial derivatives instrument (commodity futures contract index) using machine learning methods [113]. Presented a TLBO-based framework for computing coefficients for quadratic and cubic cost functions, valve point loading, piece-wise quadratic cost, and emission functions [114]. Proposed a new hybrid algorithm, Teaching–Learning-Based Cuckoo Search (TLCS), for parameter optimization in structure designing and machining [115]. Highlighted a multi-objective optimal location of on-load tap changers (OLTCs) in distribution systems at the spirit of distributed generators (DGs) using TLBO coupled with the SA algorithm (SA-TLBO) [116]. Introduced an improved TLBO variant for simultaneous allocation of distributed resources in radial distribution networks, considering multi-level load scenarios [117]. Applied TLBO to determine optimal machining conditions for satisfactory machining performances, comparing its application potential to that of GA [118]. Advocated for the novel Auto-

TLBO clustering algorithm, combining automatic k-value assignment and cluster validations into TLBO [119]. Presented a secured optimal power flow solution by integrating thyristor controlled series compensator (TCSC) with the optimization model under overload conditions, utilizing TLBO [120]. Conducted single and multi-objective design optimization of a heat pipe using TLBO, comparing results with NPGA, GEM, and GEO algorithms [121]. Evaluated TLBO performance in obtaining the optimum geometrical dimensions of a robot gripper, considering five objectives [122]. Explored TLBO's effectiveness in obtaining optimal design and operating parameters for a smooth flat plate solar air heater (SFPSAH), with thermal efficiency maximization as the objective function [123]. Investigated TLBO's performance in multi-objective design optimization of a plate fin heat sink equipped with flow-through and impingement-flow air cooling system, considering entropy generation rate and material cost as objective functions [124]. Presented the design and analysis of a Proportional-Integral-Double Derivative (PIDD) controller for Automatic Generation Control (AGC) of multi-area power systems using TLBO [125]. Hybridized TLBO with Differential Evolution for estimating unknown proton exchange membrane fuel cell (PEMFC) model parameters [126]. Proposed an effective TLBO algorithm for solving the flexible job shop scheduling problem, comparing results with other optimization algorithms [127]. Introduced an integrated approach for real-time model-based state of charge (SOC) estimation of Lithium-ion batteries, utilizing TLBO and the least square method for offline optimization [128].

TLBO method distinguishes itself as a beneficial strategy for parameter optimization, attributed to its distinctive learning-inspired approach. This method showcases a notable capacity to converge towards optimal solutions, steering clear of stagnation in local minima. Its intrinsic simplicity, efficiency, and versatility position is well-suited for addressing a broad spectrum of challenges in parameter optimization.

2.2 Classical controller

Over the years, feedback control mechanisms have exerted a significant influence on various fields such as manufacturing, robotics, aviation, and process control [129 – 133]. Numerous efficient, robust, and adaptive controllers have been proposed alongside the widely used classical proportional-integral-derivative (PID) controller. Despite the availability of alternative controllers, the acceptance and reputation of PID controllers in control systems remain unparalleled and supreme. PID control systems can be applied in different modes—proportional only (P-mode), proportional and integral (PI mode), proportional and derivative (PD-mode), and proportional, integral, and derivative (PID mode)—based on process requirements [134]. PID controllers have been employed in industries for process control applications for many years. Despite their long history, PID controllers continue to be the most widely used controllers in both process and manufacturing industries today. Research indicates that approximately ninety percent (90%) of process industries utilize PID controllers. This widespread adoption is attributed to the robustness, simplicity, and ease of retuning control parameters. The PID controller has traditionally been considered the best controller in the absence of fundamental process knowledge [135]. Despite these advantages, PID controllers also face some shortcomings, including unwanted speed overshoot and sluggish response due to unexpected variations in load torque and sensitivity to controller gains [136].

Controller tuning has long been a crucial aspect of feedback controllers, making it a significant area of research in both academic and industrial domains. The exploration and study of dynamic systems to develop efficient, reliable, and promising controllers fall under the purview of control engineering. Following the establishment of PID control, considerable interest in tuning techniques aimed at ensuring the excellent performance of PID controllers emerged. The initial rules for tuning PID controller parameters were introduced by Ziegler and Nichols in 1942 [137 – 138], with subsequent proposals of various tuning rules. Some of these approaches primarily focused on stabilizing linear systems [139], making them less suitable for nonlinear dynamic models. In contrast, certain techniques encompassed nonlinear systems with the aim of obtaining control variables satisfying stability criteria [140]. However, these latter strategies do not guarantee specific response characteristics. Consequently, a significant challenge in the field of control engineering revolves around the appropriate tuning of controller parameters. The tuning of controller variables is crucial for stabilizing closed-loop control systems and achieving objectives related to stability, durability, performance tracking, performance measurement, noise reduction, disturbance rejection, and robustness against environmental uncertainties.

When presented with a set of objectives, various tuning approaches exist for PID controllers. Numerous authors and scholars have introduced diverse classifications of PID tuning strategies. For a more thorough and detailed examination of this classification, along with application strategies, refer to Moradi and Johnson [141 – 142].Additionally, [143] provides an alternative classification for tuning PID parameters. These approaches are broadly divided into classical and computational techniques:

- <u>Traditional Approaches:</u> These approaches revolve around making assumptions regarding the plant model and the desired output. They aim to extract certain system features analytically or graphically, which are then employed to select controller settings. These methods are user-friendly, computationally efficient, and can serve as an initial step in parameter adjustment. However, the controller settings often yield desired results directly due to the assumptions made, necessitating further tuning. Some classical tuning methods include the Ziegler and Nichols [137 – 138] method and the Cohen (1953) method.

- <u>Computational or Optimization Approaches:</u> These techniques involve utilizing data modelling and optimization strategies of a cost function to tune PID parameters. These methods for adjusting controller parameters rely on a cost function that they seek to minimize. There are six commonly used cost functions for tuning PID controller parameters, which are discussed later in this work.

2.3 Computational techniques for classical controller tuning

Computational approaches are utilized for the self-tuning or automated tuning of PID controller parameters. A more detailed classification based on nature and application can be categorized into five (5) groups:

I. Analytical Methods: These methods involve analysing the closed-loop model to achieve stability. It includes computing the relationship between a given plant model and the set objective to determine appropriate PID parameters.
II. Heuristic Methods: Heuristic PID tuning strategies evolve through practical experience, where controller variables are manually selected based on the experimental knowledge of a seasoned designer. This designer utilizes information from controlled variable estimates to establish correct variable-performance relationships. The heuristic method can also serve as a formula or rule base for online tuning, often involving trade-offs in design goals or objectives.
III. Frequency Response Methods: These PID parameter tuning methods are typically offline and popular in academic settings. The technique is based on the characteristics of the target process or system, such as loop shaping, and has been presented in the use of frequency response for tuning PID controller parameters.
IV. Optimization Methods: PID controller tuning strategies in this category can be considered an exceptional form of feedback optimal control system. PID variables are obtained spontaneously using offline mathematical programming or numerical optimization techniques for a singular objective. The computation can also be implemented using computerized heuristics or evolutionary algorithms for multiple design objectives. The main characteristic of these methods is that control parameters are fixed and acquired by the solution of an offline numerical optimization method. Offline numerical optimization techniques use the derived state vector of the plant by simulation or the real plant vectors. The obtained fixed control parameters are then integrated into a closed-loop system. This review falls into this class of tuning.
V. Adaptive Tuning Methods: Adaptive tuning strategies are online, real-time tuning methods that involve the use of automated online mechanisms to tune PID controller parameter gains. The key feature of these techniques is that control parameters vary over pre-set time intervals in the closed-loop system.

This thesis is focussed on Adaptive tuning using different optimization methods, that is why the literature review is narrowed down to it. In recent literature, heuristic algorithm-based optimization strategies have emerged as a powerful tool for addressing various challenges in control engineering [144]. Metaheuristic algorithms are widely employed in process control due to their architectural simplicity, effective optimization capabilities, and rapid response. In comparison to traditional optimization approaches, metaheuristic algorithms demonstrate higher efficiency in solving optimization problems with higher dimensions. Their adaptability to existing classical controller design methodologies is facilitated by their resilience. Regardless of model order, metaheuristic algorithms can serve as a fundamental mechanism for developing both traditional and enhanced structured controllers for a range of unstable operational models. The significance of automatic PID tuning algorithms has garnered considerable attention in the industry, particularly in the last two decades [145]. Various methods, including metaheuristic techniques like GA, PSO, and SA, among others, have been applied. In the subsequent paragraphs, an overview of these algorithms and their applications in PID tuning is provided.

Classical PID controller optimization using GA algorithm: “Genetic Algorithms are search and optimization methodologies inspired by two biological principles known as the process of natural selection and the mechanism of natural genetics [146]. GA [147] is a stochastic global search method that emulates the natural evolution process. It belongs to the category of probabilistic optimization techniques utilizing natural selection and genetic inheritance to address problem-solving challenges. Presently, Genetic Algorithms (GA) have garnered significant attention, and ongoing research has delved into exploring its application. In the field of Control Engineering, its implementation has progressed significantly. Despite the multifaceted considerations of performance, system stability, static and dynamic indices, and overall system robustness in control system design, GAs have been employed to fine-tune PID parameters for diverse processes or plants. For instance, optimal methods for tuning PID settings in DC motor speed control have been discussed [148 – 149]. GA-controlled PID controller tailored for synchronous generators to enhance damping and ensure power system stability is introduced [150]. The design of a PID controller for a cascade control process using GA is elucidated [151]. An investigation is introduced into the applicability of genetic algorithms for the automatic tuning of PID controller parameters [152]”.

Classical PID controller optimization using ACO algorithm: “The ant colony optimization (ACO) algorithm is a unique approach inspired by the collective behaviour of insect swarms and initially developed for combinatorial problems. ACO serves as a stochastic metaheuristic method for addressing combinatorial optimization challenges, employing artificial ants to navigate through solution spaces. The primary objective of ACO is to discover shorter routes from ant nests to food sources by leaving behind a chemical compound, known as pheromone, enabling communication among ants. Proposed by Dorigo in his 1992 PhD thesis [47] & [153], the initial version of ACO laid the foundation for a challenging optimization algorithm rooted in both simple and complex foraging ant behaviours, gaining significant interest in the 1990s and beyond. In [154] applied ACO to tune PID controller parameters for a DC motor in a robotic arm, demonstrating superior performance compared to conventional tuning methods. Other applications include PID tuning for quadrotor stabilization [155], optimal PID regulator synthesis for human heart control [156] using ACO-based optimization, and various PID tuning applications such as Load Frequency Control [157], Autonomous Underwater Vehicle [158], Single Machine Infinite Bus control [159], position control of a DC motor [160] and multi-objective ACO for PID [161]”.

Classical PID controller optimization using ABC algorithm: “The Artificial Bee Colony (ABC) metaheuristic optimization technique was introduced by Karaboga in 2005, drawing inspiration from the intelligent foraging behaviour of honey bees.

The method is rooted in the foraging model of honey bee colonies presented by Karaboga [162], comprising three main components: working and unemployed foraging bees, as well as food sources. Employed and unemployed foraging bees seek abundant food sources in proximity to their hive, while the hive itself serves as the third component. The concept involves forager recruitment to rich food sources, leading to positive feedback, and forager desertion of low food sources, resulting in negative feedback—essential behaviours for self-organization and collective intelligence. Further details and application strategies of ABC are discussed comprehensively in [163]. The ABC algorithm is applied for precise tuning of PID settings, employing benchmark functions for testing. The algorithm's performance is compared to other approaches based on overshoot, settling time, and minimal error [164]. In [165], the author proposed the use of the ABC algorithm for tuning PID controller regulation parameters in a DC motor. ABC algorithm is employed for obtaining optimal PID controller parameters for higher order oscillatory systems [166]. Additional applications include the control of fractional order systems [167], suppression of hub motion and endpoint vibration in a double-link flexible robotic manipulator [168], control of a bench-scaled nonlinear dynamical system [169] and tuning of a single-phase inverter [170]".

Classical PID controller optimization using BA algorithm: "The Bat Algorithm (BA) is a heuristic method inspired by the echolocation behaviour of natural bats, particularly microbats, in locating prey and distinguishing between different types of insects even in low-light conditions [171], where Yang introduced the BA, drawing influence from the sonar capabilities of microbats. In the original population, each bat updates its position through a correlated process utilizing echolocation. Echolocation involves emitting powerful ultrasonic pulses to generate echoes, which are then received with varying latencies and sound frequencies. This enables bats to precisely identify their prey. In the natural environment, echolocation events are brief, lasting only a few thousandths of a second (around 8–10 ms), with frequencies ranging from 25 to 150 kHz and corresponding to air wavelengths of 2–14 mm [172]. Microbats employ sound waves for prey detection, emitting small signals during travel. However, when a potential prey is nearby, they increase the pulse rates and tune up the frequency. This frequency tuning, in combination with the pulse emission speed, reduces the wavelength of echolocations, enhancing detection precision. The BA has been successfully applied to optimize Proportional-Integral-Derivative (PID) parameters in various process plants. [173] employed BA to fine-tune PID controller parameters for a liquid level control system in a connected tank, widely used in industries such as mineral oil, food manufacturing, and water purification. In [174], BA was utilized to optimize the settings of a centrally controlled Proportional-Integral (PI) controller for a coal gas turbine, a non-linear multidimensional process with complex relationships across control loops. In [175], BA is used to tune PID controller parameters for a servo motor. Other applications of BA in PID tuning include speed control of brushless direct current drives [176], bio-inspired robot manipulators [177], and the control of microelectromechanical systems (MEMS) gyroscopes [178]".

Classical PID controller optimization using PSO algorithm: "In 1995, Kennedy and Eberhart introduced the PSO algorithm [48], a metaheuristic approach grounded in swarm intelligence principles, designed to address complex mathematical challenges encountered in engineering [179]. Renowned for its versatility, PSO has become a widely used optimization technique with diverse applications in fields such as medicine [180], finance [181], economics [182], as well as security and military domains, biology, system identification, and more [183 – 184]. In various academic pursuits, PSO has been employed to address issues in structural, hydrological, and geotechnical civil engineering [185]. Additionally, PSO has recently found utility in solving problems within the realm of operations research [186]".

Classical PID controller optimization using Cukoo algorithm: "Yang and Deb introduced the Cuckoo Search (CS) metaheuristic evolutionary optimization algorithm [187], drawing inspiration from the cuckoo bird species. Cuckoos are captivating not only for their enchanting songs but also for their proactive breeding strategies. CS techniques have been applied in numerous studies to fine-tune PID controller parameters. [188] exemplify the utilization of the cuckoo optimization approach in PID parameter tuning for monitoring an inverted pendulum, experimental validation of the Van der Pol oscillator, and the stability analysis of a capsizing problem in nonlinear systems. [189] present a novel tuning approach for PID parameters in the AVR system, leveraging the CS algorithm along with a new frequency-domain performance benchmark. Introduces an innovative method for designing PID controllers using the CS optimization algorithm, aimed at enhancing the performance of buck-boost converters in LED driver circuits [190 – 191]".

Classical PID controller optimization using GWO algorithm: "Grey Wolf Optimization (GWO) is a recently introduced metaheuristic optimization algorithm, as proposed by Mirjalili et al. [192]. The inspiration for GWO comes from the social structure and intelligent hunting strategies of grey wolves, known for their dominant position within their habitats and group dynamics typically consisting of 5–12 individuals. The distinctive feature of GWO, setting it apart from other metaheuristic algorithms, lies in its emulation of the well-defined social hierarchy among grey wolves. The primary objective of this social hierarchy is to evolve candidate solutions during each iteration, closely resembling the hunting behaviour of grey wolves as they locate and attack prey [193]. In the field of control engineering, there has been a notable surge in publications exploring the application of the GWO algorithm for tuning PID controller parameters. [194] proposed a GWO scheme to optimize the variables of a PI controller for a closed-loop condenser pressure control mechanism. The results demonstrated the effectiveness of the Grey Wolf (GW) optimizer compared to Genetic Algorithms (GA) and Particle Swarm Optimization (PSO). Similarly, [195] tuned the parameters of a basic PID controller using the GWO algorithm for a magnetic levitation system, showing superior results compared to the traditional Ziegler-Nichols (ZN) [137 – 138] tuning scheme. [196] explored the application of the GW optimizer to fine-tune control variables of a PID controller for regulating the speed of a second-order DC motor system. GWO has found application in various other domains for PID tuning, including robotics [197], speed control of DC motors [198], inverted pendulum systems [199], voltage regulation [200], and wind turbines [201]".

Classical PID controller optimization using TLBO algorithm: “Rao et al. proposed [50] the Teaching-Learning-Based Optimization (TLBO) as a solution for addressing global mechanical design challenges through large-scale non-linear optimization problems. TLBO, a population-based metaheuristic search algorithm, emulates the teaching and learning methods commonly observed in a classroom setting. In TLBO, a group of students collectively forms a solution, and the fitness of these solutions is termed as results or grades. The TLBO algorithm refines each learner's grade by incorporating learning from both the teacher and peer interactions. In the context of PID parameter tuning, TLBO has been employed to optimize the variables of PID for various processes, including automatic voltage regulator (AVR) systems, Automatic Generation Control (AGC) in power systems and the enhancement of magnetic levitation system performance [125, 202]. The algorithm has also been applied to design a PID controller for an AVR device with one and two degrees of freedom using a teaching-learning-based approach [203]. Due to its attractive features, such as a simple concept, lack of precise algorithm variables, straightforward implementation, and rapid convergence, TLBO has garnered significant attention. It has been adapted for capturing limited data and applied in various domains, including multi-objective optimization [204], large-scale problems [54, 56], and dynamic optimization problems [205]. TLBO has proven effective in diverse scientific and engineering applications [206]”.

Traditionally, [207] and [208] provided an extensive examination of various strategies for position tracking and balancing control. These cutting-edge solutions primarily followed two distinct approaches. The first, known as cross-coupling control [209]–[211], involves considering the control loops of all machine axes simultaneously, with the control objective focused on regulating contour errors [212], [213]. The second approach, termed individual axis control, addresses each axis independently. In this method, the dynamics of other axes are treated as disturbances, and the control design simplifies to that of individual axes [214]. Both approaches relied on adjusting operational parameters (e.g., pitch and yaw movement, plate angle) [215], [216] or achieving precise position tracking [217], [218], ensuring the discrepancy between measured and desired paths is minimal while satisfying specific constraints. Additionally, it is noted that the development of control action in both approaches is interdependent.

Furthermore, when dealing with individual axis control, except for a few exceptions (such as varying gain, intelligent approaches, stochastic-based control accounting for unknown noises, and fault-tolerant controllers), most nonlinear control techniques for 2DoF systems neglect parameter estimation in system dynamics. This limitation can be addressed by incorporating nonlinear and adaptive control algorithms into individual axis control architectures, especially since the issue of nonlinear position control for flexible axes is extensively explored. Indeed, several relevant studies employing techniques from nonlinear control theory have been documented in the literature concerning position and balancing control. Subsequent sections provide a comprehensive overview of these techniques for achieving various control aspects.

2.4 Control mechanisms used for 2DoF ball balancer system

LQR controller: The Linear Quadratic Regulator (LQR) problem is formulated based on a dynamical system with a state evolution equation represented as a linear function. This linear function relies on the current state and input of the system and is subjected to quadratic cost. The primary goal of the controller is to minimize the cost function associated with system variables. In the control of the ball and plate system, an alternative synthesis approach utilizes repetitive and resonant controllers, with energy minimization influenced by the principles of the Linear Quadratic Regulator (LQR) theory [219]. The LQR control offers simplicity in programming and ensures asymptotic convergence of the system to the origin while following a desired trajectory [220]. Moreover, motion planning methods incorporating LQR and discrete model predictive controllers have been proposed for a ball-plate Mecanum robot system [221]. The Infinite Horizon Discounted LQR [222] establishes non-asymptotic bounds, but this comes at the expense of increased sample complexity compared to other model-free and model-based methods. A significant limitation of LQR is its reliance on accurate knowledge of the system's state, which is often impractical. For instance, when noise is present in the system, the deterministic assumption made for the transition of other states makes the state transition challenging.

Back-Stepping Controller: The back-stepping controller engages in a recursive process utilizing a Lyapunov function and adheres to a systematic design approach for diverse nonlinear systems. In the research conducted by Kazim et al., a robust back-stepping control strategy is employed for trajectory tracking in ball and plate systems, effectively handling bounded uncertainties with unknown periodicity [223]. This controller not only ensures global stability but also enhances tracking and transient performance. The drawback of the traditional back-stepping controller lies in its applicability only to state models in strict feedback form [224]. To overcome this limitation, a hybrid approach is devised by integrating the back-stepping controller with a $H\infty$ tracking controller, addressing external disturbances in a ball and plate system and demonstrating asymptotic closed-loop stability [225]. Other control techniques, such as adaptive fuzzy control [226] or neural networks [227], have been combined with back-stepping to address inaccuracies in the model. However, these methods necessitate access to complete state output information.

Proportional–Integral–Derivative (PID) controller: While there are numerous control strategies outlined in the literature for achieving self-balancing control in balancer systems, the PID controller stands out as widely employed in practical engineering applications. The nonlinear nature of the proportional integral derivative (PID) controller has been explored for one-to-one control in classical control of the ball on the plate system [228]. The PID controller's response is enhanced when structured based on the Generalized Kalman-Yanukovych-Popov lemma (GKYPL) strategy, offering a comparison with a typical PID in terms of steady-state response [229]. The PID controller is favoured for its straightforward structure, high reliability, and robust stability. However, a notable drawback in dealing with traditional PID controllers lies in the challenge of parameter tuning. The literature provides various techniques for tuning PID parameters, including intelligent techniques, self-tuning algorithms, genetic and other algorithms [230 – 234]. The limitations of PID become apparent when the system faces parametric uncertainty and external disturbances. In practical applications, PID manages uncertainty comprehensively but is constrained to handling only constant parametric uncertainty, necessitating an accurate system model for implementation purposes. This open problem with PID is tackled in further chapters of this thesis.

2.5 Control mechanisms used for 2DoF helicopter system

The Linear Quadratic Regulator (LQR): operates on the receding horizon concept, predicting future outputs at each time step to minimize a global cost function [235]. In the context of a helicopter system, LQR control is developed through linearization [236], allowing interpolation between operating points. However, the limitations of the linearization model, with a focus on local asymptotic stability, render it inadequate for the entire system. In attempts to enhance LQR, integrating an integrator has been explored for optimal helicopter tracking [237]. While effective against parametric uncertainties, it exhibits limited robustness to unstructured uncertainty. Linear Quadratic Gaussian (LQG) controllers, as seen in [238] and [239], offer constant gains for models with uncertainty, promoting robustness. Nevertheless, challenges arise in allocating intuitive covariance matrices, directly dependent on weighting functions, before optimizing gains, making reliance on the controller difficult.

The Back-Stepping Controller: provides a systematic approach to stabilize a reference signal in control system design, utilizing Lyapunov functions [240]. In the helicopter system literature, dual boundary conditional integral back stepping [241] and integral back stepping controllers have been developed, generating reference attitude angles for position control. Despite achieving good dynamic performance, the recursive design and repeated differentiation of back stepping controllers contribute to increased system complexity, limiting applicability for controlling multiple states [242].

Proportional–Integral–Derivative (PID) controllers: are employed primarily to maintain measured process values at set points [243]. In helicopter systems, PID tuning based on fractional-order reference models is utilized for rotor control, with additional designs like cross-coupled PID for tracking purposes [244]. PID control, however, tends to cause oscillations and slower settling times in high-order systems [245]–[248]. While PID is sufficient for stabilizing aero vehicles, it struggles with highly nonlinear systems and lacks efficiency when the system deviates significantly from a fixed set point [249].

Sliding Mode Controllers (SMCs): introduce high-frequency switching terms in control signals to achieve robust disturbance rejection, finite-time stabilization, and precise tracking [250]. Various SMC techniques, including disturbance observer-based controllers, adaptive second-order sliding mode controllers, multivariable integral sliding mode, and terminal sliding mode controllers, have been proposed for

Two Rotor Multirotor Systems (TRMS) [251 – 262]. Despite their advantages, chattering issues in SMCs, causing low control accuracy, mechanical wear, and power circuit heat losses, remain a significant concern [263].

Model Predictive Control (MPC): utilizes process models to minimize an objective function for control action [264]. This involves the explicit use of a system model to predict the future process output within a specified time horizon. The controller calculates a control sequence by employing a receding strategy, displacing the horizon towards the future at each instant. This strategy involves applying the first control signal of the sequence at each step. The application of MPC to helicopter systems is discussed in [265 – 266]. In an alternative approach, a model predictive control (MPC) employs online optimization within a predictive framework, taking advantage of the receding horizon with soft constraints [267] to address exact path tracking problems. Additionally, MPC is implemented for controlling various helicopter applications in [268 – 269]. Literature suggests that the complexity of MPC algorithms leads to longer computation times compared to other controllers. Another significant drawback is the model updating scheme's computational cost, especially if the model is updated frequently. To address this, one may attempt to decrease the model updating frequency; however, this compromise may rapidly degrade the system's stability [270].

H-infinity ($H\infty$): control is employed in attitude control systems to reduce oscillations and regulate rotational moments in the presence of un-modelled dynamics and parametric uncertainties [271]. Furthermore, the weighting functions are adjusted in real-time with respect to the plant's performance during attitude control to achieve the desired and accurate model/state information of the system [272]. In [273], H2/H∞ control is developed to address the balancing and trajectory tracking of UAVs, especially in the presence of disturbances and with minimal response time. Additionally, in [274], an H-infinity observer is designed to safeguard the TRMS against partial and complete failure resulting from actuator and sensor faults.

Neural Network Controllers: The literature extensively showcases the application of neural networks to address highly nonlinear control problems, as indicated in references [275 – 277]. Neural networks operate through numerous simple processing elements at their inputs and outputs within their structure. These processing elements contain internal parameters known as weights, which play a crucial role in modifying the overall network behaviour to achieve optimal outputs for both the controller and its associated plant. Recognizing this advantage, various control and estimation approaches for TRMS (Tilt Rotor Multirotor Systems) are explored in [278], [279]. In [280], a backpropagation, feed-forward neural network model is employed for the nonlinear flight control of a 2-degree-of-freedom helicopter. [281] adopts an indirect adaptive neural network framework to develop a controller for a simulated helicopter system. This controller relies on three interconnected neural networks: the observer, actor, and critic, where the observer network is responsible for state estimation. Additionally, [282] introduces a robust adaptive neural designed to handle nonlinear non affine systems, focusing specifically on single-input single-output (SISO) helicopter systems that involve certain single-channel modes like vertical flight and pitch regulation, as well as special conditions where multiple channels become decoupled.

Fuzzy Logic Controllers (FLCs): have emerged as information processing technologies, particularly for addressing uncertainties [283]. Here, fuzzy controller is employed to address input uncertainties in quadrotor UAVs. This controller minimizes system errors based on inputs and the antecedent fuzzy sets of non-singleton fuzzy logic controllers (FLCs), showcasing robust performance attributed to knowledge-based design. Furthermore, the controller can handle all conceivable uncertainties without requiring a mathematical model of the plant [284]–[286]. In [287 – 288], intelligent adaptive fuzzy controllers are developed for tracking the output of multi-input multi-output systems by approximating errors between actuator and sensor measurements during external disturbances. While type 1 FLCs (T1FLC) find application in various fields, their limitation in dealing with higher-order uncertainties is recognized as a significant drawback [289 – 290]. On the other hand, type 2 FLCs (T2FLC) are more powerful and adept at managing higher-order uncertainties. The increasing number of fuzzy sets has enhanced the degree of freedom for T2FLC in handling uncertainties effectively [291 – 292]. However, the computational intensity of general T2FLCs is heightened due to the type reduction scenario [293 – 295]. Advancements in artificial neural networks have significantly enhanced the capabilities of T2FLC, leveraging their inherent learning ability and uniform approximation of nonlinear systems [296 – 298]. This progress has led to the development of interval type 2 fuzzy neural networks (IT2FNN), widely adopted in control applications, particularly for path tracking in UAVs [299 – 300]. Nonetheless, challenges arise in the computation process due to the burdensome calculation of correlation matrices [301]. Additionally, the drawbacks of conventional membership functions and their limited adaptability to changes in inputs have spurred the demand for innovation in computational intelligence [302].

Hybrid Controllers: In addition to various linear, nonlinear, and intelligent control approaches for diverse systems, the literature extensively discusses the hybridization of controllers [303]–[305]. These methodologies leverage the strengths of multiple controllers to address the individual limitations of each. The integration of optimization techniques such as genetic algorithms (GA) and particle swarm optimization (PSO) with nonlinear controllers is exemplified in [306 – 307]. A GA-based Proportional-Integral-Derivative (PID) approach is devised to adjust control parameters using a performance index as the fitness function [308]. An augmented control scheme, incorporating robust PID-based deadbeat control is employed to modify PID control parameters, mitigating tuning issues and providing precise control gain values [309]. In [310], a comprehensive evaluation of various conventional and intelligent control methods, including fuzzy logic and GA-based approaches, is implemented on the Tilt Rotor Multirotor System (TRMS). Furthermore, [311] explores sliding mode control (SMC) hybridized with fuzzy logic, comparing it with fuzzy sliding and fuzzy integral sliding controllers. [312] utilizes a parallel distribution of fuzzy Linear Quadratic Regulator (LQR) controllers for individual axis control of the system. Similarly, the design of the robust adaptive fuzzy controller (RAFC) incorporates the gradient descent algorithm [313]. Moreover, [314] highlight the elimination of chattering in SMC when augmented by a fuzzy controller, achieving trajectory tracking for a quadrotor. Additionally, various controllers, including fuzzy-sliding hybrid [315], adaptive sliding [316], LMI-based observer [317], and fuzzy controllers using non-monotonic Lyapunov functions [318], have been designed to enhance helicopter performance in the presence of disturbances and uncertainty.

2.6 Identified research gaps

The literature frequently highlights the advantages of optimization algorithms, despite their lack of theoretical foundation. However, in order to fully harness the potential of algorithms, several common challenges need resolution:

- It is essential to recognize that the algorithm's complexity and the complexity of parameter relations are directly influenced by the number of algorithm parameters, thereby complicating the performance analysis.
- Scientific literature acknowledges the significance of tuning parameters. Effectively applying metaheuristics to real-world problems requires the identification of an optimal starting parameter setting, a task that is time-consuming and challenging.
- Theoretical research on the analysis of landscapes (i.e., the topological structure guiding the search) for various optimization problems indicates that distinct issues and even different instances of the same issue involve different topologies. Specially while implementing such techniques on real-time world problems.
- Researchers should focus on the integration of classical and intelligent controllers while performing constraint parameter optimization. Some studies suggest that combining the two results in a faster convergence rate.

Chapter 3. Two degree of freedom (2DoF) systems

3.1 Overview

An exhaustive examination of various strategies for position tracking and balancing control is presented in literature review. These cutting-edge solutions follow two primary approaches. The first approach, known as cross-coupling control, considers the control loops of all machine axes simultaneously, addressing the control objective as a path deviation correction problem. The second approach, termed individual axis control, handles each axis independently. Both approaches involve adjusting operating parameters (such as pitch and yaw movement, plate angle, etc.) or achieving precise position tracking ensuring that the deviation between measured and desired paths is minimal, and specific constraints are met. It's noted that both approaches are interdependent in developing the control action. Additionally, in dealing with individual axis control, with a few exceptions like varying-gain, intelligent and optimization-based control considering unknown noises, most nonlinear control techniques for 2DoF systems do not incorporate parameter estimation in system dynamics. This limitation could be addressed by incorporating nonlinear and adaptive control algorithms into individual axis control architectures, especially since the challenge of nonlinear position control for flexible axes has been extensively explored. Numerous studies in the literature employ techniques from nonlinear control theory, focusing on position and balancing control. In-depth overview of these techniques, covering various aspects of control achievement has been discussed in previous chapter. Considering the research gaps identified after this study, new control mechanism is designed for real time hardware system. Before implementing these control mechanisms, understanding of mathematical model of such hardware is must. Subsequent sections is all about the mathematics involved in developing the hardware model of 2DoF ball balancer and 2DoF helicopter.

3.2 Modelling of 2DoF ball balancer model

In this section, the nonlinear model of two-degrees-of-freedom ball balancer system is discussed. The 2DOF ball balancer [319] is a typical test bench problem for attaining position tracking and balancing control with visual servo control. It comprises one quadrangular metallic plate fixed using gimbal joints with center. This plate is able to move using DC micro motors. This movement has two directions – X and Y axis. To balance the 2DOF ball balancer in these directions, rotational motion using Faulhaber series DC micro motors is used [320].

The aim is to balance the ball on quadrangular plate without falling off. This configuration of ball and plate has access to four degrees of freedom which are controlled using two different actuators. That’s why it is referred to as a two degrees of freedom i.e. X-axis and Y-axis. A part of this 2DOF ball balancer system contains a digital camera placed exactly above this plate, which is able to capture the images of these X & Y co-ordinates to track and help control the position of ball on the plate. A real time working model of this system is shown in fig. 3.1 and the working relation between different components is explained in fig. 3.2.

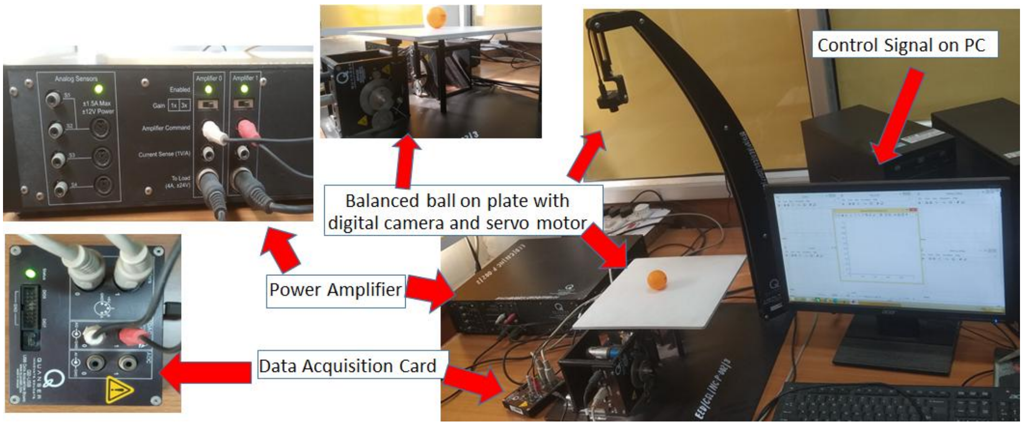


Fig. 3.1 Real time working laboratory setup of 2DoF ball balancer system

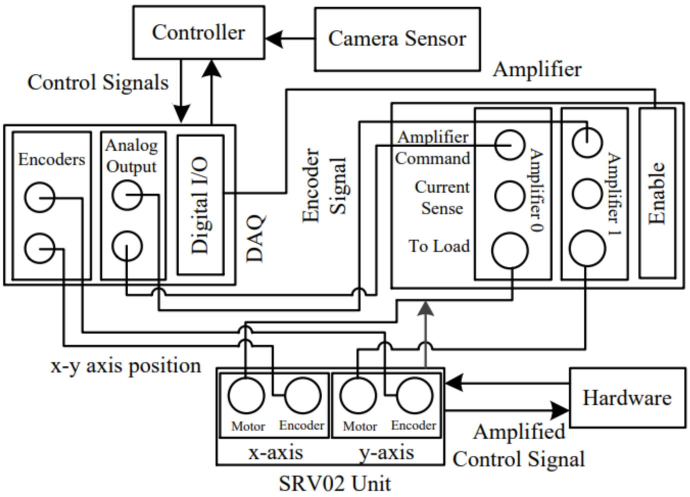


Fig. 3.2 Working Structure between different components of 2DoF ball balancer system

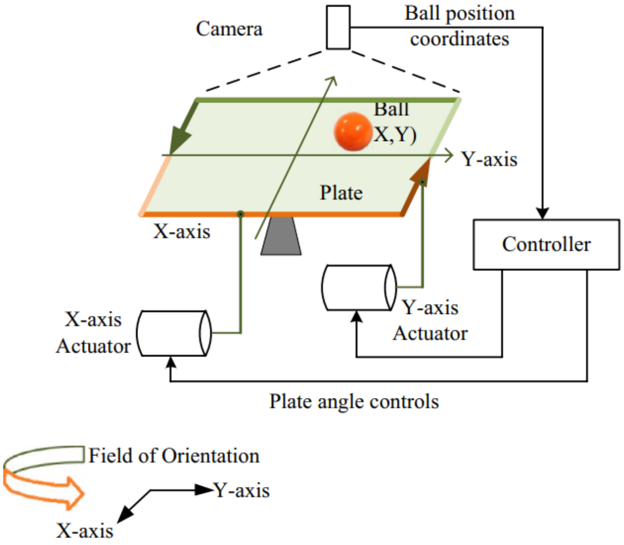


Fig. 3.3 Schematic diagram of a two degree of freedom ball balancer system.

A vision algorithm reads the co-ordinates of ball, through the image tracked using digital camera and provides information to the controller. Image processing is used as the feedback signal through personal computer in this mechanism. This process helps to adjust the angle of plate through servo motors along the X and Y axis. A typical representation of ball and plate system is given in figure 3.3, which signifies the X & Y co-ordinates with ball and plate are being controlled using a camera.

3.2.1 Transfer function

The arrangement of two degrees of ball balancer system in open loop configuration is illustrated using transfer function modelling in Figure 3.4. Dynamic relation between servo voltage input and load angle output are represented by plant W_s(s) while W_{ss}(s) represents the dynamic relation between servo gear angle load and ball position dynamics. Two rotary servo based units are responsible for the control operation of 2DOF ball balancer system which makes these servo devices symmetrical to the plate arrangement in order to provide ease in real time operation. Both device dynamics are similar. Hence, 2DOF ball balancer system can be treated as a two decoupled beam and ball system. In one dimension, transfer function can be computed as:-

$$W(s) = W_s(s) * W_{ss}(s) \tag{3.1}$$

Where,

$$W_{ss}(s) \frac{A(s)}{\beta_1(s)} \; and \; W_s(s) = \frac{\beta_1(s)}{V_m(s)} \tag{3.2}$$

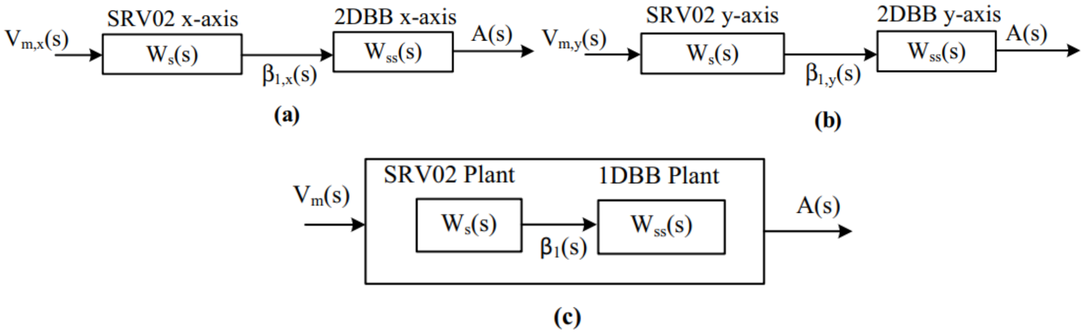


Fig. 3.4 2DOF ball balancer transfer function for (a) X-axis servo angle (b) Y-axis servo angle and (c) one dimensional ball balancer (1DBB) open loop transfer function

3.2.2 Mathematical equations for plate movement

A 2DOF ball balancer system is picturized in figure 3.5 by using its free body diagram. The ball movement due to the different plate angle can be formulated using equations of motion. Due to this change in angle, the force experienced by the ball can be explained using law of inertia as:-

$$\ddot{A}(t)\, M_{\mathrm{ball}} = F_{\mathrm{x,t}} - F_{\mathrm{x,i}} \tag{3.3}$$

Where, M_{ball} represents mass of ball, and A represents displacement of ball.

$F_{\mathrm{x,i}}$ and $F_{\mathrm{x,t}}$ are force of inertia and the translational force due to gravity respectively, in the direction of X axis along with plate.

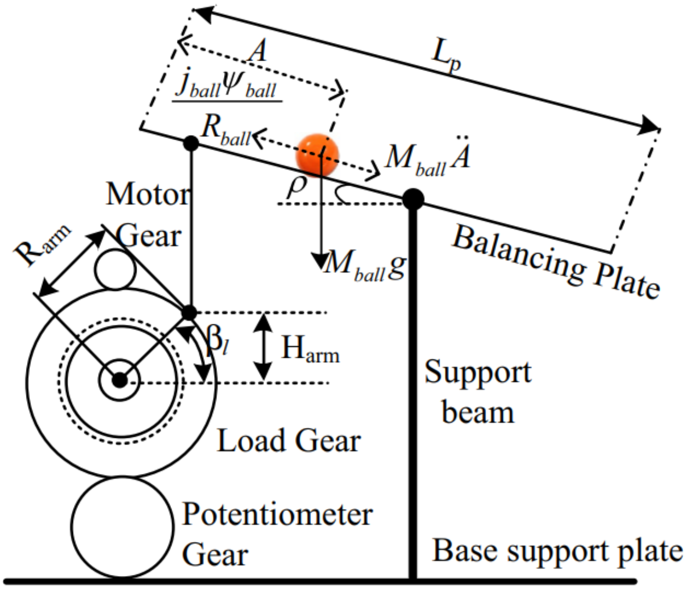


Fig. 3.5 Free body diagram of 2DOF ball balancer system.

In these calculations, friction damping and viscous constants are not taken into consideration. The mathematical equations used herein are derived from Quanser user manual for 2 DOF ball balancer model [319]. The value of translational force and force of inertia are calculated as:-

$$F_{\mathrm{x,t}} = \mathrm{G}*\sin\rho(\mathrm{t})*M_{\mathrm{ball}} \tag{3.4}$$

$$F_{\mathrm{x,i}} = \tau_{ball}/R_{ball} \tag{3.5}$$

Where, τ_{ball} is torque experienced and R_{ball} is representing radius of the ball.

This makes it clear that if j_{ball} represents inertia of the ball then motion equation of ball balancer system can be written by linear acceleration as:-

$$\ddot{A}(t) = \{\mathrm{G}*\sin\rho(\mathrm{t})*M_{\mathrm{ball}}*R_{ball}{}^{2}\} / \{R_{ball}{}^{2} * M_{\mathrm{ball}} + j_{ball}\} \tag{3.6}$$

3.2.3 Servo angle calculations

In this subsection, the position of ball around the servo load angle are representing motion and time dynamics. These dynamics helps computing the behavior of 2DOF ball balancer system. Servo angle has a relation with respect to the beam of the 2DOF ball balancer system, which is given by:

$$\sin\rho(\mathrm{t}) = \sin\beta_l(\mathrm{t})*2*R_{arm} / L_{\mathrm{p}} \tag{3.7}$$

where, length of plate in 2DOF ball balancer system is given by L_{p} and distance between shaft of SRV02 output gear and coupled joint is given by R_{arm}.

Considering servo angle to be Θ_1, in order to calculate the motion and time dynamics for representing motion equations, we need to linearize dynamics variable around Θ_1=0. Relationship between servo angle and plate angle can be explained mathematically as:-

$$\ddot{A}(t) = \{G^{*} \sin\beta_l(t)^{*}2 * M_{\text{ball}}{}^{*}R_{ball}{}^{2} * R_{arm}\} / \{R_{ball}{}^{2} * L_{\text{p}} * M_{\text{ball}} + j_{ball}\} \tag{3.8}$$

When we linearize the equation of motion, sinβ_l(t) would be replaced by β_l(t). Final equation for movement of plate for 1DBB is:-

$$\ddot{A}(t) = \{G^{*}\beta_l(t)^{*}2 * M_{\text{ball}}{}^{*}R_{ball}{}^{2} * R_{arm}\} / \{R_{ball}{}^{2} * L_{\text{p}} * M_{\text{ball}} + j_{ball} * L_{\text{p}}\} \tag{3.9}$$

3.3 Problem formulation for 2DoF ball balancer system

The 2DoF ball balancer's open-loop block diagram is presented as a decoupled model in which the x-axis servo has no impact on the y-axis response, as depicted in figure 3.4. In light of this, figure 3.6 illustrates the control model for the x-axis of the SRV02 integrated with the ball balancer system. The ball balancer block diagram elucidates control in two loops: the first loop for the SRV02 motor model and the second for the 1D ball balancer. The inner loop details the SRV02 controller ($Z\_(s)$) for position control of the D.C. series motor, estimating the required voltage to track the desired angle of the ball.

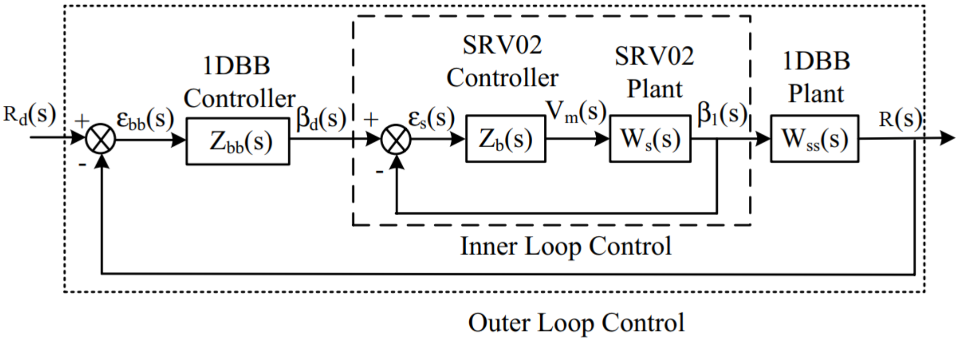


Fig. 3.6 Block diagram with closed loop control scheme of a two degree of freedom ball balancer system.

For any given controller, the discrepancy between the desired and real-time ball positions in the ball balancer model serves as the input for the controller. According to system specifications, the plate dimensions are 27.5cm length X 27.5cm width, restricting the ball's movement within a length of 13.75cm, assuming the ball is positioned at the plate's centre. However, this length causes the ball to reach the plate corners, and the research objective is to stabilize the ball in the centre. To achieve this, a range of 0-6.875 cm has been defined to confine the ball within the middle of the plate, with a reference of 3 cm chosen to keep the ball closer to the plate centre. Consequently, a 0.08 Hz square wave with an amplitude of 3 is considered as input for the system [321]. The inner loop is monitored by controlling the SRV02 position through a proportional motor gain, while external controllers can be designed for outer loop control.

3.4 Modelling of 2DoF helicopter model

3.4.1 Inter-relation between pitch and yaw axis

To examine the functionality and control capabilities, the helicopter's main and tail rotors are discussed. The main rotor generates vertical thrust, allowing the model to pitch around the horizontal axes (X), while the tail rotor produces horizontal thrust, enabling yaw movement around the vertical axis (Y). These twin-rotor DC motors are driven by voltage inputs, yielding torque $\tau 1$ for the main rotor and torque $\tau 2$ for the tail rotor, as given by Quanser educational solutions [322]. The interconnections among different subsystems of the helicopter system are illustrated in figure 3.8, where dotted lines indicate cross-coupling between the two planes. The outputs of the mechanics block represent the pitch ψ and yaw φ angles for the vertical and horizontal planes, respectively.

Furthermore, the system's mathematical modelling is based on DC motor modelling for the main and tail rotors, along with mechanics modelling for the horizontal and vertical planes. Previous research has extensively established mathematical models for the pitch and yaw rotors of helicopter systems. Therefore, this study primarily concentrates on comprehending the operation of pitch and yaw motors and assessing them to ensure the achievement of the desired control objectives.

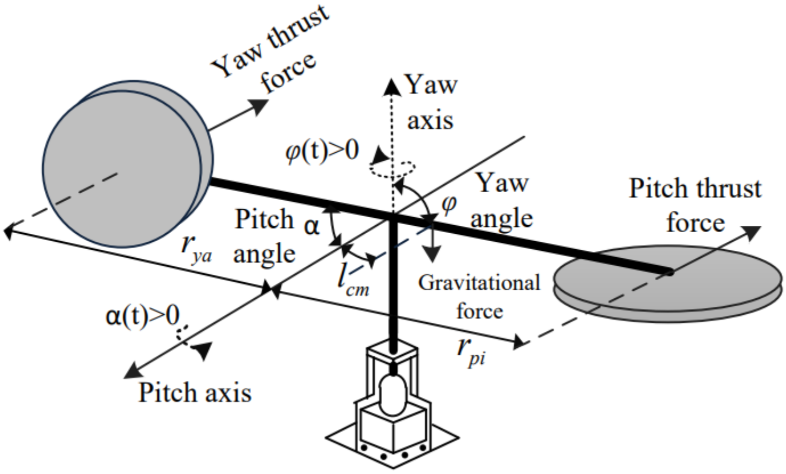


Fig. 3.7 Free body diagram of a 2DoF helicopter model.

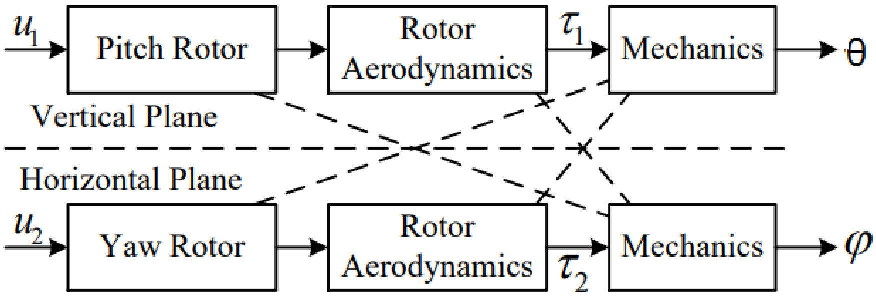
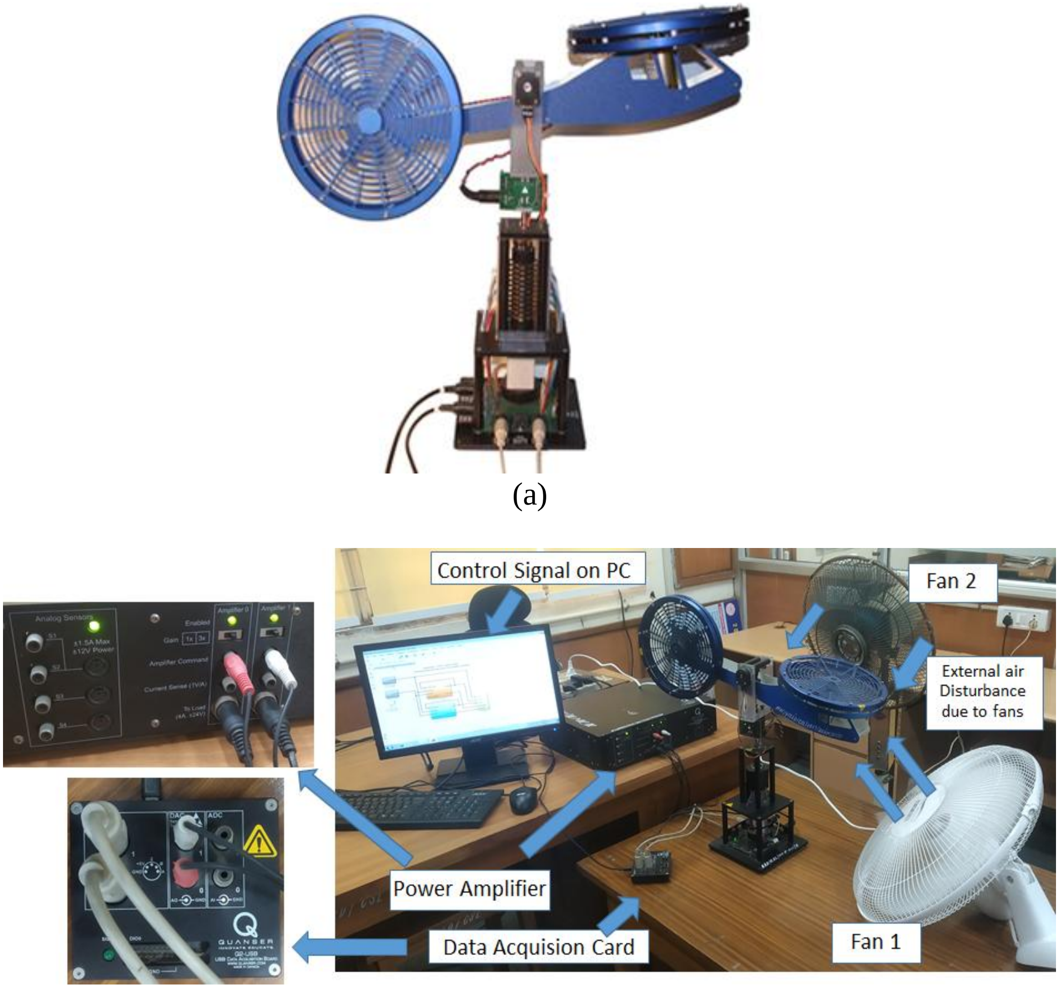


Fig. 3.8 Cross-coupling effect of pitch rotor and yaw rotor in 2DoF helicopter model

Motors used are DC type, one is for pitch movement and other one is for yaw movement. The physical modelling also includes two different propellers. These are called front propeller and back propeller. Two different encoders are also used – one for pitch and second one is for yaw movements. Body of nonlinear helicopter system is kept on one stationary fixed base platform. The 1st DC motor i.e. Pitch motor is operating the movement of front propeller, which further handles the altitude of nonlinear helicopter nose about its pitch axis. This further results in the controlling of pitch angle, denoted by θ. The 2nd DC motor i.e. yaw motor is operating the horizontal movement of nonlinear helicopter about its yaw axis. Which, further results in controlling of yaw angle, denoted by Ψ. The yaw encoder and the pitch encoder are used to measure the resultant yaw angle and pitch angle respectively. Using a slip-ring, in the form of signals, the measured angles then transferred further to provide the desired pitch and yaw angles.



(a)

(b)

Fig. 3.9 Real time experimental (a) Body and (b) setup of 2 DOF helicopter system under external disturbances.

In literature, ample of work has been carried out on laboratory setup of helicopter model but all this work tends to lag behind when it comes to the working of real time model with real time environment conditions. Especially when the weather conditions are unfavourable. Such unfavourable weather conditions are developed for 2DoF helicopter model in laboratory itself. Two high speed fans are used to create high turbulence in laboratory itself. This arrangement will be proved to be more helpful in validating the control scheme. A real time working model of helicopter system, with two high speed fans, is depicted in figure 3.9(a) and figure 3.9(b). The following constraints are considered during the modelling phase:

- Pitch angle is denoted by θ. If $\theta = 0$, it means the helicopter is horizontal.
- When the head of helicopter system moves upwards and downwards, the pitch angle increases and decreases respectively.
- If pitch angle moves positively i.e. $\theta(t) > 0$, it means the helicopter body moving in vertical upward direction with respect to pitch axis.
- If pitch angle moves negatively i.e. $\theta(t) < 0$, it means the helicopter body moving in vertical downward direction with respect to pitch axis.
- Yaw angle is denoted by Ψ. If $\Psi = 0$, it means the body is neither moving clockwise nor moving anticlockwise.
- If the yaw angle increases i.e. $\Psi(t) > 0$, it means the body moves in horizontal direction and rotates counterclockwise with respect to yaw axis.
- The force produced due to pitch thrust is denoted by Fp. If $Fp > 0$, it means pitch angle increases i.e. $\theta > 0$.
- The force produced due to yaw thrust is denoted by Fy. If $Fy > 0$, it means yaw angle increases i.e. $\Psi > 0$.

3.5 Problem formulation for 2DoF helicopter model

Distance of both the propellers are measure by taking canter of body as the reference which may determine centre of mass of nonlinear helicopter body. It gives r_p and r_y the distance of 1st and 2nd propeller from the centre respectively. In continuation of modelling of nonlinear helicopter, the transformation matrix discussion would be the next step, as the centre-of-mass of nonlinear helicopter is interrupted from central axis [323 – 324].

From this, rotation and translation in this transformation matrix are taken as:
T_{cm} : Translation about centre of mass
R_{Ψ} : Rotation about the yaw axis
R_{θ} : Rotation about the pitch axis

Which further gives mathematical equivalent of transformation matrix as:

$$T_{cm} = \begin{bmatrix} 1 & 0 & 0 & l_{cm} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.10}$$

$$R_{\theta} = \begin{bmatrix} \cos(-\theta) & 0 & \sin(-\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(-\theta) & 0 & \cos(-\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.11)$$

$$R_{\Psi} = \begin{bmatrix} \cos(-\Psi) & -\sin(-\Psi) & 0 & 0 \\ \sin(-\Psi) & \cos(-\Psi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.12)$$

$$T_0 = R_{\theta} * R_{\Psi} * T_{cm} \quad (3.13)$$

Where T_0 is resultant transformation matrix [323 – 234]. Then the lagrangian energy approach is used and one nonlinear equation of motion is achieved. In the process, the lagrangian mathematical equation (LME) considered is:

$$\text{LME} = \text{K} - \text{P} \quad (3.14)$$

Where, P denotes potential energy and K denotes kinetic energy for the nonlinear helicopter system. The kinetic energy in current discussion is the combination of three different energies. These three energies are due to three different factors in running helicopter - pitch, yaw and translation, due to movement of center of mass of nonlinear helicopter system. R_p denotes the rotational kinetic energy in vertical direction i.e. pitch axis. R_y denotes the rotational kinetic energy in horizontal direction i.e. yaw axis. T_t denotes the translation kinetic energy due to movements in center of mass. Then total kinetic energy is given by:

$$\text{R} = R_p + R_y + T_t \quad (3.15)$$

Kinetic energy in horizontal direction and kinetic energy in vertical direction depends upon the moment of inertia in the pitch axis ($Jeq{,}p$) and moment of inertia in yaw axis ($Jeq{,}y$) respectively. Whereas the translational kinetic energy totally depends upon the length of center of mass, denoted by lcm. Hence, the equivalent rotational kinetic energy due to the vertical movement and horizontal movement of nonlinear helicopter system is given in the following equation:

$$R_p = (\frac{d}{dt}\,\theta(\text{t}))^2 * Jeq{,}p * 0.5 \quad (3.16)$$

$$R_y = (\frac{d}{dt}\,\Psi(\text{t}))^2 * Jeq{,}y * 0.5 \quad (3.17)$$

The translational kinetic energy is derived considering the nonlinear dynamics of the helicopter system and further given with the help of following equation:

$$T_t = \frac{1}{2}\, m_{heli}\, l_{cm}^2 (\,\dot{\theta}^2 + \ \dot{\Psi}^{\,2}\, cos^2(\theta)) \quad (3.18)$$

Putting eq. (3.16-3.18) in eq. (3.15), we may calculate the total kinetic energy of nonlinear helicopter system. Furthermore, the total potential energy experienced on the system, due to clear effect of the gravitational force is calculated as:

$$\text{P} = m_{heli} * \text{g} * \ l_{cm} * \sin(\theta) \quad (3.19)$$

3.5.1 Calculation of ABCD parameters

Now in order to observe the nonlinear dynamics of the system, the generalized co-ordinates $[\theta, \Psi, \dot{\theta}, \dot{\Psi}]^T$ which further decides the behavior of nonlinear system are determined using the Lagrangian method. The nonlinear system parameters referred for this whole calculation are specifically mentioned in table 1 in detail, with specific values. By using this lagrangian method and substituting eq. (3.15-3.19) in eq. (3.14), the motion (both vertical and horizontal) equation for nonlinear helicopter is given as:

$$(m_{heli}\, l_{cm}^2 + J_{eq,p})\, \ddot{\theta} = K_{pp}\, V_{m,y} + K_{py}\, V_{m,p} - m_{heli} g l_{cm} \cos\theta - B_p\, \dot{\theta} - m_{heli}\, l_{cm}^2 \sin\theta\cos\theta\dot{\Psi}^2 \tag{3.20}$$

$$(J_{eq,y} + m_{heli}\, l_{cm}^2 \cos\theta^2)\ddot{\Psi} = K_{yp}\, V_{m,p} + K_{yy}\, V_{m,y} - B_y\dot{\Psi} + 2m_{heli}\, l_{cm}^2 \sin\theta\cos\theta\dot{\Psi}\dot{\theta} \tag{3.21}$$

For performing a state feedback control using LQR, the linearized model is expected to be used [325]. This will represent the dynamics of any system more clearly. In order to perform this, the nonlinear system will further linearized by equating $\theta = 0, \Psi = 0, \dot{\theta} = 0$ and $\dot{\Psi} = 0$ around origin and putting in equations (3.20) and (3.21). The obtained linearized equations are:

$$\ddot{\theta} = \frac{V_{m,y} * K_{py}}{(m_{heli} * l_{cm}^2 + J_{eq,p})} + \frac{V_{m,p} * K_{pp}}{(m_{heli} * l_{cm}^2 + J_{eq,p})} - \frac{m_{heli} * l_{cm}^2 + B_p\, \dot{\theta}}{(m_{heli} * l_{cm}^2 + J_{eq,p})} \tag{3.22}$$

$$\ddot{\Psi} = \frac{V_{m,y} * K_{yy}}{(J_{eq,y} + m_{heli}\, l_{cm}^2)} + \frac{V_{m,p} * K_{yp}}{(J_{eq,y} + m_{heli}\, l_{cm}^2)} - \frac{B_y \dot{\Psi}}{(J_{eq,y} + m_{heli}\, l_{cm}^2)} \tag{3.23}$$

The control system operating within the nonlinear helicopter system incorporates a gravitational disturbance, leading to pitch and yaw state errors. To mitigate these errors, the inclusion of two integrators is essential. Designating θ_d and Ψ_d as the desired set points for pitch and yaw angles, respectively, the relevant integrals are expressed through the following equations:

$$\gamma = \int (\Psi_d - \Psi) dt \tag{3.24}$$

$$\alpha = \int (\theta_d - \theta) dt \tag{3.25}$$

For the following state-space representation:

$$\dot{x} = Ax + Bu \tag{3.26}$$

Where, x is new state vector and formulated as:

$$x = [\theta, \Psi, \dot{\theta}, \dot{\Psi}, \dot{\alpha}, \dot{\gamma}]^T \tag{3.27}$$

and output is shown as:

$$y = Cx + Du \tag{3.28}$$

While ABCD being the system matrix, the input matrix u is:

$$u = [V_{m,p}, V_{m,y}]^T \tag{3.29}$$

With the aforementioned discussion in mind, the state space representation of the nonlinear helicopter system can be established by reformulating equations (3.22 – 3.23) into the state space model, as follows:

$$\begin{bmatrix} \dot{\theta} \\ \dot{\Psi} \\ \ddot{\theta} \\ \ddot{\Psi} \\ \dot{\alpha} \\ \dot{\gamma} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{-B_p}{J_{eq,p} + m_{heli}\, l_{cm}^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{-B_y}{J_{eq,y} + m_{heli}\, l_{cm}^2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \theta \\ \Psi \\ \dot{\theta} \\ \dot{\Psi} \\ \dot{\alpha} \\ \dot{\gamma} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{K_{pp}}{J_{eq,p} + m_{heli}\, l_{cm}^2} & \frac{K_{py}}{J_{eq,p} + m_{heli}\, l_{cm}^2} \\ \frac{K_{yp}}{J_{eq,y} + m_{heli}\, l_{cm}^2} & \frac{K_{yy}}{J_{eq,y} + m_{heli}\, l_{cm}^2} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_{m,p} \\ V_{m,y} \end{bmatrix} \tag{3.30}$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \Psi \\ \dot{\theta} \\ \dot{\Psi} \\ \dot{\alpha} \\ \dot{\gamma} \end{bmatrix} \tag{3.31}$$

The ABCD matrix are calculated by substituting values from table 1 into equation (3.30 – 3.31) and given as:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -9.2751 & 0 & 0 & 0 \\ 0 & 0 & 0 & -34955 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad ; \qquad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 2.3667 & 0.0790 \\ 0.2410 & 0.7913 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad\qquad D = 0$$

The ABCD parameters acquired will be applied within the MATLAB software's simulation platform, contributing to formulation of benchmark model for our existing problems.

3.6 Conclusion

In this chapter, the mathematical models of two-degree-of-freedom systems, namely the helicopter model and the ball balancer, were thoroughly explained. The chapter also provided a detailed description of the laboratory setup and formulated the problem for the implementation of both traditional and intelligent controllers. These models will serve as the foundation for applying various control strategies in the subsequent chapters.

Chapter 4. Optimized classical control of 2DoF systems

4.1 Classical PID controller for 2DoF systems

The PID controller is widely adopted in industrial control applications due to its reliability and straightforward implementation. Classical methods for designing and fine-tuning PID parameters (Kp, Ki, $and\ Kd$) are well-established but require expertise and practice. These traditional techniques involve a trial-and-error process for setting a starting point and refining parameters to achieve desired efficiency [326]. Metaheuristic strategies, given their dynamic nature, present a viable alternative. Manual calculation of PID gain values often leads to significant errors, especially when dealing with diverse parametric and external uncertainties. Consequently, automatic PID gain tuning becomes necessary, a task accomplished through various metaheuristic algorithms. To implement this control scheme on 2DOF system, the close loop control system strategy is preferred. Whole control scheme is demonstrated using block diagram in figure 4.1.

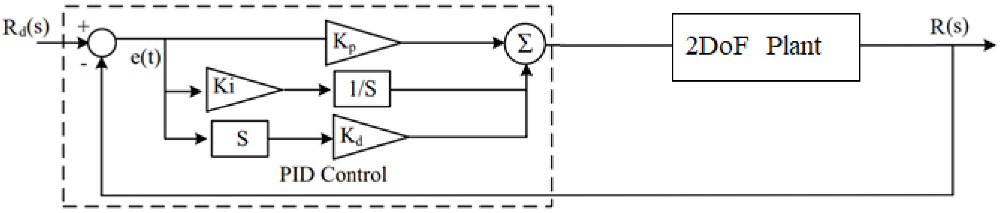


Fig. 4.1 Close loop control system with PID constraints

Ideally, PID controller will deliver some output which is the outcome of error (e) between the output processed value of the whole system and set point given by the user. If we consider R_d(s) to be the set point and R(s) is assumed to be the process value then, this error may be written as:-

$$e(t) = R_d(s) – R(s) \tag{4.1}$$

Furthermore, the PID control in time domain is essential to be computed in order to attain the initial operating gain for the controlling of 2DoF system, which can be given as:-

$$\beta(t) = K_{p,initial}\,(R_d(t) – R(t)) + K_{d,initial}\left(\frac{d}{dt}R_d(t) - \frac{d}{dt}R(t)\right) + K_{i,initial}\int (R_d(t) – R(t))\,dt \tag{4.2}$$

where, $K_{p,initial}$, $K_{d,initial}$ and $K_{i,initial}$ are initial values of proportional, integral and derivative gains constant values respectively. 2DoF plant may refer to many systems as discussed in literature but for this work only 2DoF helicopter system and 2DoF ball balancer system are considered. Block diagram of these two systems with classical controller are discussed in the subsection one by one.

Block diagram given in figure 4.2 illustrates the classical PID based control of a ball balancer system. Here two loops are used to control the servo angle of plate. If control of balancer system is need to be achieved, the outer loop of the figure 4.2 is to be considered, where no servo dynamics are considered initially. Observing this, the Laplace equivalent of equation 4.2 is written in equation no (3) as:-

$$\beta(s) = \left(K_{p,initial} + \frac{K_{i,initial}}{s} + K_{d,initial} * s\right)[R_d(s) - R(s)] \tag{4.3}$$

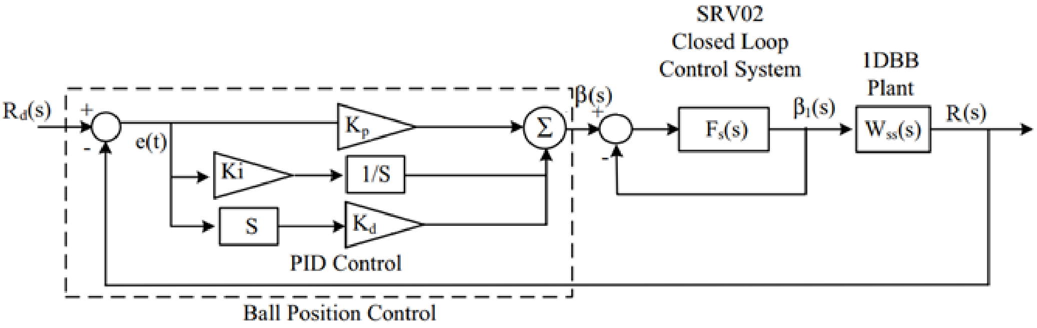


Fig. 4.2 Block diagram of PID control scheme of 2DoF ball balancer system

Plant equations of this 2DoF system has already been discussed in previous chapter of this thesis. In this sub section of current chapter, focus will be on calculating the constraint parameters of the controller. Since, action of no servo dynamic force is observed, it clearly means the required load points are equal to the estimated ones. From here, we can calculate closed loop equations after the substitution of 1DBB system with outer loop controller (in equation 3) as:-

$$\frac{R(s)}{R_d(s)} = \frac{(K_{p,initial} * s + K_{i,initial} + K_{d,initial} * s^2) * K_G}{s^3 + (K_{p,initial} * s + K_{i,initial} + K_{d,initial} * s^2) * K_G} \tag{4.4}$$

Where, constant gain for model is given by K_G . PID gains may be calculated using third order prototype equations for which we need to consider:-

$$(s^2 + 2s\omega_n\Im + \gamma_n^2)*(s + p_0) \tag{4.5}$$

Where $\Im$ represents the damping ratio, ω_n represents the natural frequency and p_0 refers to the pole location. After expansion, this third order equation may be re-written as:-

$$s^3 + (2\omega_n\Im + p_0)\, s^2 + (\omega_n^2 + 2\omega_n\Im p_0)s + \omega_n^2 p_0 \tag{4.6}$$

Following the equation no. 4.6, the third order characteristic equation given in equation no. 4.4 may be given as:-

$$s^3 + (K_G * K_{d,initial})* s^2 + (K_G * K_{p,initial})*s + (K_G * K_{i,initial}) \tag{4.7}$$

Equating equation no. (4.6) with equation no. (4.7), we may conclude the following:-

$$K_G * K_{i,initial} = \omega_n^2 p_0 \tag{4.8}$$

$$K_G * K_{p,initial} = \omega_n^2 + 2\omega_n\Im p_0 \tag{4.9}$$

$$K_G * K_{d,initial} = 2\omega_n\Im + p_0 \tag{4.10}$$

After solving the equation no. (4.8-4.10) again, we may conclude that the initial gains for the PID controller may be computed as following:-

$$K_{i,initial} = \frac{\omega_n^2 p_0}{K_G} \tag{4.11}$$

$$K_{p,initial} = \frac{\omega_n^2 + 2\omega_n \Im p_0}{K_G} \tag{4.12}$$

$$K_{d,initial} = \frac{2\omega_n \Im + p_0}{K_G} \tag{4.13}$$

Equations (4.11-4.13) will be useful in the calculation of initial gain values. The procedure where we calculates these gain values, will create a large error, especially when we are operating under various external disturbances. This creates some scope for the automatic tuning of these PID parameters. This automatic tuning is achieved using Teaching Learning Based Optimization (TLBO) in current chapter.

Similarly, the gain parameters of PID controller are also considered to control a two degree of freedom (2DoF) helicopter system. For this a closed loop control mechanism is developed for handling the cross-coupling between the pitch axis and yaw axis of helicopter model. Block diagram shown in figure 4.3 illustrates control of 2DoF helicopter model:-

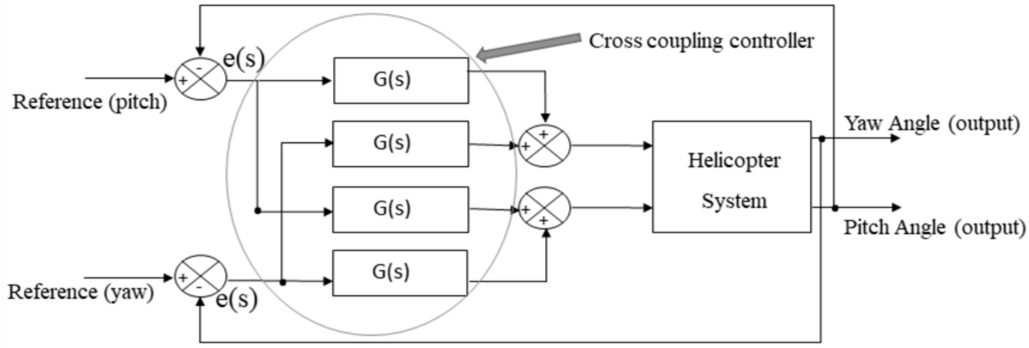


Fig. 4.3 Block diagram of cross-coupled controller for 2DoF helicopter system

The control model for the cross coupled yaw and pitch axis of the twin rotor system integrated with unmanned helicopter system is illustrated in figure 4.3. The parameters of controller G(s) are being handled using optimization algorithm. Two degree of freedom (2DOF) helicopter system is handled using variety of control techniques ranging from traditional controllers like PID & LQR to optimization techniques involving heuristic and metaheuristics (refer section 2.4). For any problem having some large numbers of constraint, it provides a near optimum solution but it is matter-of-fact that a minimum change in algorithm parameter may change the overall effectiveness of algorithm. Using such algorithm during the unfavourable weather conditions, may affect the working of helicopter system. This attracts present work, to use an algorithm whose results on helicopter model is free from the parameters of algorithm. Using such an algorithm will definitely provide a more stable control environment for cross coupled design of 2DOF unmanned helicopter system. In the figure, the controller ‘G(s)’ is optimized using proposed algorithm and implemented on error ‘e(s)’ signal, which is obtained by subtracting the pitch and yaw angle signal from the pitch and the yaw reference signal respectively. Further, these optimized parameters of controller are handling the working of hardware system of helicopter model and the whole process continues until a more stable waveform for the pitch angle (θ) and the yaw angle (Ψ) is achieved.

4.1.1 Defining cost functions

The cost (or objective) functions IAE (Integral of Absolute Error), ITSE (Integral of Time-weighted Squared Error), ITAE (Integral of Time-weighted Absolute Error), IE (Integral of Error), MSE (Mean Square Error) and ISE (Integral of Squared Error) find widespread use in the realms of control theory and system optimization. Their primary role is to quantify the performance of a control system and aid in the adjustment of controller parameters. Presented below is a concise summary of each:

1. IAE (Integral of Absolute Error):

Application: Computes the cumulative sum of absolute error values over time.

Objective: Emphasizes the reduction of error magnitudes.

$$\textbf{IAE} = \int |\boldsymbol{e}(\boldsymbol{t})|\, \boldsymbol{dt} \tag{4.14}$$

2. ITSE (Integral of Time-weighted Squared Error):

Application: Determines the sum of squared errors over time, with each error term weighted by time.

Objective: Prioritizes the minimization of squared errors, assigning greater importance to persistent errors.

$$\textbf{ITSE} = \int \boldsymbol{t}\boldsymbol{e}^{\boldsymbol{2}}(\boldsymbol{t})\, \boldsymbol{dt} \tag{4.15}$$

3. ITAE (Integral of Time-weighted Absolute Error):

Application: Similar to ITSE but focuses on the integral of absolute error, incorporating time weighting.

Objective: Highlights the reduction of error magnitudes over time.

$$\textbf{ITAE} = \int \boldsymbol{t}|\boldsymbol{e}(\boldsymbol{t})|\, \boldsymbol{dt} \tag{4.16}$$

4. IE (Integral of Error):

Application: Computes the cumulative sum of errors over time without squaring or considering absolute values.

Objective: Offers a general measure of overall error, without accentuating positive or negative deviations.

$$\textbf{IE} = \int \boldsymbol{e}(\boldsymbol{t})\boldsymbol{dt} \tag{4.17}$$

5. ISE (Integral of Squared Error):

Application: Calculates the sum of squared errors over time without time weighting.

Objective: Concentrates on minimizing squared errors without accounting for the duration.

$$\textbf{ISE} = \int \boldsymbol{e^2(t)\, dt} \tag{4.18}$$

6. MSE (Mean of Square Error):

Application: MSE is widely used in various fields, including control theory.

Objective: It assesses the average squared error over time, making it sensitive to both the magnitude and duration of errors.

$$\textbf{MSE} = \mathbf{(1/n)}\boldsymbol{\frac{1}{n}} \sum_{i=0}^{n}(\mathbf{yi} - \mathbf{pi})\text{\textasciicircum}\mathbf{2} \tag{4.19}$$

These cost functions play a critical role in evaluating a control system's performance during operation. Through the analysis of these metrics, engineers and researchers can refine controller parameters to enhance the system's responsiveness and minimize errors, ensuring improved control and stability. The choice of a specific cost function depends on the unique requirements and characteristics of the controlled system, which are further used in this work.

4.2 TLBO algorithm based control mechanism

4.2.1 TLBO algorithm

Teaching-Learning-Based Optimization (TLBO) is an optimization algorithm inspired by natural processes, specifically the dynamics of teaching and learning within a classroom. First introduced by Rao et al. [50], TLBO is a population-based approach crafted for tackling optimization problems. We will need two controlling parameters – number of generations and population size for the implementation of this algorithm. TLBO is based on the "teaching pattern" followed in an academic class to improve the knowledge of students. Working of TLBO method is described in two different phases. These are – "The Teaching Phase" and "The Learning Phase". These two phases are composed of different algorithm and working of each phase is discussed in detail herewith.

"The Teaching Phase"

In this phase, learning of students in a class is demonstrated with the help of a teacher. In this portion of TLBO algorithm, a teacher focus on increasing the mean results of all the students available in the class for any specific subject which is being taught by him/her, totally depending upon the capacity of that particular teacher. Considering, I stands for iteration, 'n' is student population and 'm' be the total number of subjects, then for a particular subject 'j', the population size is 'k' is defined as:

k = 1,2,3,4,5,6,7,8,……n while the number of subjects be:

j = 1,2,3,4,5,6,7,8……m (4.20)

The best result ($X_{j,best,i}$) achieved amongst the intact population of students by taking all the subjects all-together, say the result of best learner i.e. $X_{j,best,i}$ is the result of a best student in a subject j. Focus of a teacher is to train the students to attain improved results out of the learner/students. The teacher is supposed to someone who is an extremely educated person. Hence, we will select our teacher to be that learner who is identified as the best learner using the algorithm. The best result of a teacher ($X_{j,best,i}$) and the mean result of each subject (X_{mean}) are taken in the following equation to estimate the difference mean ($\text{Dmean}_{j,k,i}$) as:-

$$\text{Dmean}_{j,k,i} = r * T_f\{(X_{best}/T_f) - (X_{mean})\} \tag{4.21}$$

where, r is representing a random number and T_f is representing the teaching factor. The teaching factor is the deciding factor for the value of mean, which is aimed to be updated. Here, r will always be less than 1 and more than 0 which means r will always occur in a specific range [0,1]. The value of teaching factor (T_f) will be planned using the following equation, which is also based on random selection and given as:-

$$T_f = \text{round } (r+1) \tag{4.22}$$

Using the difference mean ($\text{Dmean}_{j,k,i}$) calculated in equation no. 4.21, the value of updated new solution ($X_{j,k,i}(new)$) in the teaching phase is:-

$$X_{j,k,i}(new) = \text{Dmean}_{j,k,i} + X_{j,k,i}(old) \tag{4.23}$$

Equation no. 4.23 will help update and regulate the algorithm with each iteration. Here, $X_{j,k,i}(new)$ will only be acknowledged under one condition only. That condition is – new solution provides upgraded function value after analyzing the greedy solution with the above equation. While applying this greedy solution, the new solution will either be accepted or be stand rejected, based on the fact that either it gives improved function value or not. At the completion of this teaching phase, all these upgraded function values becomes input for learning phase, which proves dependency of learning phase on the teaching phase.

“The Learning Phase”

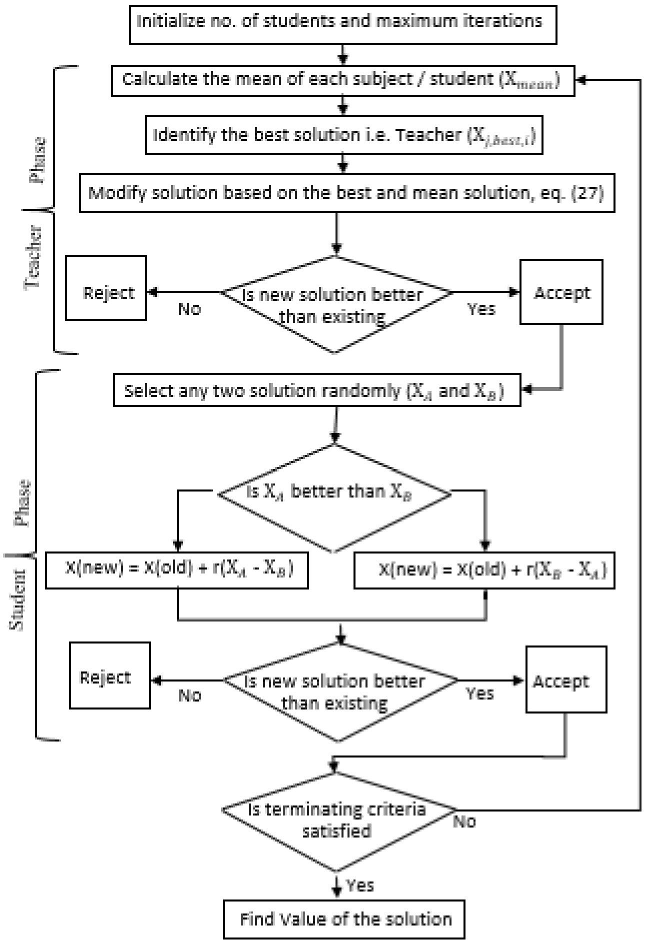
This phase of optimization deals with the improvement of knowledge of students or learners through knowledge among each other all by themselves. One random student will be chosen by another student, this will be known as the selection of a partner student and will use the other student’s knowledge to boost own knowledge. First students will learns from the second if the other learner has supplementary knowledge. For a given population size (say ‘n’ number of learners or students), assume that A and B are the two randomly selected students in such a way that $X_{A,i}(new)$ *and* $X_{B,i}(new)$ are the restructured explanation for $X_{A,i}(old)$ *and* $X_{B,i}(old)$ which are further well-defined in equation no. (4.24) and (4.25).

$$X_{A,i}(new) = r * [X_{A,i}(old) - X_{B,i}(old)] + X_{A,i}(old) \tag{4.24}$$

Equation (4.24) is deemed successful in addressing a given maximization problem only when the updated function evaluation for the current learner surpasses that of the partner learner. Likewise, its effectiveness in tackling a minimization problem is contingent upon the updated function assessment for the current learner being lower than that of the partner learner.

$$X_{A,i}(new) = r * [X_{B,i}(old) - X_{A,i}(old)] + X_{A,i}(old) \qquad (4.25)$$

Equation (4.25) is considered successful in addressing a given maximization problem only when the updated function evaluation for the current learner is less than that of the partner learner. Similarly, its effectiveness in solving a minimization problem is contingent upon the updated function assessment for the current learner being greater than that of the partner learner. This whole process is continued until the optimal solution is achieved. This proper implementation of teaching learning based optimization algorithm, including both phases, is explained with the help of a flow chart in figure 4.4 below.



Initialize no. of students and maximum iterations
Calculate the mean of each subject / student (X_{mean})
Identify the best solution i.e. Teacher ($X_{j,best,i}$)
Modify solution based on the best and mean solution, eq. (27)
Teacher Phase
Reject
No
Is new solution better than existing
Yes
Accept
Select any two solution randomly (X_A and X_B)
Is X_A better than X_B
$X(new) = X(old) + r(X_A - X_B)$
$X(new) = X(old) + r(X_B - X_A)$
Student Phase
Reject
No
Is new solution better than existing
Yes
Accept
Is terminating criteria satisfied
No
Yes
Find Value of the solution

Fig. 4.4 Flow chart for TLBO algorithm.

4.2.2 TLBO algorithm based control of 2DoF ball balancer

To implement teaching learning based optimization integrated while tuning the constraint parameters of proportional-integral-derivative controller of a 2DoF ball balancer system, a MATLAB SIMULINK platform is used and the blocks are arranged as shown in figure 4.5. The required calculations for error signal i.e. e(s) (in Fig. 4.5), which may be re-written from eq. (4.1) as:

$$e(s) = R_d(s) - R(s) \tag{4.51}$$

where, $R_d(s)$ is the desired set point and R(s) is the output in the form of self-control and self-balancing of balancer plate. PID controller is implemented after going through the optimization process using TLBO, on this error signal. The output R(s), error signals e(s) and set point $R_d(s)$ are calculated separately for both the axis i.e. X-axis and Y-axis for the balancer plate. TLBO algorithm is used to optimize the parameters of PID controller, as shown in figure 4.5. Where, $k_{d,initial}$, $k_{i,initial}$ and $k_{p,initial}$ are initial derivative, initial integral and initial proportional gains used to control the balancer plate, for which the initial operating points are given as:

$$\beta_d(s) = \left(k_{d,initial}.s + \frac{k_{i,initial}}{s} + k_{p,initial}\right)[R_d(s) - R(s)] \tag{4.52}$$

Initial parameters of PID controller are calculated using eq. (4.11 – 4.13). Then, controller gains are computed after each iteration of TLBO algorithm as explained in previous subsections of this thesis, which is followed by cost/objective function calculations, as explained already in equation no. (4.14 – 4.18).

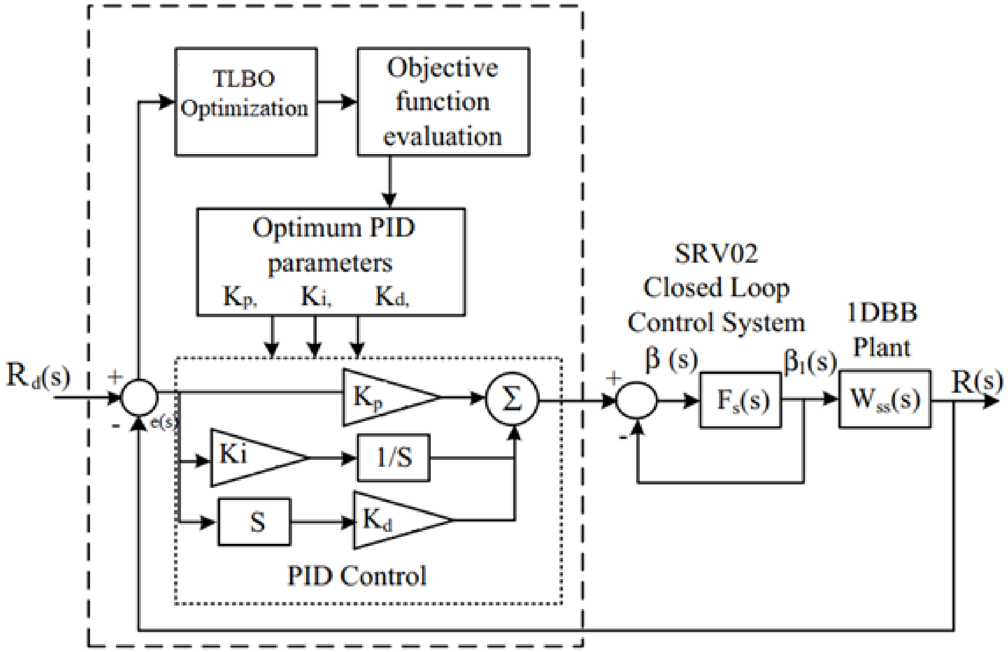


Fig. 4.5 Block diagram representing TLBO based tuning of PID controller of a 2DOF ball balancer system.

The objective is to manage the position of the ball on the rectangular plate, a task achieved through the utilization of optimization algorithm. The crucial aspect of this controller implementation lies in its meticulous calibration to attain the desired output.

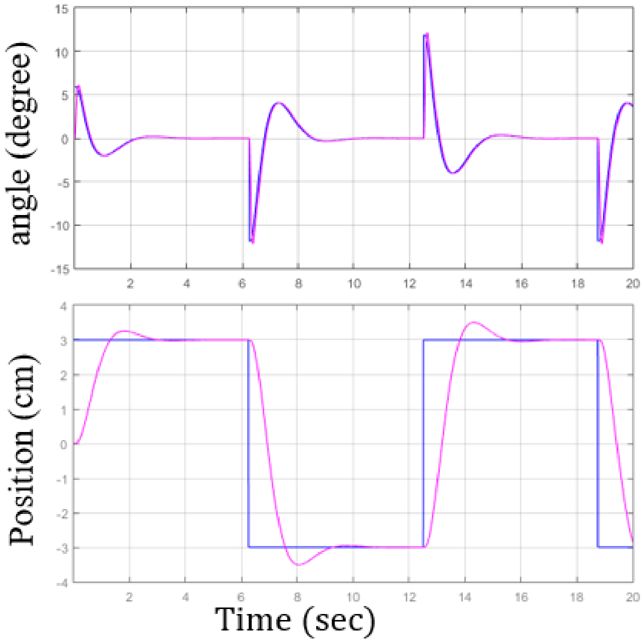


Fig. 4.6 Simulated response of default PD controller showing output variations of servo load angle and ball position of a 2DoF ball balancer system.

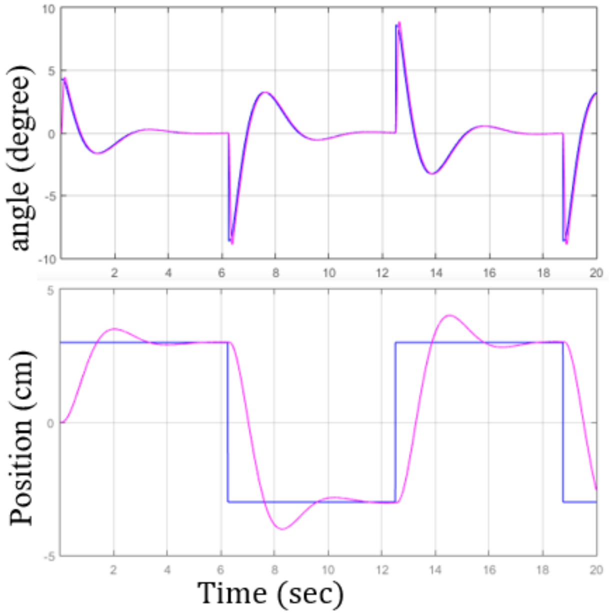


Fig. 4.7 Simulated response of TLBO-PID controller showing output variations of servo load angle and ball position of a 2DoF ball balancer system.

Subsequently, different MATLAB simulations are conducted, and the different outcomes obtained from these simulations are presented in Figs. 4.6 and 4.7. The evaluation of the proposed controller's performance, specifically concerning the enhancement of the servo angle (θ) and ball position (cm) is analysed through graphs that track the variation of these angles over time. In Fig. 4.6, a square reference signal is used on ball position signal while having a servo load angle, illustrating the resulting behaviour of the actual load angle and actual ball position as output when default PD controller is used, as given by Quanser [319], is used as a controller. Integral part is introduced to PD controller, hence making it PID.

Then TLBO algorithm is implemented on this PID controller hence obtaining the graphs in fig. 4.7. The successful control of a nonlinear balancer system is attained through the implementation of the proposed control schemes. This method is contrasted with previous endeavours in the field, employing various controllers. The breakdown of the time response of these simulated result is done further, enabling a comparative evaluation of the superior performance of the proposed control scheme within the simulation model.

In the graphical representations, the blue line serves as the reference trajectory, while the pink line illustrates the actual output. The graphical results distinctly show that the actual output aligns closely with the reference signal when implementing the proposed control schemes.

4.2.2.a Time response analysis

The successful control of a nonlinear balancer system is attained through the implementation of the proposed control schemes. This method is contrasted with previous endeavors in the field, employing various controllers. Table 4.1 delineates the breakdown of the time response, enabling a comparative evaluation of the superior performance of the proposed control scheme within the simulation model. In the graphical representations, the blue line serves as the reference trajectory, while the pink line illustrates the actual output. The graphical results distinctly show that the actual output aligns closely with the reference signal when implementing the proposed control schemes.

Table 4.1: Comparison of time response of servo angle results

Control Scheme used	Rise Time, T_r (sec.)	Settling Time, T_s (sec.)	Peak Overshoot, M_p (%)
Default PD [319]	2.98	3.5	22.9
ZN-PID [327]	2.4	2.76	16.62
TLBO-PID	0.93	1.98	14.1

In the graphical and mathematical results, it is clearly seen that the proposed control schemes are tracking down the servo load angle trajectories of 2DoF ball balancer system. The proposed control scheme helps the balancer to have a stable operation and distortion less processing of feedback mechanism. The graphical results are also demonstrating the mathematical values which are further mentioned in Table 4.1, which makes it clear that the results obtained using the proposed control scheme is providing a more stable behaviour to balancer system on MATLAB/Simulation platform. For comparison purpose, time response analysis is done.

4.2.2.b Cost function analysis

The error observed during this simulation process, is processed through the objective functions and the results observed are shown in Table 4.2 and discussed herewith. The Integral of Absolute Error (IAE), Integral of Square of Error (ISE) and Integral of time weighted Absolute error (ITAE) are calculated as follows:

$$\text{IAE} = \int |e(t)|\, dt \tag{4.53}$$

$$\text{ISE} = \int e^2(t)\, dt \tag{4.54}$$

$$\text{ITAE} = \int t|e(t)|\, dt \tag{4.55}$$

The utilization of the TLBO optimization process for fine-tuning the constraint parameters of the controller led to diminishing in the objective functions from their initial values. In the case of employing the TLBO-PID controller, notable improvements were observed: ITAE decreased to 0.0050, IAE reduced to 0.0049, and ISE diminished to 0.0031, signifying improved error values. The response of the objective functions to the applied optimization algorithms is depicted in Figure 4.8.

Table 4.2: Objective function values

Controller	ISE	IAE	ITAE
ZN-PID	0.0219	0.03533	0.0699
TLBO-PID	0.0031	0.0049	0.0050

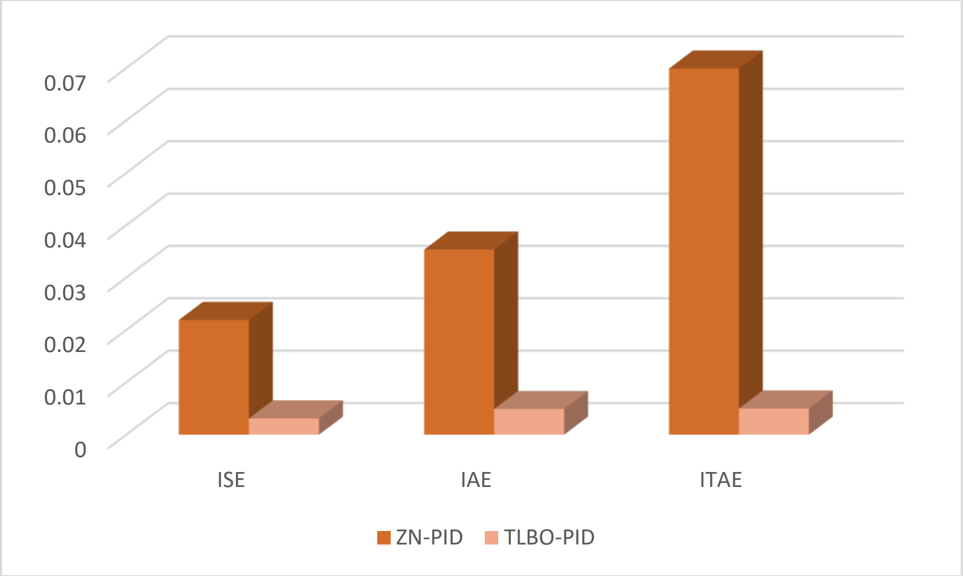


Fig. 4.8 Bar chart showing the values of different objective functions after using classical approach and TLBO optimization algorithm

4.2.3 TLBO algorithm based control of 2DoF helicopter system

To implement teaching learning based optimization while tuning the constraint parameters of proportional-integral-derivative controller of a 2DoF helicopter system, a similar approach is used as it was used to tune the earlier 2DoF system. MATLAB SIMULINK platform is used and the blocks are arranged as shown in figure 4.9. The required calculations for error signal i.e. e(s) (in Fig. 4.9), which may be re-written from eq. (4.1) as:

$$e(s) = R_d(s) - R(s) \tag{4.56}$$

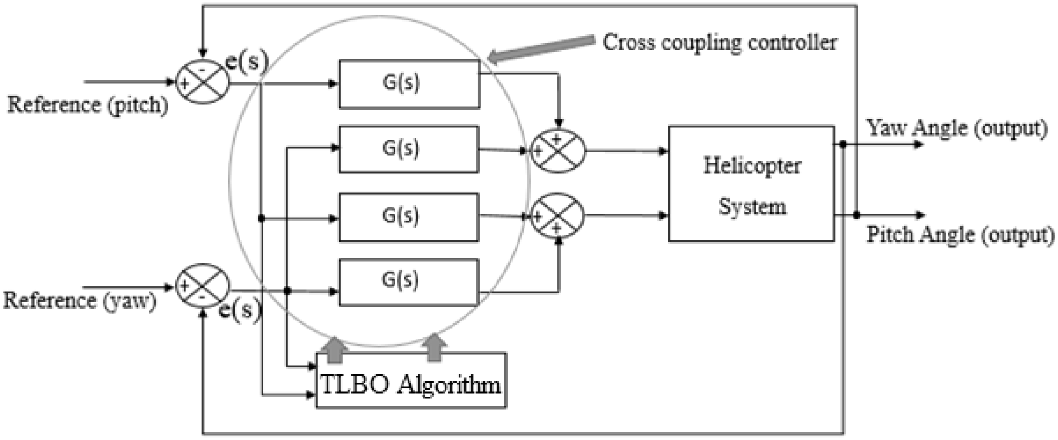


Fig. 4.9 Block diagram representing TLBO based tuning of PID controller of a 2DOF helicopter.

where, $R_d(s)$ is the desired set point and $R(s)$ is the output in the form of pitch and yaw angle of helicopter model. The controller block G(s) is represented for PID controller. The algorithm block represents the TLBO algorithm. PID controller is implemented after going through the optimization process using TLBO, on error signal. The output R(s), error signals e(s) and set point $R_d(s)$ are calculated separately for both the axis i.e. pitch-axis and Yaw-axis for the helicopter Simulink model. TLBO algorithm is used to optimize the parameters of this PID controller.

4.2.3.a Time response analysis

For the pitch axis of the nonlinear system, an error signal e(s) is minimized using TLBO algorithm, representing the difference between the desired and actual pitch angles in radians. Similarly, the error signal of yaw axis is also minimized using the same algorithmic approach. The algorithms are using proportional-integral-derivative controller’s constraints for the optimization process. Error signals are delivered as inputs for the PID controllers. Two PID controllers are utilized—one to manage yaw error and the other to handle pitch error. Both controllers collaborate to achieve a common objective: reaching the reference altitude, optimizing speed, and aligning with the angle values specified by the reference trajectory. The crucial aspect of the implemented controller is fine-tuning the constraint parameters. Subsequently, different MATLAB simulations are conducted and the diverse outcomes obtained from these simulations are presented in Figs. 4.10 and 4.11.

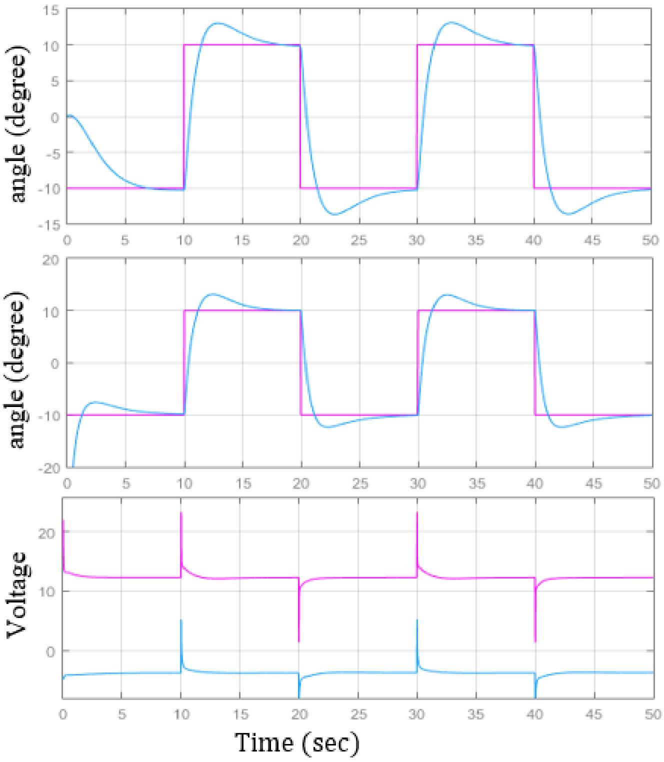


Fig. 4.10 Simulated response of LQR based controller showing output variations of yaw angle trajectory, pitch angle trajectory and voltage.

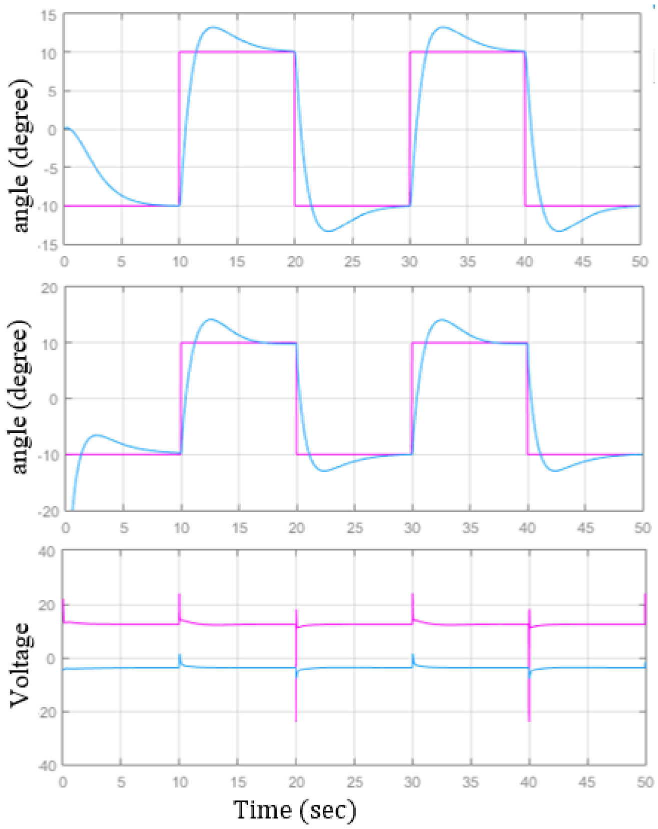


Fig. 4.11 Simulated response of TLBO-PID control mechanism showing output of yaw angle trajectory, pitch angle trajectory and voltage variations.

The evaluation of the proposed controller's performance, specifically concerning the enhancement of the pitch angle (θ) and yaw angle (Ψ) is analyzed through graphs that track the variation of these angles over time. In Fig. 4.10, a square reference signal is applied to the pitch trajectory and yaw trajectory, illustrating the resulting behavior of the actual pitch angle and yaw angle output when default LQR controller, as given by Quanser [322] is used as a controller. In Fig. 4.11, same square reference signal is applied to the pitch trajectory and yaw trajectory, illustrating the resulting behavior of the actual pitch angle and yaw angle output when TLBO optimized PID controller is used as the control mechanism.

The effective regulation of a nonlinear helicopter model is achieved by implementing the proposed control strategies. In comparison to previous approaches within the field, where Quanser [322] utilized the LQR controller to minimize pitch and yaw errors in a two-degree-of-freedom helicopter model, our current work employs a similar control system architecture with different controllers. To illustrate this, step and sinusoidal reference input trajectories are employed and visualized on the MATLAB Simulink interface. Tables 4.3 & Table 4.4 present a breakdown of the time response, enabling a comparative evaluation of the performance superiority of the proposed control scheme in the simulation model.

Table 4.3: Comparison of simulation results obtained after time response analysis of pitch angle

Control Scheme used	Rise Time (sec.)	Settling Time (sec.)	Peak Over-shoot (%)
LQR [322]	1.1	4.72	8.99
TLBO-PID	0.66	3.99	4.18

Table 4.4: Comparison of simulation results obtained after time response analysis of yaw angle

Control Scheme used	Rise Time (sec.)	Settling Time (sec.)	Peak Over-shoot (%)
LQR [322]	1.0	5.12	18.3
TLBO-PID	0.5	4.01	8.19

4.2.3. b Cost function analysis

The tracking of reference pitch and yaw trajectories for the 2-degree-of-freedom helicopter model is carried out using step references, and the results obtained are subjected to a comparative analysis. The aforementioned outcomes are sufficient to establish the superiority of the hybridized method over individual algorithms. The TLBO algorithm effectively manages the rise time, peak time, peak overshoot, and settling time for both the pitch and yaw axes. Various derivatives of error serve as objective functions in this simulation analysis, namely Integral Square Error (ISE), Integral Absolute Error (IAE), and Integral Time Square Error (ITSE), demonstrating the reduction in error during the implementation of optimization algorithms are calculated as follows:

$$\text{IAE} = \int |e(t)|\, dt \tag{4.56}$$

$$\text{ISE} = \int e^2(t)\, dt \tag{4.57}$$

$$\text{ITSE} = \int te^2(t)\, dt \tag{4.58}$$

The utilization of the TLBO optimization process for fine-tuning the constraint parameters of the controller led to diminishing the objective functions from their initial values. In the case of employing the TLBO-PID controller, notable improvements were observed which are mentioned in Table 4.5.

Table 4.5: Values of different error derivatives after applying TLBO-PID based control mechanism

Controller	IAE	ISE	ITSE
TLBO-PID	0.8396	0.11	1.714

4.3 GPC algorithm based control mechanism

4.3.1 GPC algorithm

The expression "Giza pyramid construction algorithm" lacks recognition within the realms of computer science, engineering, or any affiliated discipline. It appears to be a term employed solely in the context of an analogy or metaphor. In this particular context, the analogy highlights the strategic management of workers constructing the Giza pyramids by a leader known as Pharaoh's agent, aiming to attain optimal performance. This analogy draws parallels with the control strategy of the GPC algorithm. The GPC algorithm draws inspiration from the construction methods employed by workers building the Giza pyramids [49]. These workers, overseen by a leader referred to as Pharaoh's agent, were strategically managed to extract optimal performance from each individual. The control strategy of the GPC algorithm is based on the method employed by these workers to transport stone blocks to their assigned positions. Initially, the stone blocks are randomly positioned at the pyramid construction site. Workers are tasked with pushing these blocks to their desired locations, utilizing a ramp for this purpose. The workers manually slide the stone blocks over the ramp, revealing both the starting positions and associated costs of the blocks. This movement is influenced by factors such as the friction of the ramp and the capabilities of the workers. The interchangeability of workers, guided by Pharaoh's agent based on their varying powers and capabilities, allows for efficient execution of the task.

Figure 4.12 illustrates the free-body movement of stone blocks over the ramp, showcasing all the forces acting on the blocks. In Figure 4.13, a detailed procedural flow chart outlines the steps to be followed for the proper execution of the GPC algorithm. The algorithm operates under certain assumptions:

- Only one straight ramp with an even surface is utilized.
- The maximum angle of the ramp is set at 15 degrees.

- Friction for workers is neglected.
- Depending on the project's needs, workers may be substituted as required.

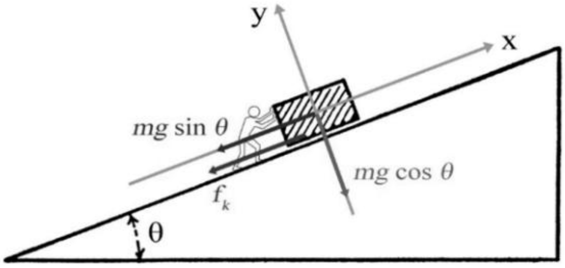


Fig. 4.12 Stone block position on the ramp with all the forces

Friction force (Kinetic) and Newton’s second law are detailed as:

$$\mu_k f m g \cos\theta = f_{kf} \tag{4.59}$$

$$0 = m a_s + m g \sin\theta + f_{kf} \tag{4.60}$$

In these equations, 'm' signifies the mass of an individual block, 'g' denotes gravity, 'θ' represents the ramp angle, 'a_s' stands for acceleration, and the friction coefficient is denoted by μ_{kf}. 'v_0' and 'd' indicate the initial velocity and the distance covered by the stone respectively. By employing the aforementioned formulas in conjunction with Newton's equations for motion, we can derive the following results:

$$-g\left(\sin\theta + \mu_{kf}\cos\theta\right) = a_s \tag{4.61}$$

$$\frac{v_0^2}{2g\left(\sin\theta + \mu_{kf}\cos\theta\right)} = d \tag{4.62}$$

In every iteration of GPC, the initial velocity of stone block is defined using:

$$v_0 = rand(0,1) \tag{4.63}$$

Given that the initial velocity falls within the range of 0 to 1, the determination of the friction coefficient range is established by:

$$\mu_{kf} = rand\left[\mu_{k\_min}, \mu_{k\_max}\right] \tag{4.64}$$

μ_{k_min} and μ_{k_max} are predefined in the algorithm. Friction is distributed randomly across the surface, and the calculation of the worker's movement is determined by:

$$\frac{v_0^2}{2g\sin\theta} = x \tag{4.65}$$

The updated positions of both the stone and the worker are determined by computing the values of 'd' and 'x'. Assume the initial position is denoted by vector $\overrightarrow{p_\iota}$, and $\overrightarrow{\epsilon_\iota}$

represents the random uniform distribution vector. Consequently, in the subsequent iteration, the calculation for the new worker position is expressed as:

$$\vec{p} = (\overrightarrow{p_i} + d) \times x\overrightarrow{\epsilon_i} \tag{4.66}$$

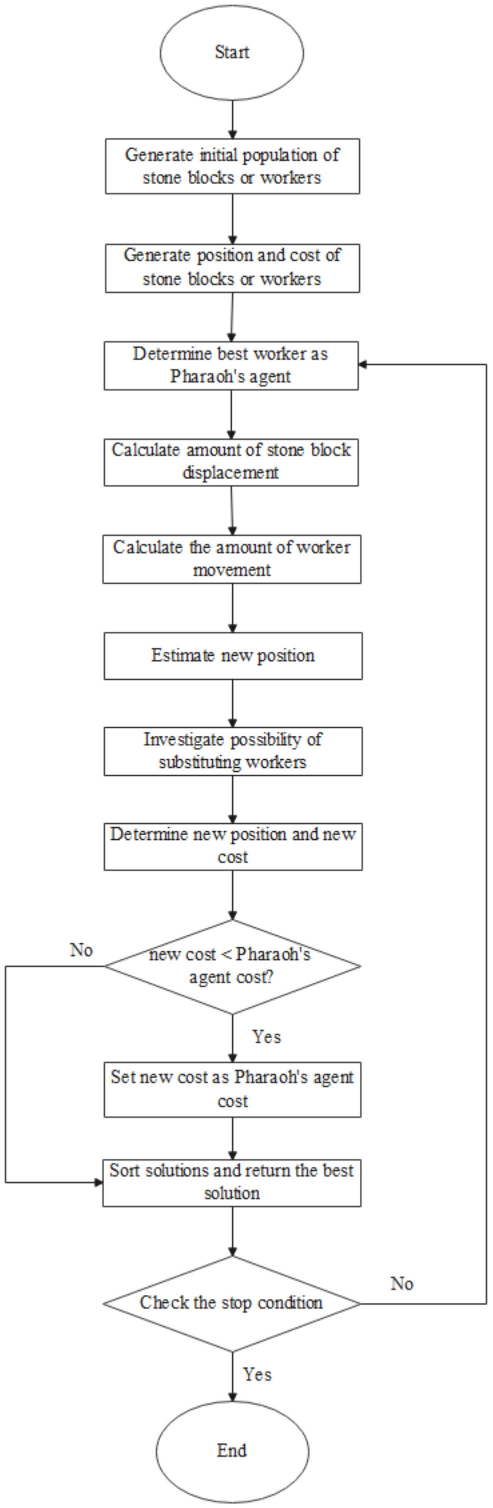

Start
Generate initial population of stone blocks or workers
Generate position and cost of stone blocks or workers
Determine best worker as Pharaoh's agent
Calculate amount of stone block displacement
Calculate the amount of worker movement
Estimate new position
Investigate possibility of substituting workers
Determine new position and new cost
new cost < Pharaoh's agent cost?
No
Yes
Set new cost as Pharaoh's agent cost
Sort solutions and return the best solution
Check the stop condition
No
Yes
End

Fig. 4.13 Detailed flow-chart for GPC

The position derived from the aforementioned equation serves as the basis for a fresh solution in each iteration. Subsequently, the determination of the effective positions for all workers is contingent upon the probability of worker substitution, set at 50%. This substitution process is executed through the utilization of the uniform crossover operator. Here, the initial operator solution and the generated solution are outlined as follows:

$$\phi = (\varphi_1, \varphi_2, \cdots\cdots \varphi_n) \tag{4.67}$$

$$\psi = (\psi_1, \psi_2, \dots, \psi_n) \tag{4.68}$$

Updated solution after substitution: $Z = (\zeta_1, \zeta_2, \dots, \zeta_n)$ (4.69)

The formula employed for worker substitution in each iteration of the algorithm is determined by the following equation:

$$\zeta = \begin{cases} \psi, if\ rand[0,1] \leq\ 0.5 \\ \varphi, otherwise \end{cases} \tag{4.70}$$

4.3.2 GPC algorithm based control of 2DoF ball balancer system

Giza pyramid construction (GPC) algorithm is used to tune the constraint parameters of proportional-integral-derivative controller of a 2DoF ball balancer system. A similar MATLAB SIMULINK platform is used for this purpose, which was used to implement the TLBO algorithm earlier in this chapter. The blocks are re-arranged as shown in figure 4.14 below. The required calculations for error signal is same what we used in eq (4.51) which may be re-written from eq. (1) as:

$$\text{e(s)} = R_d\text{(s)} - \text{R(s)} \tag{4.71}$$

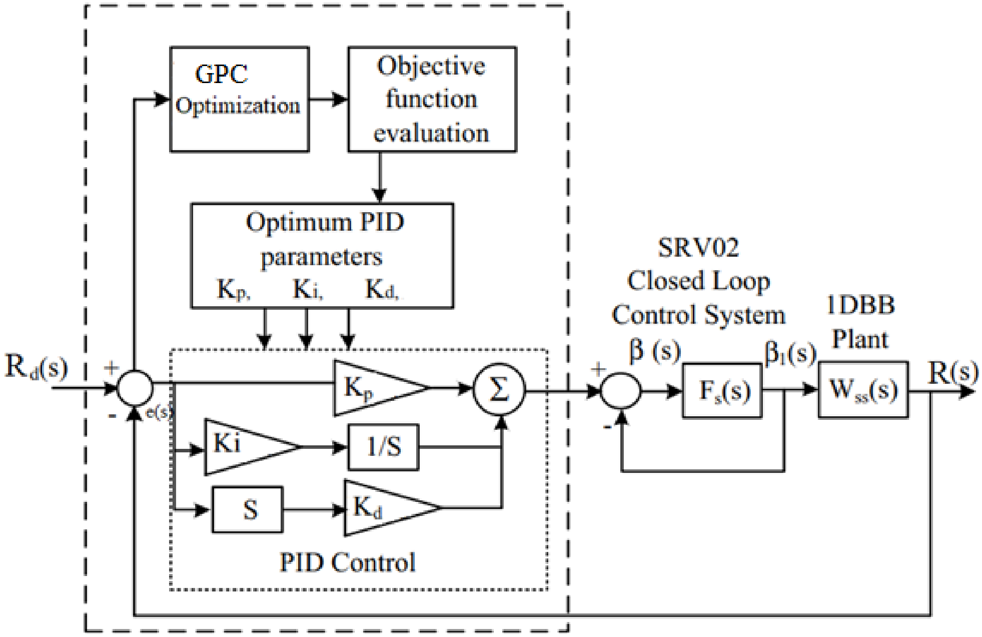


Fig. 4.14 Block diagram representing GPC based tuning of PID controller of a 2DOF ball balancer system

where, $R_d(s)$ is the desired set point and $\mathrm{R}(s)$ is the response in the form of position and angle control of 2DoF ball balancer. After undergoing optimization through GPC, the PID controller is applied to the error signal. The initialization process, detailed in equations (51) and (52) in this chapter, sets the initial parameters of the PID controller using equations (4.11 – 4.13). Subsequently, cost/objective functions are computed, as previously described in equations (4.14 – 4.18). The key to the effectiveness of this controller implementation lies in its careful calibration to achieve the desired output.

Following that, a series of MATLAB simulations are executed, and the diverse outcomes derived from these simulations are depicted in Figs. 4.15 and 4.16. The assessment of the performance of the suggested controller, particularly in terms of enhancing the servo angle (θ) and ball position (cm), is examined through graphs illustrating the variation of these angles over time. Fig. 4.15 employs a square reference signal on the ball position signal with a servo load angle, showcasing the observed behavior of the actual load angle and actual ball position when using the default PD controller provided by Quanser [319].

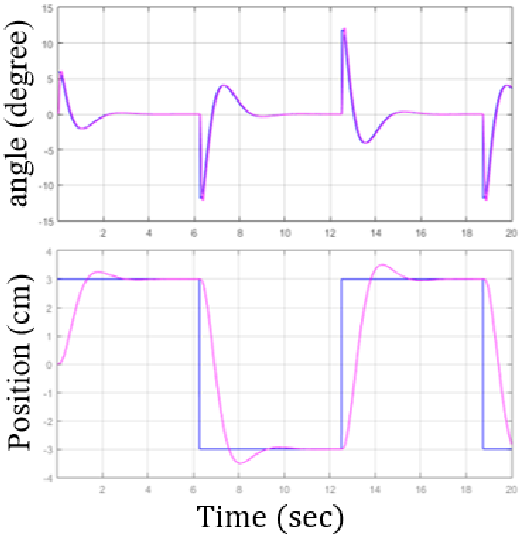


Fig. 4.15 Simulated response of PD controller showing output variations of servo load angle and ball position

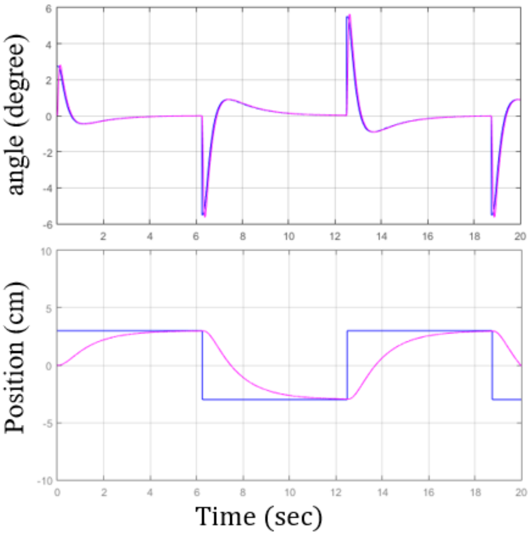


Fig. 4.16 Simulated response of GPC-PID control mechanism showing output of servo load angle and ball position

The introduction of the integral part transforms the PD controller into a PID controller. Subsequently, the GPC algorithm is applied to this PID controller, resulting in the graphs presented in Fig. 4.16. The effective control of a nonlinear balancer system is achieved through the implementation of the proposed control schemes, which are compared with previous approaches in the field involving various controllers. Further analysis breaks down the time response of these simulated results, facilitating a comparative assessment of the superior performance of the proposed control scheme within the simulation model. In the graphical representations, the blue line signifies the reference trajectory, while the pink line depicts the actual output. The graphical results distinctly demonstrate a close alignment between the actual output and the reference signal when employing the proposed control schemes.

4.3.2.a Time response analysis

The effective regulation of a nonlinear balancer system is achieved through the application of the proposed control schemes. This approach is compared to prior initiatives in the field that utilized different controllers. Table 4.6 outlines the breakdown of the time response, facilitating a comparative assessment of the superior performance of the proposed control scheme within the simulation model. In the visual representations, the blue line represents the reference trajectory, while the pink line depicts the actual output. The graphical results unmistakably reveal a close alignment between the actual output and the reference signal when employing the proposed control schemes. Both in the visual and mathematical outcomes, it is evident that the proposed control schemes accurately follow the servo load angle trajectories of the 2DoF ball balancer system. The proposed control scheme contributes to the balancer's stable operation and distortion-free processing of the feedback mechanism. The graphical results also illustrate the mathematical values, as further detailed in Table 4.6, making it evident that the results obtained using the proposed control scheme provide a more stable behaviour for the balancer system on the MATLAB/Simulation platform. Time response analysis is conducted for comparison purposes.

Table 4.6: Comparison of time response servo angle results

Control Scheme used	Rise Time (sec.)	Settling Time (sec.)	Peak Over-shoot (%)
Default PD [319]	2.98	3.5	22.9
ZN-PID [327]	2.4	2.76	16.62
GPC-PID	0.83	1.8	12.33

4.3.2.b Cost function analysis

The error encountered in this simulation is subjected to the objective functions, and the outcomes are presented in Table 4.7, discussed below. The Integral of Absolute Error (IAE), Integral of Square of Error (ISE), and Integral of Time Weighted Absolute Error (ITAE) are computed as follows:

$$\text{IAE} = \int |e(t)|\, dt \tag{4.72}$$

$$\text{ISE} = \int e^2(t)\, dt \tag{4.73}$$

$$\text{ITAE} = \int t|e(t)|\, dt \tag{4.74}$$

Table 4.7: Objective function values

Controller	ISE	IAE	ITAE
ZN-PID	0.0219	0.03533	0.0699
GPC-PID	0.0033	0.0054	0.0066

The application of the TLBO optimization process to fine-tune the constraint parameters of the controller resulted in a reduction of the objective functions from their initial values. When utilizing the TLBO-PID controller, significant enhancements were observed: ITAE decreased to 0.0050, IAE reduced to 0.0049, and ISE diminished to 0.0031, indicating improved error values. The response of the objective functions to the applied optimization algorithms is illustrated in Figure 4.17.

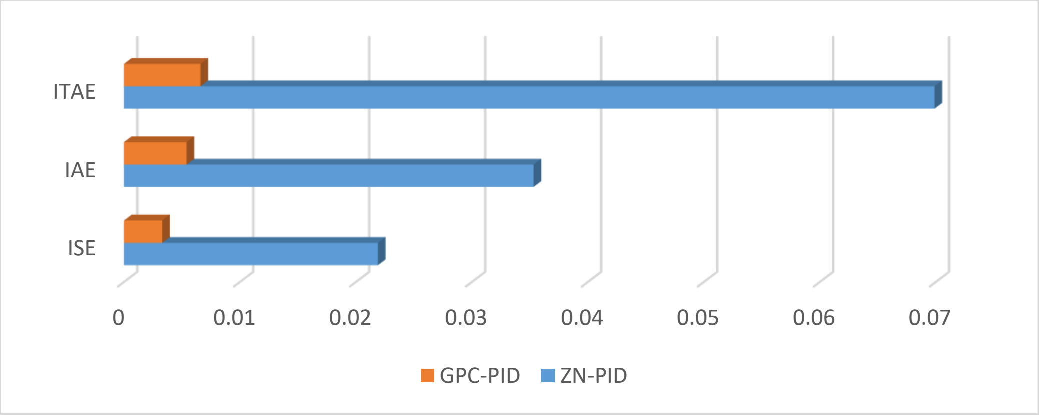


Fig. 4.17 Comparison of ISE, IAE & ITAE response of GPC algorithm with ZN tuned PID

4.3.3 GPC algorithm based control of 2DoF helicopter system

To incorporate Giza pyramid construction (GPC) optimization algorithm for adjusting the constraint parameters of a proportional-integral-derivative controller in a 2DoF helicopter system, similar method is used which we preferred to incorporate the TLBO based control scheme. The MATLAB SIMULINK platform is utilized, and the arrangement of blocks is depicted in Figure 4.18. The necessary computations for the error signal, has already been discussed in earlier sections in details.

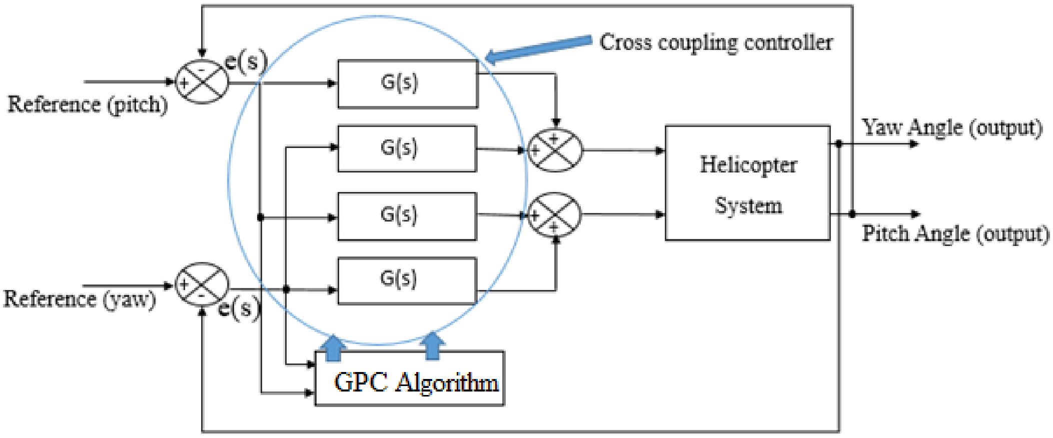


Fig. 4.18 Block diagram representing TLBO based tuning of PID controller of a 2DOF helicopter

Fig. 4.18 is same as fig. 4.9 and fig. 4.3 and has been discussed in detail earlier in the chapter. Only the algorithm approach is altered and rest whole functioning remains the same. GPC algorithm is used to optimize the parameters of this PID controller. In optimizing the pitch axis of the nonlinear system, the GPC algorithm is employed to minimize the error signal, denoted as e(s), which signifies the difference between the desired and actual pitch angles measured in radians. Likewise, the error signal for the yaw axis undergoes a similar minimization process using the GPC algorithm. These algorithms utilize the constraints of the proportional-integral-derivative (PID) controller for the optimization procedure. The error signals are fed as inputs to the PID controllers, where two PID controllers are employed—one to address yaw error and another to handle pitch error. In this way, both the trajectory angles are controlled using the same optimization approach.

4.3.3.a Time response analysis

Both controllers collaborate to attain a shared objective: achieving the reference altitude, optimizing speed, and aligning with the angle values specified by the reference trajectory. The critical aspect of the implemented controller involves meticulous adjustment of the constraint parameters. Following this, various MATLAB simulations are executed, and the diverse outcomes derived from these simulations are illustrated in Figs. 4.19 and 4.20. The assessment of the proposed controller's performance, particularly in enhancing the pitch angle (θ) and yaw angle (Ψ), is analysed through graphs depicting the variation of these angles over time. The effective regulation of a nonlinear helicopter model is achieved by implementing the proposed control strategies. In contrast to previous methodologies in this experiment, where Quanser [322] employed the LQR controller to minimize pitch and yaw errors in a two-degree-of-freedom helicopter model, our present investigation adopts a similar control system architecture but utilizes distinct controllers. To illustrate this, step and sinusoidal reference input trajectories are applied and visualized using the MATLAB Simulink interface. Tables 4.8 & Table 4.9 break down the time response, enabling a comparative evaluation of the superior performance of the proposed control scheme in the simulation model.

Table 4.8: Comparison of simulation results obtained after time response analysis of pitch angle

Control Scheme used	Rise Time (sec.)	Settling Time (sec.)	Peak Over-shoot (%)
LQR [322]	1.1	4.72	8.99
GPC-PID	0.58	1.11	4.3

Table 4.9: Comparison of simulation results obtained after time response analysis of yaw angle

Control Scheme used	Rise Time (sec.)	Settling Time (sec.)	Peak Over-shoot (%)
LQR [322]	1.0	5.12	18.3
GPC-PID	0.45	3.94	8.21

In Fig. 4.19, a square reference signal is applied to the pitch and yaw trajectories, showcasing the resulting behaviour of the actual pitch angle and yaw angle output when the LQR controller, as provided by Quanser [322], is utilized. In Fig. 4.20, the same square reference signal is applied to the pitch and yaw trajectories, demonstrating the resulting behaviour of the actual pitch angle and yaw angle output when the GPC-optimized PID controller is employed as the control mechanism.

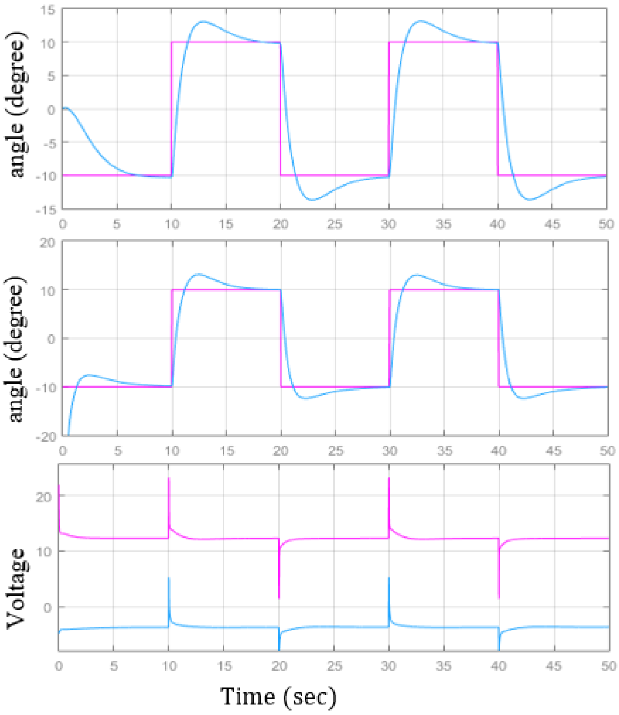


Fig. 4.19 Simulated response of LQR based controller showing output variations of yaw angle trajectory, pitch angle trajectory and voltage

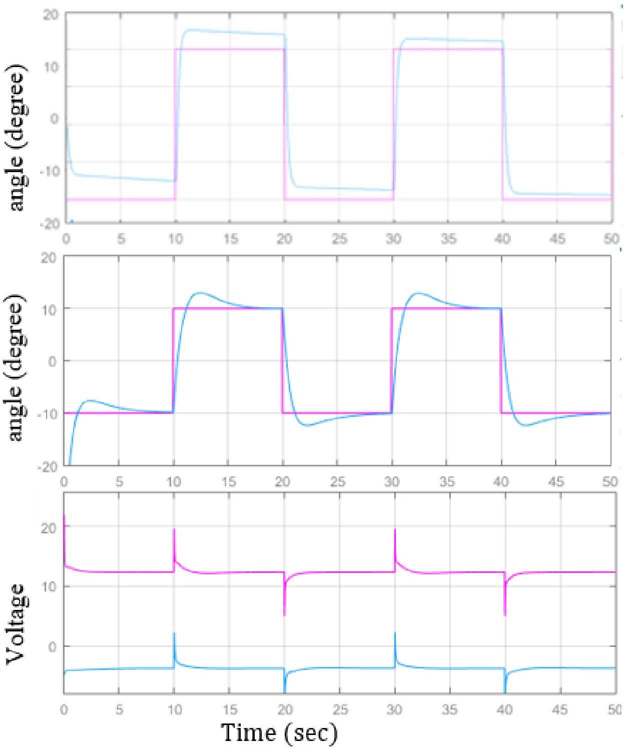


Fig. 4.20 Simulated response of GPC-PID control mechanism showing output of yaw angle trajectory, pitch angle trajectory and voltage variations

4.3.3.b Cost function analysis

The tracking of reference pitch and yaw trajectories in the 2-degree-of-freedom helicopter model is conducted using step references and the resulting outcomes go through a comparative analysis. These results are adequate to affirm the superiority of the hybridized method over individual algorithms. The GPC algorithm effectively controls the rise time, peak time, peak overshoot, and settling time for both the pitch and yaw axes. Various error derivatives serve as objective functions in this simulation analysis, specifically Integral Square Error (ISE), Integral Absolute Error (IAE), and Integral Time Square Error (ITSE), demonstrating the reduction in error during the implementation of optimization algorithms, and they are computed as follows:

$$\text{IAE} = \int |e(t)|\, dt \tag{4.75}$$

$$\text{ISE} = \int e^2(t)\, dt \tag{4.76}$$

$$\text{ITSE} = \int te^2(t)\, dt \tag{4.77}$$

Applying the GPC optimization algorithm to adjust the constraint parameters of the controller resulted in a reduction of the objective functions from their initial values. When implementing the GPC-PID controller, significant enhancements were observed, as detailed in Table 4.10.

Table 4.10: Values of different error derivatives after applying TLBO-PID based control mechanism

Controller	IAE	ISE	ITSE
GPC	0.7717	0.1046	1.616

4.4 Conclusion

This work demonstrates the application of optimization algorithms to both a 2 DOF helicopter system and a 2DoF ball balancer system, with a primary focus on refining the constraint parameters of the PID controller to govern the servo angle trajectory. The emphasis then shifts towards trajectory control of flight angles. The finely-tuned results are compared with outcomes obtained from an LQR controller and PD controller. The performance analysis specifically investigates the servo angle, yaw angle, and pitch angles using various control techniques. The constrained optimization of the PID controller notably reduces cross-coupling errors between the angles of the helicopter model and the servo angle error of the rectangular plate on which a ball is balanced. To evaluate the output simulation performances of the tuned PID controller, default PD, and LQR controller, the MATLAB simulation platform is employed. Graphical and time response analyses are utilized to provide a comprehensive comparison among these controllers.

Chapter 5. Improved TLBO Algorithm Based Control of 2DOF Systems

Teaching Learning-Based Optimization (TLBO) provides a well-known solution for optimization problems. Although TLBO has demonstrated successful applications across various problem domains, like any algorithm, it is not without its drawbacks. Some potential limitations of TLBO includes:

- Parameter Sensitivity: Similar to many optimization algorithms, TLBO involves adjusting several parameters and the algorithm's performance can be highly sensitive to these settings. Identifying the optimal parameter set for a specific problem can pose a challenging task.
- Applicability to Specific Problem Types: TLBO may not be uniformly effective across all types of optimization problems. Its performance can fluctuate based on the characteristics of the problem being addressed, and it may not be the most appropriate choice for certain optimization tasks.
- Convergence Speed: TLBO may demonstrate slower convergence rates compared to certain other optimization algorithms. This limitation becomes significant when dealing with problems requiring swift convergence or under constraints of limited computational resources.
- Challenges in Noisy or Stochastic Environments: TLBO may face challenges when applied to optimization problems featuring noisy or stochastic objective functions. The deterministic nature of TLBO might not be well-suited to handle such scenarios.
- Limited Mathematical Foundation: Similar to many other nature-inspired algorithms, TLBO may lack a robust theoretical foundation compared to more established optimization techniques. This absence can make it challenging to provide guarantees about its performance in specific situations.

While acknowledging these possible constraints, it is important to understand that the success of TLBO relies on the particular problem at hand and its application context. The optimization procedure employed by the TLBO algorithm has been extensively examined in the preceding chapters of this thesis. It has been noted that the generation of the teaching factor plays a crucial role in the optimization process. This teaching factor is influenced by the generation of random numbers within the algorithm, introducing sensitivity to randomization in calculations. This opens up opportunities for enhancing the generation of the teaching factor to ensure the effective implementation of the teaching-learning based optimization process. Subsequent sections of this thesis will delve into discussions regarding these enhancements.

5.1 Improved TLBO (iTLBO)

5.1.1 Updating process for teaching factor

The previous chapter extensively covers the computation and significance of the teaching factor in the TLBO algorithm. However, the algorithm does not assess the cost as a meaningful parameter since it is randomly categorized by the algorithm. In examining the outcomes from operations on various test functions conducted by R.V.

Rao et al. [50], it was observed that the algorithm exhibits quicker responses when the teaching factor (T_f) is maintained within the range of 1 and 2. According to R.V. Rao's research, the selection process for the teaching factor value is entirely dependent on the algorithm's randomized selection procedure. Moreover, the algorithm yields satisfactory results only when the value is either 1 or 2.

These statements highlight a lack of data involvement in the TLBO algorithm. To simplify the implementation of the algorithm and prevent complications, the final value of the teaching factor is determined as either 1 or 2 by rounding off the corresponding equation from the previous chapter, which is given as:

$$T_f = \text{round}\ (r+1) \tag{5.1}$$

However, if we refrain from adopting unnecessary shortcuts and take a closer look at the algorithm's randomized behavior, it becomes evident that the elimination of this random selection can be achieved by making the teaching factor dependent on the available data during algorithm implementation. The focus of this work is to explore the core of proposed algorithm, aiming to enhance overall results.

In pursuit of this improvement, a factor of enhancement (i_f) is preferred to eliminate the random value (r) command from the teaching-learning-based optimization algorithm. The formula determining the value of i_f is employed in the algorithm throughout the optimization process as:

$$i_f = (1 + (X_{j,best,i} - X_{mean}))^{-1} \tag{5.2}$$

Where the value of $X_{j,best,i}$ & X_{mean} will be calculated based on the calculations already discussed in "teaching phase" of TLBO algorithm. In the TLBO algorithm, the teaching factor is currently assigned a randomized rounded-off number, indicating a potential area for enhancement. To address this issue in the algorithm, the revised formula for calculating the teaching factor can be referred to as:-

$$T_f = \frac{2 + (X_{j,best,i} - X_{mean})}{1 + (X_{j,best,i} - X_{mean})} \tag{5.3}$$

5.1.2 Updated data-driven solution

The updated new results are approached using following equations, as adopted from "teaching phase" and "learning phase" of TLBO algorithm, as derived in previous chapter:-

$$X_{j,k,i}(new) = \text{Dmean}_{j,k,i} + X_{j,k,i}(old) \tag{5.4}$$

$$X_{A,i}(new) = r * [X_{A,i}(old) - X_{B,i}(old)] + X_{A,i}(old) \tag{5.5}$$

$$X_{A,i}(new) = r * [X_{B,i}(old) - X_{A,i}(old)] + X_{A,i}(old) \tag{5.6}$$

$$T_f = \frac{2 + (X_{j,best,i} - X_{mean})}{1 + (X_{j,best,i} - X_{mean})} \tag{5.7}$$

Once the teaching factor updating procedure is complete using equation no (5.7), the assessment given by equation no (5.4), (5.5) and (5.6) also gets updated in the consequent iterations of algorithm. Hence, new solution during the teaching phase can be explained using the following formula:

$$X_{j,k,i}(new) = X_{j,k,i}(old) + \left\{\frac{1}{1+(X_{j,best,i}-X_{mean})}\right\} * \left\{X_{best} - \frac{2+(X_{j,best,i}-X_{mean})}{1+(X_{j,best,i}-X_{mean})} X_{mean}\right\} \tag{5.8}$$

Above equation (i.e. Eq. (5.4)) provides new solution after the teaching phase completion in iTLBO algorithm. This is also responsible for improved results during implementation of the learning phase of iTLBO. To make the new solution in the generation of algorithm in learning phase total independent on random behavior, the Eq. (5.2) is integrated with Eq. (5.5 – 5.6) and provides the data derived new solutions for learning phase as:

$$X_{A,i}(new) = X_{A,i}(old) + \left\{\frac{1}{1+(X_{j,best,i}-X_{mean})}\right\} * \chi_1 \tag{5.9}$$

Where, $\chi_1 = \{X_{A,i}(old) - X_{B,i}(old)\}$ and equation no. (5.9) will be considered valid for solving a maximization problems if and only if χ_1 is giving positive value and valid for solving a minimization problems if and only if χ_1 is giving a negative value.

Further, this gives us new solution as:

$$X_{A,i}(new) = X_{A,i}(old) + \left\{\frac{1}{1+(X_{j,best,i}-X_{mean})}\right\} * \chi_2 \tag{5.10}$$

Where, $\chi_2 = [X_{B,i}(old) - X_{A,i}(old)]$ and equation no (5.10) will be considered valid for solving a maximization problems if and only if χ_2 is giving a positive value and valid for solving minimization problems if and only if χ_2 is giving a negative value.

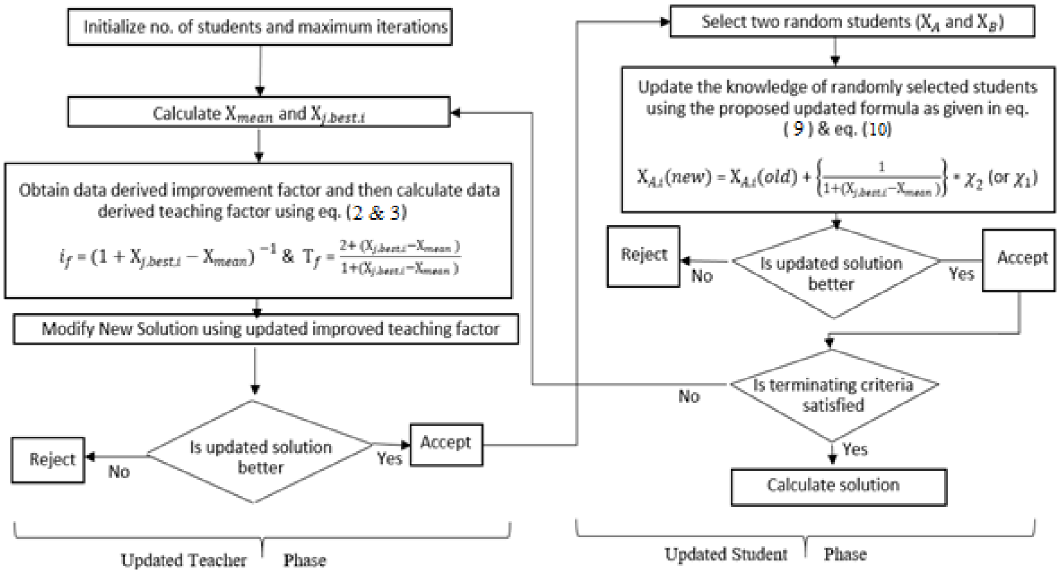

Initialize no. of students and maximum iterations
Calculate X_{mean} and $X_{j,best,i}$
Obtain data derived improvement factor and then calculate data derived teaching factor using eq. (2 & 3)
$i_f = (1 + X_{j,best,i} - X_{mean})^{-1}$ & $T_f = \frac{2+(X_{j,best,i}-X_{mean})}{1+(X_{j,best,i}-X_{mean})}$
Modify New Solution using updated improved teaching factor
Is updated solution better
Reject
No
Yes
Accept
Select two random students (X_A and X_B)
Update the knowledge of randomly selected students using the proposed updated formula as given in eq. (9) & eq. (10)
$X_{A,i}(new) = X_{A,i}(old) + \left\{\frac{1}{1+(X_{j,best,i}-X_{mean})}\right\} * \chi_2$ (or χ_1)
Reject
No
Is updated solution better
Yes
Accept
No
Is terminating criteria satisfied
Yes
Calculate solution
Updated Teacher Phase
Updated Student Phase

Fig. 5.1 Detailed flow chart illustrating the iTLBO algorithm

When we eliminates randomized behaviour from algorithm of teaching learning based optimization method using iTLBO approach, the algorithm becomes more consistent and attains better precision. The data driven updated approach gives stable output and quicker response with test benchmark functions and 2DOF system. These are discussed further in details and the whole execution of improved TLBO algorithm is explained using a flow chart in figure 5.1. It helps in providing a textual representation of algorithm flow for optimization process of iTLBO, which provides general outline or description of the typical steps involved in this improvised optimization algorithm.

5.2 iTLBO on traditional benchmark functions

This section provides evidence of the superior performance of the proposed optimization procedure compared to existing techniques found in the literature. Various experiments have been conducted to substantiate the enhanced efficacy of the Improved Teaching-Learning-Based Optimization (iTLBO) algorithm. These experiments encompass a range of benchmark functions, including four distinct constrained benchmark functions. Among them, two exhibit linear characteristics, while the remaining two showcase nonlinear characteristics in their objective functions. This problem is resolved using particle swarm optimization, artificial bee colony and teaching learning based optimization by Rao et. al. [50]. The specific attributes of these functions are elaborated upon below:

- Benchmark test function 1

For this we are taking a minimization linear problem which is having nine linear equality constraints and total thirteen design variables. As far as the number of active constraint are considered at the optimum point, there will be total six, which finally gives the global minimum at y^* = (1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 1) and have single objective functional value given as f(y^*) = – 15. The problem may be defined using the following equations.

$$\text{Min } f(y) = 5\sum_{i=1}^{4} y_i - 5\sum_{i=1}^{4} y_i^2 - 5\sum_{i=5}^{13} y_i \tag{5.11}$$

S.T. $f_1(y) = 2y_1 + 2y_2 + y_{10} + y_{11} - 10 \leq 0$,

$f_2(y) = 2y_1 + 2y_3 + y_{10} + y_{12} - 10 \leq 0$,

$f_3(y) = 2y_2 + 2y_3 + y_{11} + y_{12} - 10 \leq 0$,

$f_4(y) = -8y_1 + y_{10} \leq 0$,

$f_5(y) = -8y_2 + y_{11} \leq 0$,

$f_6(y) = 8y_3 + y_{12} \leq 0$,

$f_7(y) = -2y_4 - y_5 + y_{10} \leq 0$,

$f_8(y) = -2y_6 - y_7 + y_{11} \leq 0$,

$f_9(y) = -2y_8 - y_9 + y_{12} \leq 0$,

$0 \leq y_i \leq 1$ (i = 1,2,3,……..9) $0 \leq y_i \leq 100$ (i = 10,11,12) $0 \leq y_i \leq 1$ (i = 13)

Setting the number of generations as 500 and size of the population to be 50 in iTLBO algorithm proven to have the best result for given problem is premeditated and shown below in Table 5.1. iTLBO took much less numbers of function evaluations to evidence the superiority over the considered optimization approaches.

Table 5.1: Optimization results on benchmark function 1

Optimization	Best solution	Function evaluations
PSO	– 15	350,000
ABC	– 15	240,000
TLBO	– 15	100,000
iTLBO	– 15	90,000

- Benchmark test function 2

For this we are taking a minimization nonlinear problem which is having four nonlinear equality constraints and total seven design variables. As far as the number of active constraint are considered at the optimum point, there will be total two, which finally gives the global minimum at y^* = (2.330499, 1.951372, – 0.4775414, 4.365726, – 0.6244870, 1.1038131, 1.594227) and have single objective functional value given as f(y^*) = 680.6300573. The problem may be defined using the following equations.

$$\text{Min f(y)} = (y_1 - 10)^2 + 5(y_2 - 12)^2 + y_3^4 + 3(y_4 - 11)^2 + 10y_5^6 + 7y_6^2 + y_7^4 - 4y_6y_7 - 10y_6 - 8y_7 \quad (5.12)$$

S.T. $f_1(\text{y}) = -127 + 2y_1^2 + 3y_2^4 + y_3 + 4y_4^2 + 5y_5) \le 0$,

$f_2(\text{y}) = -282 + 7y_1 + 3y_2 + 10y_3^2 + y_4 - y_5) \le 0$,

$f_3(\text{y}) = -196 + 23y_1 + y_2^2 + 6y_6^2 - 8y_7 \le 0$, $f_4(\text{y}) = 4y_1^2 + y_2^2 - 3y_1y_2 + 2y_3^2 + 5y_6 - 11y_7 \le 0$, $-10 \le y_i \le 10$ (i = 1, 2, 3, 4, 5, 6, 7)

Setting the number of generations as 2000 and size of the population to be 50 in data driven proposed algorithm, proven to have the best result for given problem is premeditated and shown in Table 5.2. iTLBO took much less numbers of function evaluations to evidence the superiority over the considered optimization approaches. Furthermore, these two benchmark test functions proved that the proposed iTLBO method gives stable and precise algorithmic solutions. Hence, we may go for a decent describing function to use this method in self-balancing and self-controlling of a 2DOFBB system.

Table 5.2: Optimization results on benchmark function 2

Optimization	Best solution	Function evaluations
PSO	680.630	350,000
ABC	680.634	240,000
TLBO	680.630	100,000
iTLBO	680.63006	90,000

- Benchmark function 3

A maximization nonlinear problem having one nonlinear equality constraint and ten design variables are used. At the optimum point there is one active constraint given the global maximum at y^* = (1/√h, 1/√h, 1/√h, ……..) having one objective function value f(y^*) = 1. The inequality constraint shown as |h| ≤ ε, which is converted from equality constraint, where ε = 0.001.

$$\text{Max f(y)} = (\sqrt{h})^{h} \prod_{i=1}^{h} y_i \tag{5.13}$$

S.T. $g(y) = \sum_{i=1}^{4} y_i^2 - 1 = 0$

Where, h = 10 and $0 \le y_i \le 10$ (i = 1,2,3,……..h)

Setting the maximum number of generations up to 2000 and size of population as 50 in improved TLBO algorithm, with best result for problem is calculated and shown in Table 5.3. iTLBO requires less function evaluations to prove its superiority over the other considered optimization methods.

Table 5.3: Optimization results on benchmark function 1

Optimization	Best solution	Function evaluations
PSO	0.9	350, 000
ABC	1	240, 000
TLBO	1	100, 000
iTLBO	1	90, 000

- Benchmark function 4

A minimization linear problem having three linear inequality constraints, three nonlinear inequality constraints and eight design variables are used. At the optimum point there are three active constraint given the optimum solution at y^* = (286.165, 579.3066, 5109.9707, 1359.9709, 5109.9707, 295.601, 182.0177, 217.982, 395.6012) having one objective function value as f(y^*) = 7049.248021. The inequality constraint shown as |h| ≤ ε, which is converted from equality constraint, where ε = 0.000001.

$$\text{Min f(y)} = y_1 + y_2 + y_3 \tag{5.14}$$

S.T. $f_1(y) = -1 + 0.0025(y_4 + y_6) \le 0$

$f_2(y) = -1 + 0.0025(y_5 + y_7 - y_4) \le 0$

$f_3(y) = -1 + 0.01(y_8 - y_5) \le 0$

$f_4(y) = -y_1 y_6 + 833.33252 y_4 + 100 y_1 - 83333.333 \le 0$

$f_5(y) = -y_2 y_7 + 1250 y_5 + y_2 y_4 - 1250 y_4 \le 0$

$f_6(y) = -y_3 y_8 + 1250000 + y_3 y_5 - 2500 y_5 \le 0$

Where, $-100 \le y_1 \le 10000$

$-1000 \le y_i \le 10000$ (i = 2,3)

$-100 \le y_i \le 10000$ (i = 4,5,6,7,8)

Setting the maximum number of generations up to 2000 and size of population as 50 in iTLBO algorithm, with best result is evaluated and shown in Table 5.4. The improved TLBO requires less function evaluations to solve the benchmark function problem.

Table 5.4: Optimization results on benchmark function 4

Optimization	Best solution	Function evaluations
PSO	7049.38	350, 000
ABC	7053.904	240, 000
TLBO	7049.24	100, 000
iTLBO	7049.24	90, 000

5.3 CEC functions used for validating iTLBO

IEEE Congress on Evolutionary Computation (CEC) is a famous benchmark series used to calculate the dominance of optimization algorithms. It is most elaborated platform used for the overall comparison of evolutionary, nature inspire and behavior based algorithms. The benchmark functions approved in CEC are used to evaluate the state of art algorithms. In 2005 IEEE CEC (CEC'05) [328], 25 benchmark functions were included for calculating the fitness landscape. Eight years down the line, in CEC'13 [329] three benchmark functions were approved, while including additional test functions in CEC'05. This count of test function is further increased in CEC'14 [330], where 30 benchmark functions were proposed. In CEC'17 [331], different features were added to already existing 30 benchmark functions. These functions were adopted in CEC'18, CEC'19 and CEC'20 [332]. With time, the intelligence in algorithm was increase and so the level of benchmark functions. CEC'21 [333] included different combination of operators in benchmark functions. These benchmark functions and latest hybrid functions, as approved in latest CEC are used herein to prove the dominance of iTLBO algorithm. Search range for all these functions is $[-100\ 100]^D$, where D is the dimension of the problem. The iTLBO algorithm is implemented on these test benchmark functions and the results are compared with other optimization techniques, as shown in Table 5.5. The test benchmark functions used in this process are [334]:

- **High Condition Elliptic Function:**

$$f_1(x) = \sum_{i=1}^{D}(10^6)^{\frac{i-1}{D-1}} z_i^2 + f_{bias} \tag{5.15}$$

z = M(x – 0) and M stands for a rotational orthogonal matrix which is generated from standard normal distribution using Gram-Schmidt Orto-normalization.

- **Bent Cigar Function:**

$$f_2(x) = z_i^2 + \sum_{i=2}^{D}(10^6)\, z_i^2 + f_{bias}\ , \text{z = M(x-0)} \tag{5.16}$$

- **Discus Function:**

$$f_3(x) = (10^6) z_i^2 + \sum_{i=2}^{D} z_i^2 + f_{bias}\ , \text{z = M(x-o)} \tag{5.17}$$

- **Rosenbrock Function:**

$$f_4(x) = z_i^2 + \sum_{i=1}^{D-1}(100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2) + f_{bias} \qquad (5.18)$$

Where, z = M(2.048(x-o)/100)

- **Weierstrass Function:**

$$f_5(x) = \sum_{i=1}^{D}(\sum_{k=0}^{k} \max\left[a^k\, cos\left(2\pi b^k (x_i + 0.5)\right]\right) - D\sum_{k=0}^{k} \max\left[a^k\, cos\left(2\pi b^k . 0.5)\right]\right) + f_{bias} \qquad (5.19)$$

a = 0.5, b = 3, k(max) = 20, z = M(0.5(x – 0)/100)

- **Rastrigin Function:**

$$f_6(x) = \sum_{i=1}^{D}(z_i^2 + 10) + f_{bias} \text{ , z = M(5.12(x-o)/100)} \qquad (5.20)$$

- **Modified Schwefel Function:**

$$f_7(x) = 418.9829 * D - \sum_{i=1}^{D} g(c_i) + f_{bias} \qquad (5.21)$$

c_i = z_i + 4.209687462275036e+0.02, z = M(1000(x-o)/100)

- **HGBat Function:**

$$f_8(x) = \left|(\sum_{i=1}^{D} z_i^2)^2 - (\sum_{i=1}^{D} z_i)^2\right|^{0.5} (0.5\sum_{i=1}^{D} z_i^2 + \sum_{i=1}^{D} z_i) / D + 0.5 + f_{bias} \qquad (5.22)$$

z = M(5(x – 0)/100)

- **Expanded Scaffer f_5 Function:**

$$f_9(x) = g(x_1, x_2) + g(x_2, x_3) + \cdots + g(x_D, x_1) \qquad (5.23)$$

$$g(x, y) = 0.5\, \frac{(sin^2\left(\sqrt{x^2+y^2}\right) - 0.5)}{(1+0.001(x^2+y^2))^2} \qquad (5.24)$$

z = M(x-o)

- **Hybrid Function 1:**

N = 3, p = [0.3, 0.3, 0.4] (5.25)

g1 : Modified Schwefel's Function $f_7(x)$
g2 : Rastrigin's Function $f_6(x)$
g3: High Conditioned Elliptic Function $f_1(x)$

- **Hybrid Function 2:**

N = 3, p = [0.3, 0.3, 0.4] (5.26)

g1 : Bent Cigar Function $f_2(x)$
g2 : HGBat Function $f_8(x)$
g3 : Rastrigin's Function $f_6(x)$

Furthermore, TLBO and iTLBO algorithms are used to handle the above mentioned benchmark functions. In the process, different parameters were taken into consideration. Note that initial number of students for experiment is to be taken as 50. Maximum number of function evaluations is D*10^4, where D is problem dimensions. For the above mentioned CEC functions, we use D = 30. Moreover, MATLAB scripts are used for programming. TLBO & iTLBO are implemented by applying above mentioned settings. The same settings of parameters are used in [335], where a considerable good survey of above mentioned setting on CEC benchmark functions is available. For comparison analysis, results of other optimization algorithms are taken from [335]. All algorithms are run for 30 times on each CEC test function independently. The mean value of these 30 independent solutions is shown in Table 5.5.

Table 5.5: Mean solution comparison of iTLBO algorithm with optimization algorithms available in literature after implementing on CEC benchmark functions

Functions	PSO	BAT	DE	TLBO	iTLBO
$f_1(x)$	2.44e + 06	3.05e + 08	2.84e + 07	2.75e + 05	1.89e + 06
$f_2(x)$	3.10e + 06	3.12e + 01	1.79e + 01	8.15e – 01	0.88e + 00
$f_3(x)$	5.81e + 02	1.08e + 05	2.21e + 02	4.33e + 04	1.00e + 03
$f_4(x)$	1.10e + 02	3.29e + 03	1.07e + 02	6.64e + 01	1.92e + 01
$f_5(x)$	6.79e + 00	4.63e + 01	2.17e + 01	1.53e + 01	5.02e + 01
$f_6(x)$	5.80e + 01	1.76e + 02	3.26e – 01	6.82e + 01	7.90e – 08
$f_7(x)$	6.38e + 01	2.15e + 02	1.21e + 02	7.31e + 01	3.66e + 01
$f_8(x)$	6.35e – 01	2.78e + 00	9.28e – 01	2.46e + 00	1.55e - 01
$f_9(x)$	2.54e + 00	1.06e + 02	3.24e – 01	2.53e - 01	1.74e - 01
Hybrid 1	3.45e + 05	4.87e + 06	1.48e + 06	1.19e + 05	2.40e + 03
Hybrid 2	2.96e + 03	3.10e + 03	1.13e + 04	2.58e + 03	5.33e + 02

5.4 Describing function and control scheme implementation

A methodology referred to as an extended adaptation of the frequency response describing function is employed to predict, examine, and assess the nonlinear dynamics of a system, generating a limit cycle. Specifically applied to two-degree-of-freedom systems, the describing function is developed through simulation using MATLAB scripts. Its primary objective is to establish the correlation between the ideal voltage and the input motor voltage post-transmission through the DAQ board and amplifier. The 2DOF system model, incorporating the describing function via MATLAB scripts, utilizes a numerical integration method. The simulation model's graphs, obtained through MATLAB, offer a more comprehensive understanding of stability analysis calculations. To achieve this, step response analysis of 2DOF systems is conducted. The controller employed in this configuration is a PID controller, optimized using the improved algorithm (iTLBO) and implemented in a closed-loop control system fashion. The step information characteristics of the system developed i.e., peak undershoot (γ_1), peak overshoot (γ_2), settling time (γ_3) and rise time (γ_4) are calculated using Simulink model. In order to attain stable output for step response, The experimental values derived from graphs representing the aforementioned four characteristics of the nonlinear model should be minimized as much as possible. Hence, to generate the describing function used in the proposed minimization process, a combination of code and Simulink is processed to calculate the describing function J(x) used in this process as:

$$J(x) = \min[\min(\gamma_1) + \min(\gamma_2) + \min(\gamma_3) + \min(\gamma_4)] \tag{5.27}$$

First step is to calculation of error signal i.e. e(s) of closed loop control block diagram of 2DoF systems, which may be written as:

$$e(s) = R_d(s) – R(s) \tag{5.28}$$

where, $R_d(s)$ is the desired set point and $R(s)$ is the output in the form of self-control and self-balancing of balancer plate. PID controller is implemented after going through the optimization process using iTLBO, on this error signal. In frequency domain of 2DOF system, optimized PID controller is implemented where $k_{d,initial}$, $k_{i,initial}$ and $k_{p,initial}$ are initial derivative, initial integral and initial proportional gains used to control the servo angle, pitch angle and yaw angle of 2DoF systems, for which the initial operating points are given as:

$$\beta_d(s) = \left(k_{d,initial}.s + \frac{k_{i,initial}}{s} + k_{p,initial}\right)[R_d(s) – R(s)] \tag{5.29}$$

The Ziegler-Nichols (ZN) tuning method is initially employed to calculate the initial parameters, and subsequently, the controller gains are determined in each iteration of the iTLBO algorithm, as detailed in the preceding chapters of this thesis. This is then succeeded by the computation of describing function values according to equation number (5.27). The optimization of PID gain values is pursued through the following steps, meticulously monitored within each iteration of the iTLBO algorithm using a MATLAB script:

i. Three constrained parameters of the considered problem as k_d, k_i and k_p are defined along with their upper and lower bound.

ii. Size of maximum iteration and Number of learners are defined.

iii. The Teaching Phase starts where mean of each subject is calculated and teacher is identified.

iv. Mean of each subject is updated and new teacher is identified by solving the equations in this sequence – eq. (5.2), (4.21), (5.3) and (4.23).

v. Describing function is used to check the new solution using updated value through eq. (5.27).

vi. In the teaching phase, randomly select two learners and compare them as given in eq. (5.8).

vii. Follow eq. (5.9) and (5.10) to update the new solution, when one learner is better than other.

viii. Use describing function to validate the new solution through eq. (4.23) and accept it if it is found better.

ix. For the next iteration a new fitness value will be considered.

x. Current iteration terminates.

xi. Identification of teacher and the mean of learners gets updated after each iteration. Continuously use eq. (4.23) to calculate value of three dimensions defined in first step.

xii. Repeat Steps (iii) to (x) until the limit of maximum iterations is reached.

xiii. Final iteration will give the optimized value of three constrained parameters which is the optimized gain value of PID controller.

xiv. Optimized gain value of k_d , k_i and k_p is then used as the constrained parameters for PID controller which is to be implemented on 2DoF systems in further subsections.

5.5 iTLBO based control of 2DoF ball balancer system

The 2DoF ball balancer system is working with dead zone nonlinearity in simulation model on MATLAB platform. Here, actuator dynamics are fed to dead zone nonlinearity of model to generate describing function gain. The simulation model of ball balancer is modified as shown in figure 5.2 and all the fourteen steps mentioned in earlier subsection are followed. The results obtained in the process, are then compared with the existing results. After comparing it is clear that the time response parameters of the ball balancer model are improved, which shows the excellence of iTLBO method on 2DOF ball balancer system.

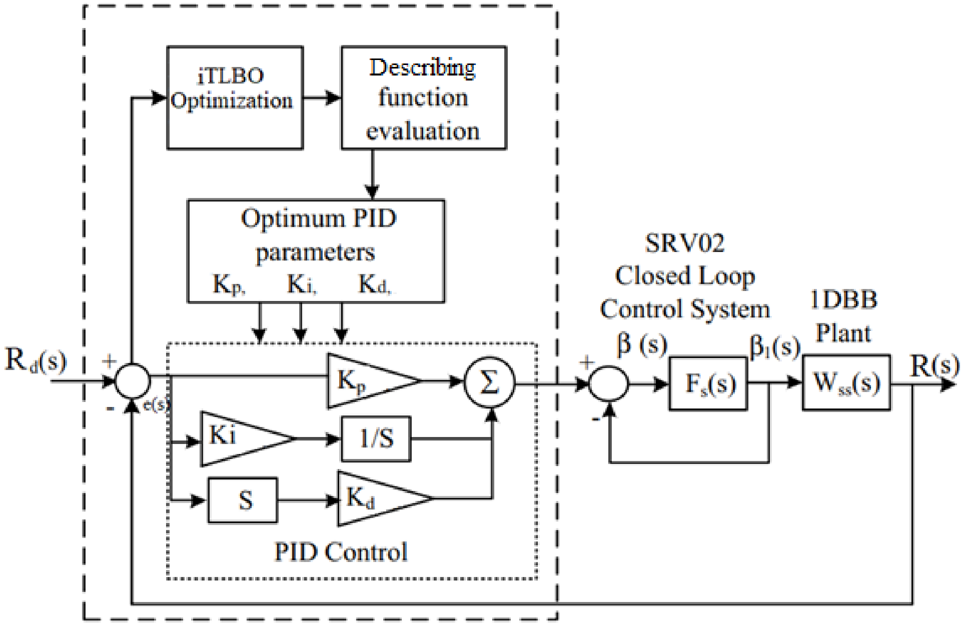


Fig. 5.2 Complete block diagram to represent the angle and position control of a 2DOF ball balancer system using iTLBO optimization method while tuning a PID controller

5.5.1 Numerical Simulation

The improved parameters are mentioned separately in table 5.6 further. The graphical results obtained after this implementation are shown in figure 5.3 & figure 5.4. Figure 5.3 shows the variations of x-axis obtained after implementing PID controller on the simulation model of 2DOF ball balancer system, whereas figure 5.4 shows the variations of same axis obtained after implementing iTLBO method on the modified simulation model, as shown already in figure 5.2.

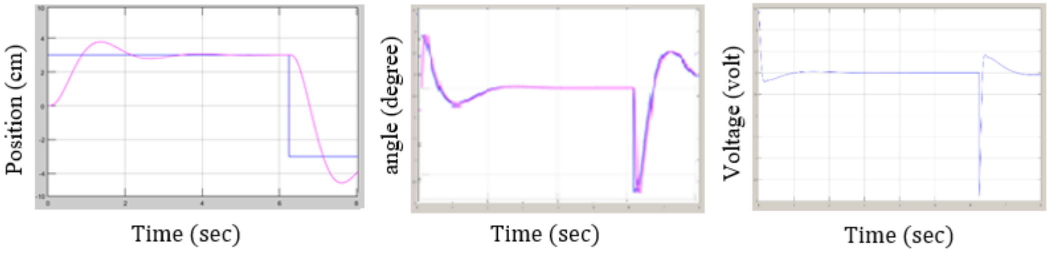


Fig. 5.3 Simulation response of (a) x-axis of ball balancer system after using the PID controller (b) servo angle variations (c) voltage experienced by servo motor.

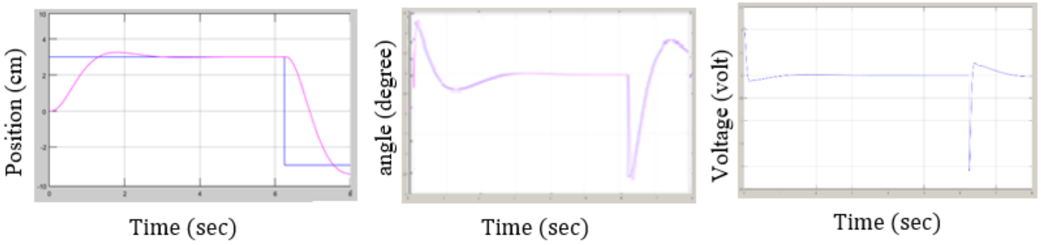


Fig. 5.4 Simulation response of (a) x-axis of ball balancer system after using iTLBO optimized PID controller approach (b) servo angle variations (c) voltage experienced by servo motor

The parameters like peak overshoot, settling time and peak time are improved considerably. The calculations are done after taking the square reference having amplitude of 3 units. The blue line is showing the reference signal and the pink line is showing the output obtained after using the implemented control action. Two different control actions are used, first one is the PID and second one is the proposed iTLBO approach. The peak overshoot is improved from 22.9% to 11.11%, the settling time is improved from 2.76 sec to 1.95 sec and the peak time is improved from 2.4 sec to 1.32 sec. One more important parameter of time response analysis is steady state error. In this case, it refers to the time taken by the system in achieving stability of ball over the balancer plate. This is a highly variable factor in case of balancing type problems. In simulation, it can be calculated in terms of distance (as unit) and by observing the graphs obtained on the simulation model it is found that the steady state error is improved from 1.15 cm to 0.33 cm, which is again very desirable when it comes to controlling a nonlinear system. The servo angle range 6.08 to -12.16 degree by using PID controller, as shown in figure 5.3(b). Whereas the improved range in proposed approach is 4.4 to -8.85 degree, shown in figure 5.4(b). Which proves the stability of servo angle is improved. The voltage range experienced by this servo motor using PID controller is 1.45 to -2.9 volts, shown in figure 5.3(c). This is also improved using proposed approach and turned out to be 1.05 to -2.1 volt, as shown in figure 5.4(c). The root mean square error (RMSE) is also improved in the process, as shown by table 5.6. These calculations are clearly proving the superiority of the proposed approach over the PID controller being used by the researchers.

Table 5.6: Time response parameters and RMSE value

Controller	t_s (sec)	t_p (sec)	M_p (%)	e_{ss} (cm)	RMSE	
					Position	Angle
PID	2.76	2.40	22.90	1.15	5.1894	3.4049
iTLBO	1.95	1.32	11.11	0.33	3.4280	2.5167

5.5.2 Real-time Experiment

This controlling action is achieved by using different controlling approaches. Initially, PID controller is employed to the nonlinear model then using the improved teaching learning based optimization method, gain values are further tuned in order to have the stable steady state operation. The stability is proved already by using time response analysis. PID controller and proposed iTLBO-PID control scheme are implemented on 2DoF ball balancer. The error signal [e(s)] is reduced to minimum and the desired balanced angle and ball position is achieved. The obtained results are further compared with the classical PID controller on the same simulation model using MATLAB/Simulink.

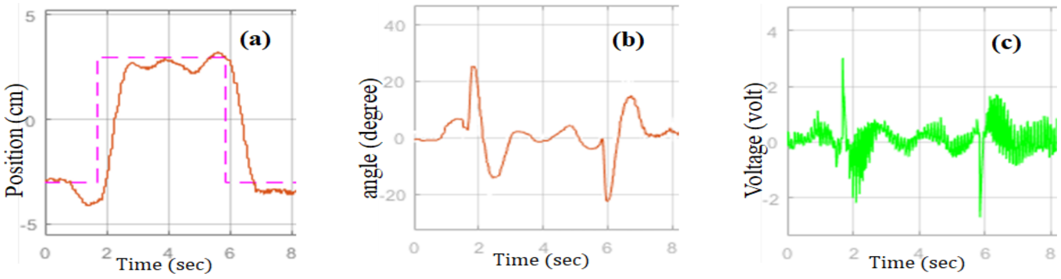


Fig. 5.5 Real time responses of (a) ball position on x-axis (b) servo angle (c) input voltage experienced by servo motor of ball balancer system using PID controller

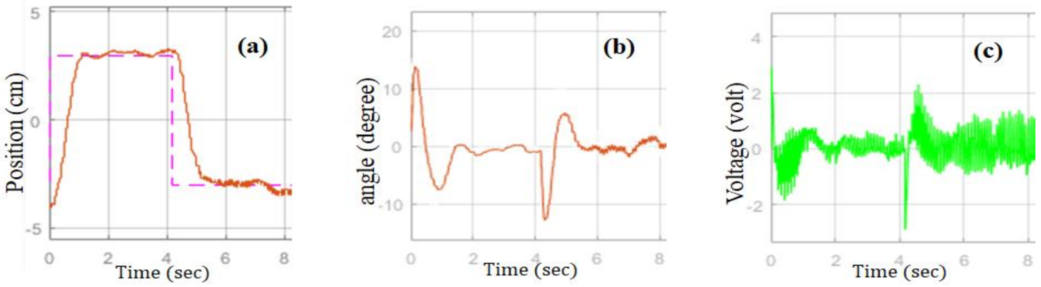


Fig. 5.6 Improved real time responses of (a) ball position on x-axis (b) servo angle (c) input voltage experienced by servo motor of ball balancer system using proposed iTLBO-PID controller

Real-time results indicate that the primary cause of steady-state error in PID was its significant peak overshoot, which is minimized in iTLBO. This reduction contributes to improved settling time, peak time, and reduced root mean square error, as demonstrated in Table 5.6. The performance of proposed iTLBO controller in terms of real time ball position, servo angle and experimental voltage is shown by graphs, as shown in Figure 5.5 & 5.6. In these graphs, variation of servo angles, ball position and voltage is documented over a period of time. It is observed that proposed iTLBO provides stabilized operation for 2DOF ball balancer system by balancing the ball on square plate with nearly zero oscillations visible on the position graph, as shown in fig. 5.5(a) & fig. 5.6(a). The response is recorded with square reference signal using different controllers. Fig. 5.5(b) and 5.6(b) proves the superiority of iTLBO control approach over PID control approach, as the stability angle is varying between –21 to 22 degrees in PID control technique but it is improved in iTLBO technique to –12 to 12 degrees. This signifies that the proposed approach is capable to provide smooth operation during real time working of 2DOF ball balancer model. Furthermore, in Fig. 5.5(c) & Fig. 5.6(c), the voltage action is defined to stabilize the plate angle in order to have a balanced ball on square plate.

5.6 iTLBO based control of 2DoF helicopter system

To implement improved teaching learning based optimization (iTLBO) algorithm while tuning the constraint parameters of proportional-integral-derivative controller of a 2DoF helicopter system, a similar approach is used as it was used to tune the controller parameters of 2DoF ball balancer system. MATLAB SIMULINK platform is used and the blocks are arranged as shown in figure 5.7.

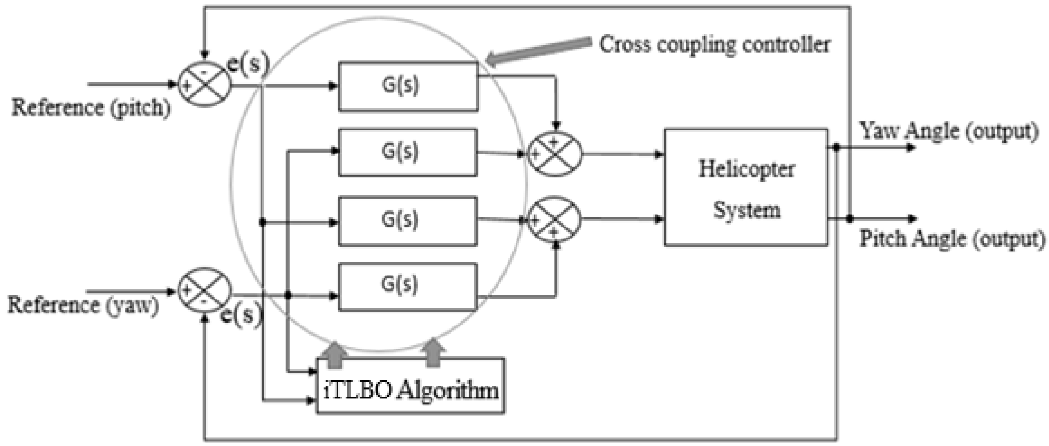


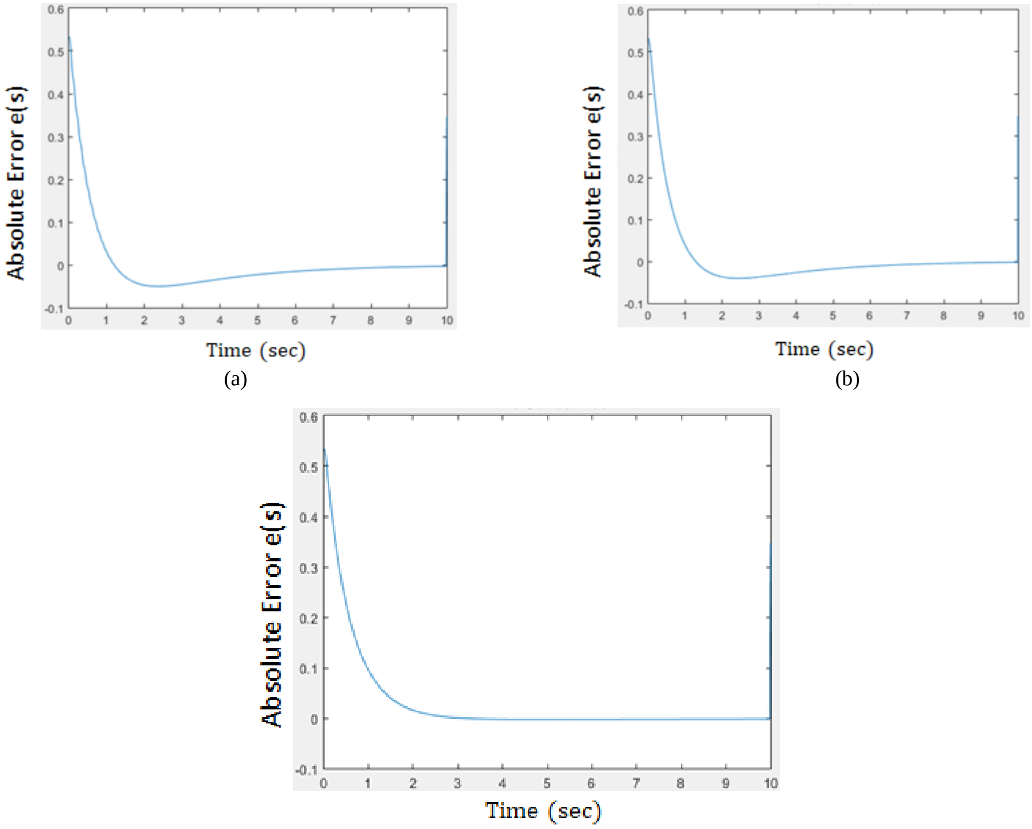
Fig. 5.7 Block diagram representing iTLBO based tuning of PID controller of a 2DOF helicopter

5.6.1 Convergence analysis

Two different PID controllers are used to handle the flight trajectories of 2DoF helicopter system. The controller is imposed on error signal which is calculated as:

$$e(s) = R_d(s) – R(s) \tag{5.30}$$

where, $R_d(s)$ is the desired set point and $R(s)$ is the output in the form of pitch and yaw angle of helicopter model. The controller block G(s) is represented for PID controller. PID controller is implemented while going through the optimization process using LQR method, TLBO and iTLBO optimization technique on error signal. The output R(s), error signals e(s) and set point $R_d(s)$ are calculated separately for both the axis i.e. pitch-axis and Yaw-axis for the helicopter Simulink model.



(a) (b)

(c)

Fig.5.8 Absolute Error vs Time graph obtained by (a) LQR controller [322] (b) TLBO algorithm [63] (c) iTLBO algorithm

Convergence of absolute error graph is obtained using eq. (5.30) by applying three different controlling techniques i.e. LQR, TLBO & proposed iTLBO. Using simulation shown in figure 5.7, signal e(s) is sent to workspace of MATLAB code and then error value is traced using three different controlling techniques. Figure 5.8 shows the path followed by error shown by pitch response of helicopter system when controlled using three controlling approaches i.e. LQR, TLBO & proposed iTLBO. As seen clearly in Fig. 5.8, the error is converged to nearly zero after 7.8 seconds while using LQR controller, got conversed after 4.1 seconds when TLBO optimization method is used and got conversed to nearly zero after 2.9 seconds when iTLBO optimization method is used on the pitch response of helicopter

5.6.2 Numerical Simulation

The impact of proposed algorithm is further discussed in Table 5.7 & 5.8, in which the response parameters i.e. settling time, peak time and peak overshoot obtained from the MATLAB/Simulation results along with the real time results of the helicopter system are shown further in Figure 5.9 and Figure 5.10. It is observed that peak overshoot of LQR is high which causes huge oscillations and made it difficult to handle the external disturbances. Afterwards, by implementing TLBO, peak overshoot is reduced but the external disturbance is still an issue, which creates the need of improved controller. The results of iTLBO-PID controller, shows that not only the peak overshoot is reduced but the oscillations due to external disturbance is also reduced to almost zero and excellent balance of both yaw propeller and pitch propeller is achieved using the proposed method.

Table 5.7: Time response parameters obtained for pitch axis for various controllers during simulation

Method	Pitch Response		
	t_s (sec)	t_p (sec)	M_p (%)
LQR [322]	4.72	2.8	8.99
TLBO-PID	3.99	1.2	4.18
iTLBO-PID	2.83	1.0	0.92

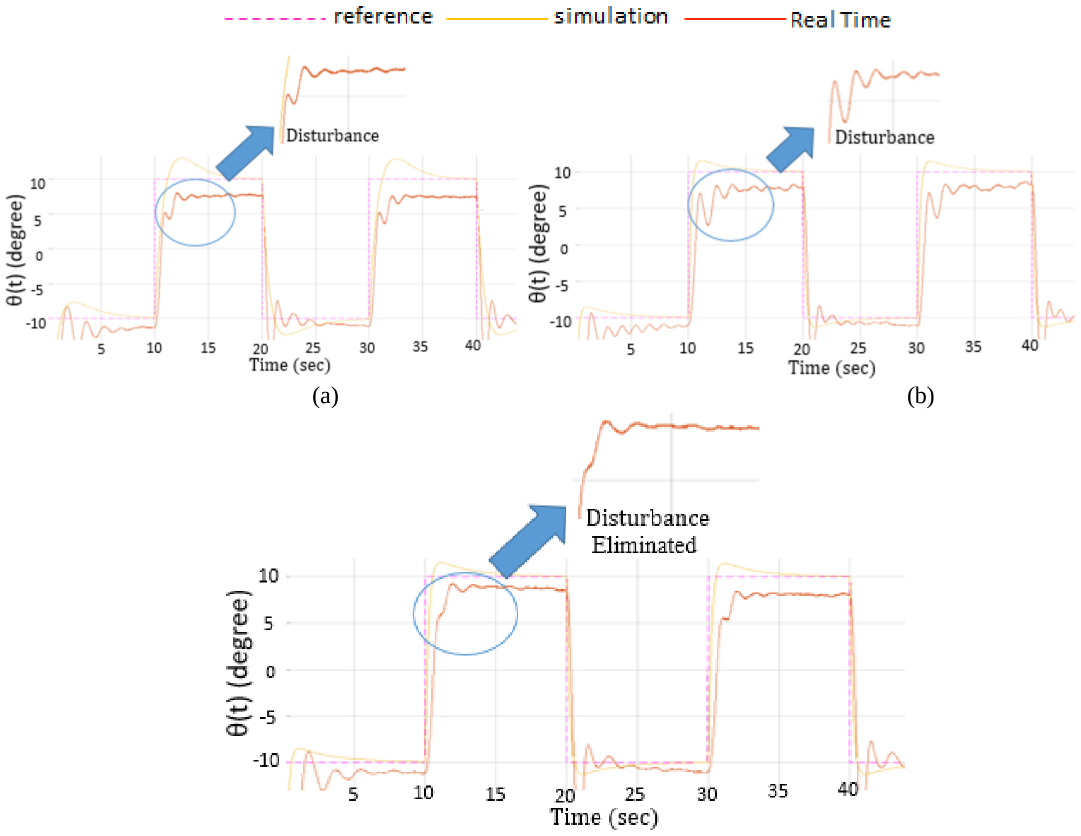
Table 5.8: Time response parameters obtained for yaw axis for various controllers during simulation

Method	Yaw Response		
	t_s (sec)	t_p (sec)	M_p (%)
LQR [322]	5.12	2.1	18.3
TLBO-PID	4.01	0.95	8.19
iTLBO-PID	2.67	0.79	6.23

5.6.3 Real-time Experiment

The performance of proposed controller in terms of improved pitch angle (θ) and yaw angle (Ψ), under the presence of external disturbance (which is created using two Fans) is assessed by the graphs in which variation of these angles is recorded over the time. While placing the fans at different locations near the body of helicopter model, it is observed that orientation of pitch axis changes depending upon the locations of the fans but the pitch and yaw angles gets similar error irrespective of the location of the fans. This is due to the fact that the two propellers of the helicopter model are connected with a fixed base. Hence, only pitch and yaw angles are sensitive to the external disturbances. Further, pitch and yaw trajectories traced using the proposed approach under the influence of external disturbance is compared with other controller's response under the same external disturbance in Fig. 5.9 and Fig. 5.10.

The response is recorded with square reference signal using different controllers. Fig. 5.9(a) and 5.10(a) shows the trajectory of pitch axis and yaw axis respectively, obtained by using the default LQR controller. Fig. 5.9(b) and 5.10(b) shows the trajectory of pitch axis and yaw axis respectively, obtained by using the TLBO optimization method on PID controller. Fig. 5.9(c) and 5.10(c) shows the trajectory of pitch axis and yaw axis respectively, obtained by using the iTLBO controller. The depiction clearly shows that the proposed iTLBO using PID, eliminates the disturbance to almost zero, in real time while improving the working of the helicopter system on MATLAB/ Simulation platform.

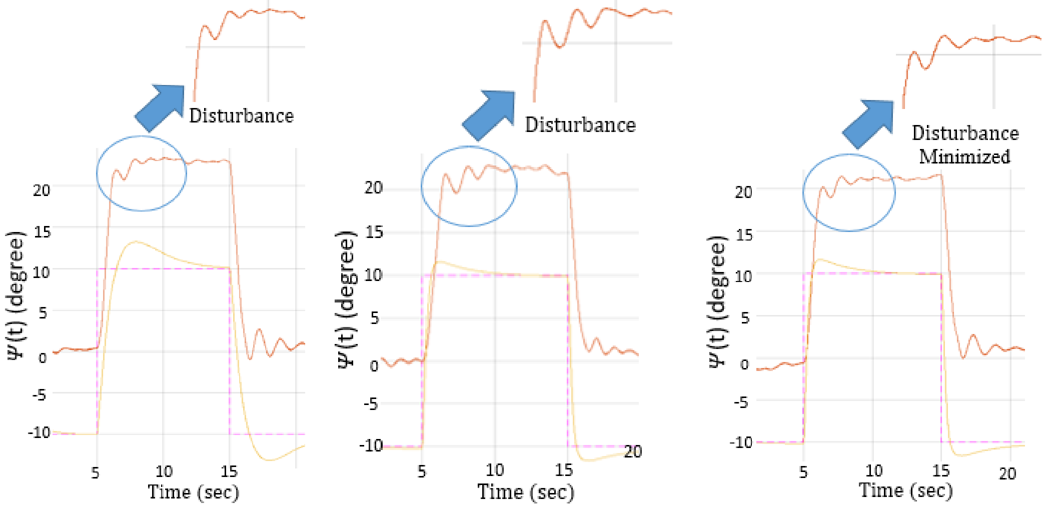


(a)

(b)

(c)

Fig. 5.9 Pitch angle trajectory with the square reference signal during external disturbance (a) default Quanser [322] LQR controller (b) TLBO algorithm (c) proposed iTLBO algorithm



(a)

(b)

(c)

Fig. 5.10 Yaw angle trajectory with the square reference signal during external disturbance using (a) default Quanser [322] LQR controller (b) TLBO algorithm (c) iTLBO algorithm

5.7 Conclusion

In this chapter, improved teaching learning based optimization (iTLBO) algorithm is developed. Two applications of which are demonstrated as – servo angle of two degree of freedom ball balancer system are controlled under the dead zone linearity and pitch axis & yaw axis trajectories of a two degree of freedom helicopter system are controlled under the presence of external disturbance by using iTLBO algorithm on these 2DoF systems. In the process, tuning of gain parameters of proportional integral derivative (PID) controller is performed. Comparing to other optimization techniques, the performance of iTLBO is more satisfactory on the linear and nonlinear benchmark functions. The performance of iTLBO is validated using CEC functions. The results obtained after experimental analysis of helicopter system proved that the developed approach turned out to be superior in eliminating the external disturbance. All the results are validated on simulation platform and gives noteworthy performance within the framework of traditional controller structure. Results obtained are authorized using time response analysis on the MATLAB/Simulation platform. To achieve these results, the describing function is derived using time response parameters only. The proposed controller is proved adaptable and adequate which gives satisfactory performance on helicopter system when tested under the presence of external air disturbance in the laboratory. Proposed iTLBO does not require any algorithmic specific parameters and eliminates randomness from TLBO algorithm, as the result control action is more reliable and achieves remarkable performance while handling yaw axis and pitch axis of helicopter system and maintaining the desired ball position by controlling the servo angle of plate of ball balancer system.

While performing the tuning process, initial parameters are self-employed which are either calculated using traditional method, as discussed in modelling of ball balancer controller, or the use of ZN tuning method is done, as done in handling the initial controller constraints of pitch and yaw axis of helicopter. All these methods involves the required of initial manual input for optimization process. This problem is handled using Fuzzy Inference System as the initial constraint generator for PID controller. This is done while implementing the Fuzzy-PID controller on these 2DoF systems, while using the iTLBO optimization algorithm in further chapter of this thesis. This will further help the system to show robust performance and to be more susceptible to withstand disturbances.

Chapter 6. Hybrid intelligent-classical control for 2DoF systems

6.1 Limitations of classical control strategy and need of hybrid controller

PID controllers, renowned for their simplicity and efficacy, are extensively utilized in control systems. Traditional proportional-integral-derivative (PID) controllers remain the predominant choice across industries for diverse control tasks, owing to their straightforward design, simplicity, and cost-effectiveness in deployment. Nonetheless, standard PID controllers often fall short in effectively regulating systems characterized by added intricacies like time delays, pronounced oscillations (stemming from complex poles with minimal damping), fluctuations in parameters, nonlinear behaviours, and the presence of Multi Input and Multi Output (MIMO) configurations. These limitations have prompted the development of new and advanced tuning methods including Fuzzy Logic, Adaptive Control, Internal Model Control etc. These methods aim to improve the capability and performance of traditional PID controllers while enhancing the flexibility inherent in conventional PID control. Though numerous problems related to PID controller are tackled in control system using intelligent optimization algorithms, like the one used in earlier chapter of this thesis, still there is one limitations which lacks improved real time results on MIMO systems. This problem is the generation of initial constraints of PID controller while formulating the optimization algorithms. This problem may be addressed using a cascaded intelligent-classical controller i.e. Fuzzy-PID controller, as PID controllers have certain limitations comparing with fuzzy PID controllers:

- Challenges with Nonlinear Systems: PID controllers are crafted based on linear system paradigms, rendering them less adept at managing systems imbued with substantial nonlinearities. In contrast, fuzzy PID controllers excel in this domain due to their capability to approximate nonlinear relationships utilizing fuzzy logic.
- Complexity in Tuning: Achieving optimal performance through tuning poses a daunting task for PID controllers, particularly in intricate or dynamically evolving systems. Fuzzy PID controllers present greater tuning flexibility by integrating expert knowledge and linguistic variables, thereby streamlining the tuning process.
- Robustness: PID controllers may weaken in maintaining robustness when confronted with uncertainties and disruptions within the system. Conversely, fuzzy PID controllers bolster robustness by incorporating adaptable fuzzy logic rules tailored to varying operational conditions.
- Management of Complex Systems: PID controllers may prove inadequate in orchestrating control performance for systems characterized by intricate dynamics or interconnected variables. Fuzzy PID controllers excel in handling such complexity by flexibly capturing variable relationships in an adaptive manner.
- Integration of Expert Knowledge: In scenarios where expert insight plays a pivotal role in control strategy formulation, fuzzy PID controllers offer a

distinct advantage. They facilitate the integration of qualitative expert knowledge via linguistic variables and fuzzy rules, thereby enhancing control effectiveness.

- Adaptability: Unlike PID controllers, which maintain fixed parameters post-tuning, fuzzy PID controllers exhibit adaptability by adjusting control strategies based on real-time feedback. This feature renders them better suited for systems with fluctuating dynamics or uncertainties.

In summary, while PID controllers are revered for their simplicity and effectiveness in control systems, fuzzy PID controllers provide notable advantages in managing complex systems, incorporating expert knowledge, and ensuring robustness amidst uncertainties and disturbances. One fuzzy-PID (FPID) controller is used to optimize the controller action of 2DoF systems, which is further discussed in this chapter.

6.2 Optimized FPID control mechanism

As discussed above, a fuzzy PID controllers offer significant advantages due to their inherent structure. By employing analytical formulas within the fuzzy control law, designers can seamlessly integrate these controllers into real-time systems, such as 2DoF system, without encountering computational burdens, thanks to minimal computational delays. This makes them particularly well-suited for rapid processes. Additionally, their self-tuning capabilities render them suitable for handling non-stationary processes. The growing acceptance of fuzzy PID controllers in various industrial sectors underscores their efficacy. In exploring analytical formula-based fuzzy PID controllers, a preference for triangular membership functions is indicated due to their computational convenience across diverse input combinations. Nonetheless, alternative membership functions remain open for exploration, as the one we used in this work i.e. Gaussian membership function. Moreover, enhancing the granularity of fuzzy sets for input and output variables can improve accuracy and facilitate more precise corrective actions. Additionally, employing optimization techniques enables optimal tuning of controller gains. This portion is explored deeply in current chapter.

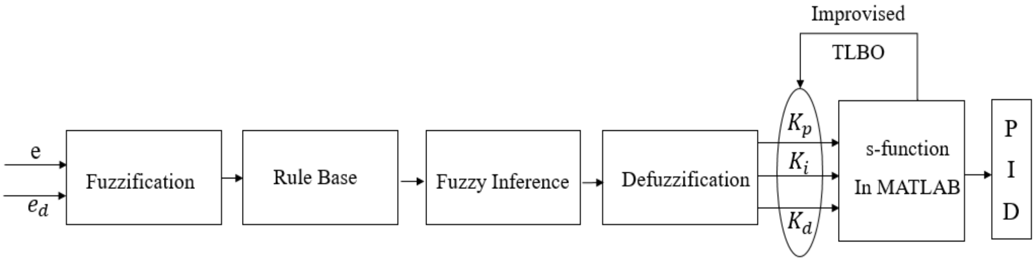


Fig. 6.1 Hybrid Fuzzy+Proportional+Integral+Derivative (FPID) control mechanism

The proposed control action employs a hybrid approach, as illustrated in Figure 6.1. Here, 'e' represents the error signal, while 'e_d' represents its derivative, aiding in understanding the rate of change of the error in the control process. These signals are quantized and utilized as subsets in fuzzy rule sets. A fuzzy logic-based rule system generates fuzzy inferences, which are then responsible for establishing three distinct constraints post-de-fuzzification. The output is initially obtained in fuzzy subset domain, but scaling factors are applied to obtain results in the basic domain. These results are processed through an S-function in MATLAB programming, which integrates them into an algorithm aimed at satisfying a predefined cost function. An improved version of the teaching-learning-based optimization method is implemented accordingly. This iterative process continues until the predefined cost function yields minimized error, resulting in stability of required axis of 2DoF nonlinear systems.

Seven distinct sets of linguistic values are applied to input and output fuzzy subsets, namely Big Positive, Big Negative, Medium Positive, Medium Negative, Low Positive, Low Negative, and Zero. A total of 49 fuzzy rules are formulated using these seven sets, as delineated in Table 6.1. These rules are constructed utilizing IF-THEN logic rules, employing Gaussian-shaped membership functions, as depicted in Figure 6.2. This method leverages a knowledge base akin to human thinking, serving as the essence of fuzzy logic control action. The degrees of membership functions range from 0 to 1, with 0 representing complete inconsistency and 1 indicating total consistency. In these logical operations, the error signal—defined as the disparity between the desired and actual values—is considered as the input. "Big Negative" (BN) signifies a large negative angle, while "Big Positive" (BP) denotes significant positive angles.

Table 6.1: Rule set used for fuzzy logic control action

e/e_d	**BN**	**MN**	**LN**	**Z**	**LP**	**MP**	**BP**
BN	BN	BN	BN	BN	BN	MN	Z
MN	BN	MN	MN	MN	MN	LN	Z
LN	BN	LN	MN	LN	Z	Z	LP
Z	MN	MN	LN	Z	LP	MP	MP
LP	LN	Z	Z	LP	MP	LP	BP
MP	Z	LP	MP	MP	MP	MP	BP
BP	Z	MP	BP	BP	BP	BP	BP

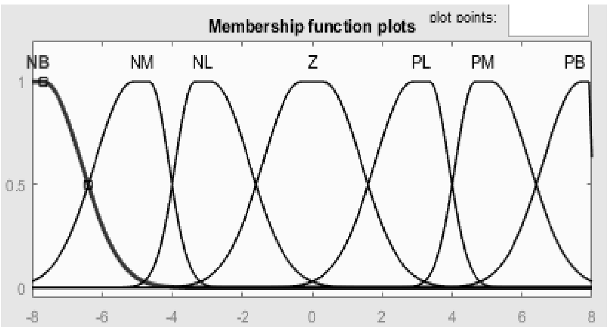


Fig. 6.2 Fuzzy logic membership function

The fuzzy logic rules are designed to guide the system back to the desired angle trend. The coefficients of the proportional-integral-derivative (PID) controller are adjusted using membership functions derived from these fuzzy logic rules. By employing the S-function block in MATLAB, a programming function automates the linguistic level control strategy on a real-time model. This process furnishes empirical insights to fine-tune the servo angle through the fuzzy logic rules. Three distinct outputs are generated, which serve as constraints in the cost function of the optimization algorithm within the S-function block. In this chapter, an enhanced version of the Teaching-Learning-Based Optimization (TLBO) algorithm, previously introduced in the preceding chapter, is utilized to optimize the functioning of hybrid FPID controller. This optimization mechanism is applied to both the 2 Degree of Freedom ball balancer system and the 2Degree of Freedom helicopter system to ensure precise balancing and positioning of the servo angle axis as well as the pitch & yaw axis.

6.3 Hybrid FPID based control of 2DoF ball balancer system

The proposed control optimized method operates with iTLBO algorithm. Initially, fuzzy rule logic is utilized in the first loop, where the response is obtained as constrained parameters. These parameters serve as constraints for the second loop's PID control action. In this subsequent loop, an improved version of the TLBO algorithm is applied to minimize the error on the balancing plate. One loop manages the SRV02, while the other handles the feedback operation in a one-dimensional ball balancer system. A detailed depiction of this approach is presented in Figure 6.3. The error detected in the closed-loop feedback system, along with its derivative, is given to the controller, effectively utilizing it as input to the fuzzy rule-based controller. The output of this controller yields three variable constraints, which are then optimized using an updated TLBO algorithm. This entire process is elaborated extensively in this section. Following the implementation of this approach, stable real-time balancing of the rectangular plate is achieved.

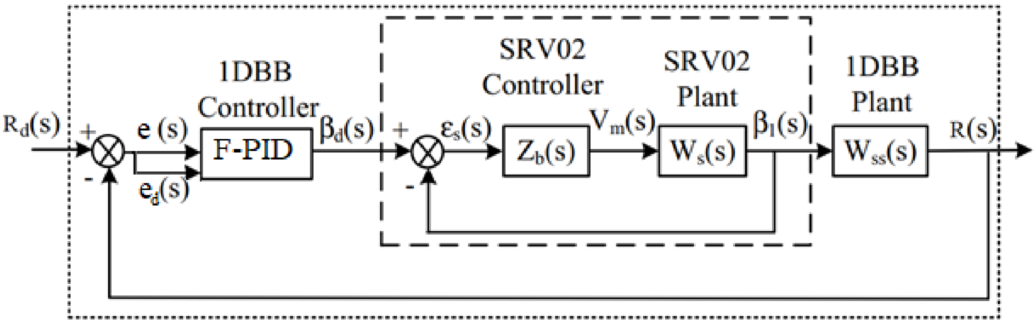


Fig. 6.3 Control mechanism to handle the balancing of 2DoF ball balancer system

6.3.1 Simulated results and comparison

The graphical and mathematical outcomes resulting from the implementation of the proposed approach on this model within a simulation platform are compared with those yielded by the default controller. Consequently, the superiority of the proposed algorithm becomes evident. Table 6.2 enumerates the parameters discussed. These parameters, as defined in section 3 are peak time (T_p), settling time (T_s), peak overshoot (M_p) and steady state error (e_{ss}). The proposed approach implemented on the simulation platform evidently enhances factors contributing to disturbance and instability. In the positioning and balancing of ball over the rectangular plate in this two degree of freedom (2DoF) nonlinear ball balancer system. In addition to evaluating the cost function parameters, both the proposed and default approaches also compute the root mean square error (RMSE). Remarkably, the RMSE is enhanced through the utilization of state-of-the-art methods. This underscores the superiority of the proposed methodology over existing approaches, as illustrated in Table 6.3.

Table 6.2: Time response parameters

Controller	T_p (sec)	T_s (sec)	M_p (%)	e_{ss} (cm)
PID	2.40	2.76	22.90	1.15
FPID-iTLBO	1.11	1.77	9.9	0.21

Table 6.3: Root Mean Square Error

controller	RMSE	
	Position(cm)	Angle (Θ)
PID	5.1894	3.4049
FPID-iTLBO	3.4280	2.5167

The graphical representations are depicted in Fig. 6.4 and Fig. 6.5. In Fig. 6.4, the variations in the plate servo angle for the x-axis of the balancer model are displayed when employing a default PID controller. Conversely, Fig. 6.5 displays the variations in the plate servo angle for the same x-axis of the balancer model, this time utilizing an F-PID controller optimized with the iTLBO.

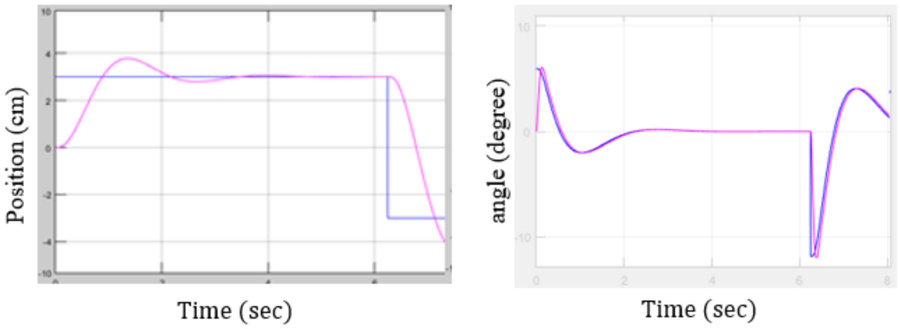


Fig. 6.4 Simulation response of variations experienced by x-axis and servo angle deflection of ball balancer system using the PID controller

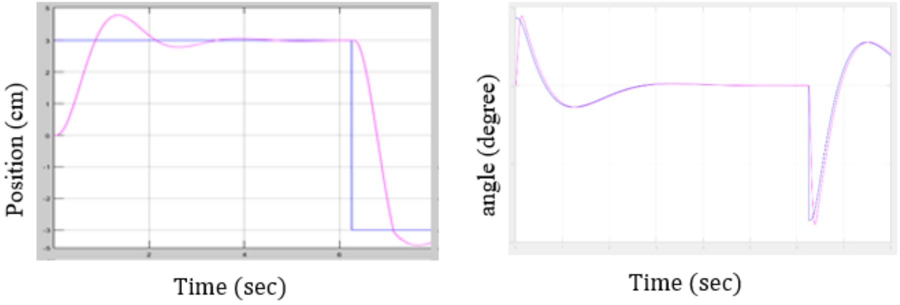


Fig. 6.5 Simulation response of variations experienced by x-axis and servo angle deflection of ball balancer system using proposed approach

Both figures are generated using simulation on the MATLAB platform. Through this implementation, notable improvements are observed in the time response parameters, namely peak time, settling time, peak overshoot and steady-state error. The simulation results utilize a square reference trajectory, represented by the blue lines in both figures. The variations in the plate servo angle are denoted by the pink line. Specifically, the rise time is enhanced from 2.4 seconds to 1.11 seconds, settling time from 2.76 seconds to 1.77 seconds, peak overshoot from 22.9% to 9.9%, and steady-state error from 1.15 cm to 0.21 cm. These compelling mathematical and graphical findings serve to validate the efficacy of the implemented control mechanism. Furthermore, these results are compared with classical technique while working on real time simulator also in the upcoming subsections of this work herein. The ball position error and real time servo angle error are checked in the process. This performance is supported using integral time absolute error, as the improved ITAE is achieved and the same is shown using Table 6.4 and figure 6.6.

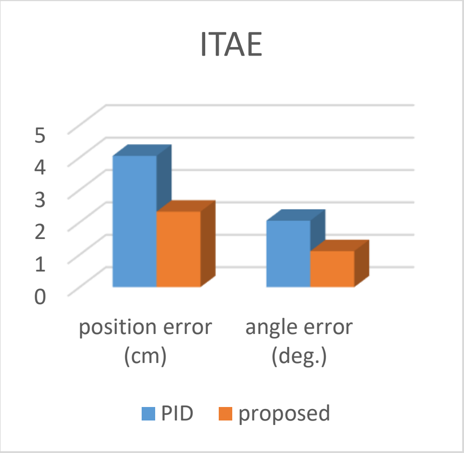


Fig. 6.6 Integral time absolute error observed while using the PID controller and proposed hybrid FPID-iTLBO controller

Table 6.4: Integral time absolute error values

Controller	Position error	Angle error
PID	4.05 cm	2.05 degree
FPID-iTLBO	2.33 cm	1.11 degree

6.3.2 Real-time results

To validate enhanced stability, a comprehensive time response analysis has been conducted. The proposed tuning methodology demonstrates the ability to achieve stable balancing and self-positioning for the hardware model. Through this approach, the minimization of the error signal is achieved, thereby enhancing the system's reliability. Furthermore, the position and angle errors are computed utilizing the proposed approach, employing techniques such as Root Mean Square Error (RMSE) and Integral of Time multiplied by Absolute Error (ITAE). The outcomes of this analysis are compared with those obtained from the classical PID control approach, revealing reduced error rates when employing the proposed controller action during real-time operation. Real-time graphs generated using PID controller and FPID-iTLBO controller are plotted on MATLAB platform. Fuzzy logic enables a flexible and intuitive approach to managing uncertainty in position by representing linguistic variables and rules, allowing the controller to effectively adapt to different position feedback.

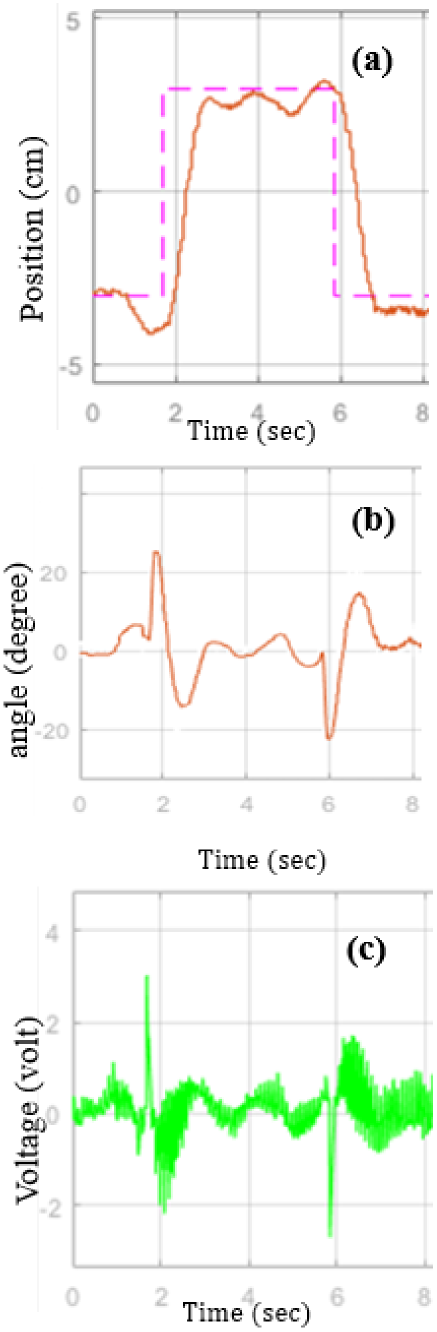


Fig. 6.7 Response of PID controller during real time experiment showing (a) ball position (b) servo angle (c) input voltage signal received by servo motor

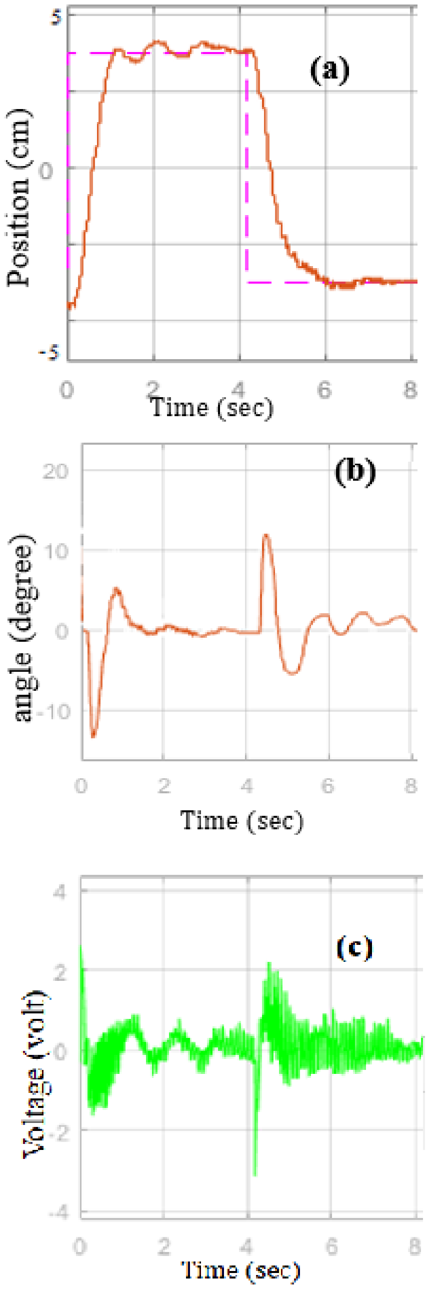


Fig. 6.8 Response of FPID-iTLBO control mechanism during real time experiment showing (a) ball position (b) servo angle (c) input voltage signal received by servo motor

In contrast, the PID control mechanism ensures stability and resilience by continuously adjusting control signals based on error, thereby ensuring precise and timely responses to disturbances. Graphs track the ball position during real-time operation, as illustrated in Figure 6.7 (a) and Figure 6.8 (a). Figure 6.7 corresponds to the operation with the default PID controller, while Figure 6.8 corresponds to operation with the proposed tuning approach of the F-PID controller. Additionally, variations in the plate angle (in degrees) via the servo motor are depicted in Figure 6.7 (b) and Figure 6.8 (b). Furthermore, fluctuations in the voltage signal are observed and graphed in Figure 6.7 (c) and Figure 6.8 (c).

Both scenarios utilize a square reference trajectory. Comparing Figure 6.7(a) and Figure 6.8(a), it is evident that the ball experiences significantly fewer oscillations, contributing to improved stability. Similarly, a comparison between Figure 6.7(b) and Figure 6.8(b) reveals a reduction in the variance of the servo angle of the plate, from (-21 to 22) degrees to (-11 to 11) degrees. This underscores the effectiveness of the proposed tuning method, as it demonstrates the capability to minimize fluctuations in the servo angle. Furthermore, figure 6.7(c) and Figure 6.8(c) illustrate the variations observed in motor voltage during real-time operations, attributable to consistent feedback signals.

6.4Hybrid FPID based control of 2DoF helicopter system

This control technique commences by employing an initial fuzzy logic rule base, with its output subsequently directed to the PID controller as constraints. Following this, the iTLBO method is employed as an optimization technique to reduce error. The schematic depiction of this control method is shown in Fig. 6.9. Here, the error signal and its derivative are input to the fuzzy logic controller, which generates proportional, derivative, and integral constants as output. These constants, combined with the iterative process of the iTLBO algorithm, are then utilized as input for the PID controller. This iterative operation of algorithm along with the FPID controller action is discussed already in earlier subsection of this thesis. Consequently, using this optimization and controlling strategy, the system attains a stable waveform for both yaw and pitch angles.

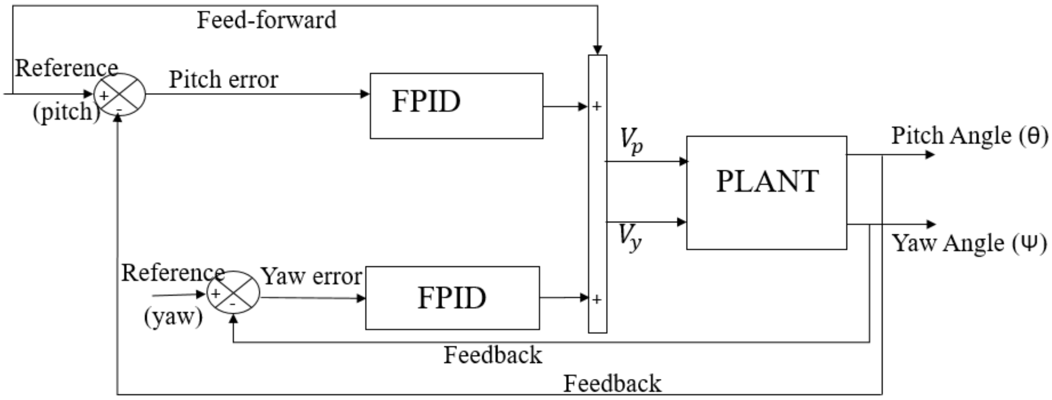


Fig. 6.9 Controller mechanism used to handle pitch and yaw angles of 2DoF helicopter system

Formation of cost function for this problem:

The MATLAB script is employed to formulate the cost function for a nonlinear helicopter system, utilizing higher-order transfer function properties in the time domain. The ABCD matrices are applied to construct a second-order transfer function model using the RGA technique proposed by E. Bristol [325]. The resultant transfer function serves as the plant model in conjunction with a fuzzy PID controller, as depicted in Fig. 6.9. The time domain attributes of the developed system are utilized as benchmarks for the current problem's cost function. MATLAB script calculates these attributes, including rise time (τ_1), settling time (τ_2), peak overshoot (τ_3) and peak undershoot (τ_4). For optimal trajectory determination of the nonlinear helicopter model, the experimental values for these characteristics must be minimized during simulation on the MATLAB platform. A dependable blend of Simulink and coding is employed in the development of the cost function for the nonlinear helicopter model, presented as follows:

$$C(x) = [\min(\tau_1) + \min(\tau_2) + \min(\tau_3) + \min(\tau_4)] \tag{6.1}$$

6.4.1 Convergence analysis

Absolute error is taken for the convergence discussion in the subsection of thesis. The absolute error is shown in fig. 6.9 and discussed in following equation as:

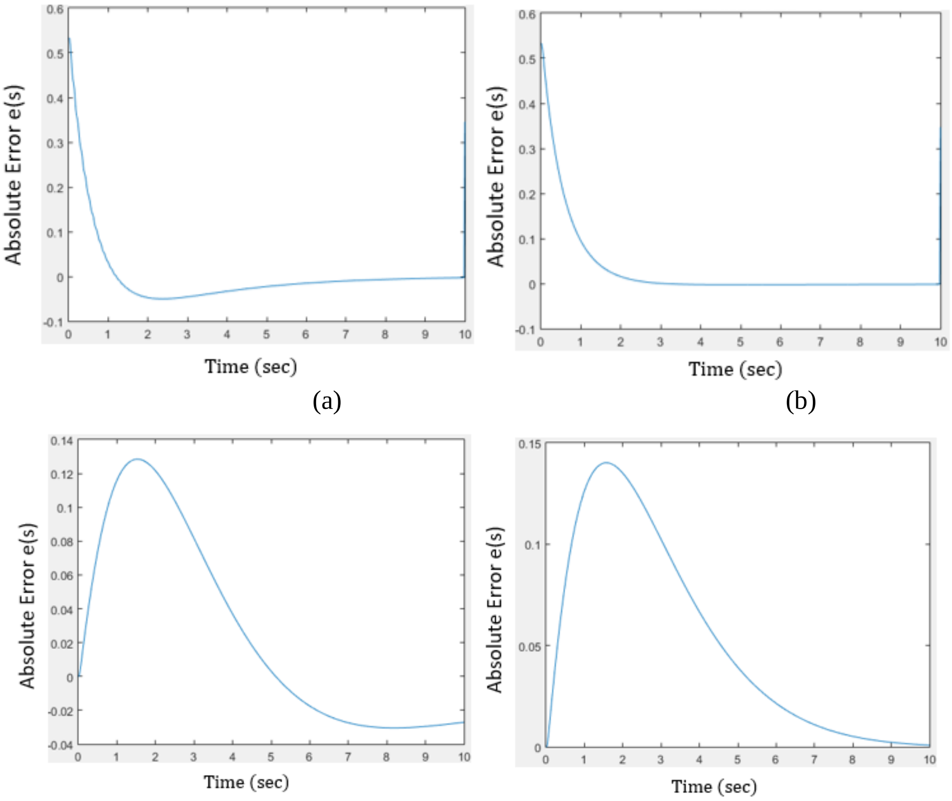
$$\text{Absolute Error } [e(s)] = \text{Desired angle} - \text{actual angle} \tag{6.2}$$

Reference angle provided in the simulation model is termed as the desired angle, while the angle output from the plant is termed as the actual angle. Both these angles are depicted in figure 6.9. Absolute error is computed for both pitch and yaw axes. The pitch angle absolute error and yaw angle absolute error are determined through simulations conducted on the MATLAB platform. Absolute error calculations are performed twice for each axis: first, with the default controller and second, after applying the proposed controller. The signal e(s) depicted in Figure 6.9 is transferred to the workspace and plotted on graphs. Figure 6.10 illustrates four distinct convergence graphs depicting the absolute error convergence observed during real-time simulations of a 2DoF helicopter system.

In Figure 10(a), the absolute pitch error of the helicopter before the application of the fuzzy-PID controller, based on the iTLBO algorithm, is plotted. It is evident that the error converges to nearly zero after 7.8 seconds. Conversely, in figure 10(b), the graph represents the absolute pitch error after applying the proposed controller, demonstrating superior efficiency as the error converges to nearly zero after 2.7 seconds. This underscores the dominance of the proposed controller in managing the pitch axis of the nonlinear helicopter model.

Additionally, figure 10(c) displays the absolute yaw error of the helicopter before the application of the fuzzy-PID controller, where the error fluctuates near zero but fails to converge even after 10 seconds. However, in figure 10(d), the convergence graph of the absolute yaw error after applying the proposed fuzzy-PID controller based on the helicopter model converges to zero after 9.6 seconds. This further confirms the superiority of FPID-iTLBO control strategy in handling the pitch and yaw axes of the

2DoF helicopter model. Moreover, the proposed fuzzy-PID controller, utilizing the improved TLBO algorithm, yields more satisfactory and stable outcomes during the real-time operation of the helicopter system, effectively managing external disturbances and turbulence effects, as shown in next subsection of this thesis.



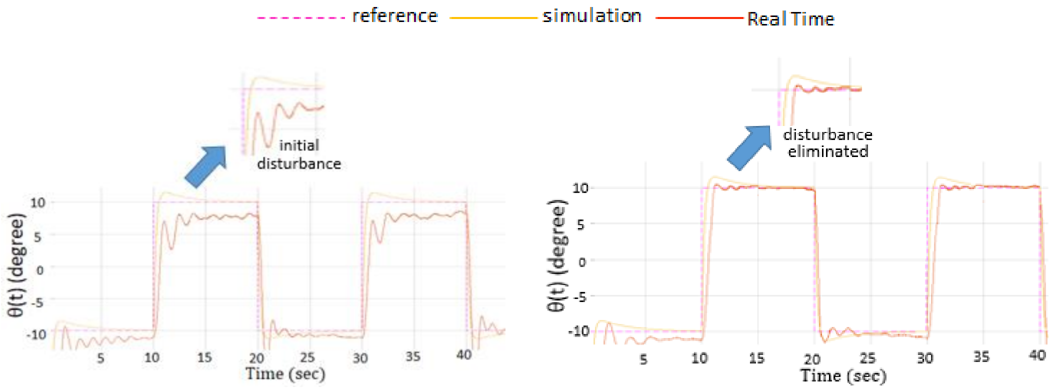
(a) (b)

(c) (d)

Fig. 6.10 Error convergence graph of (a) pitch angle with LQR controller (b) pitch angle with FPID-iTLBO controller (c) yaw angle on LQR controller (d) yaw angle with FPID-iTLBO controller

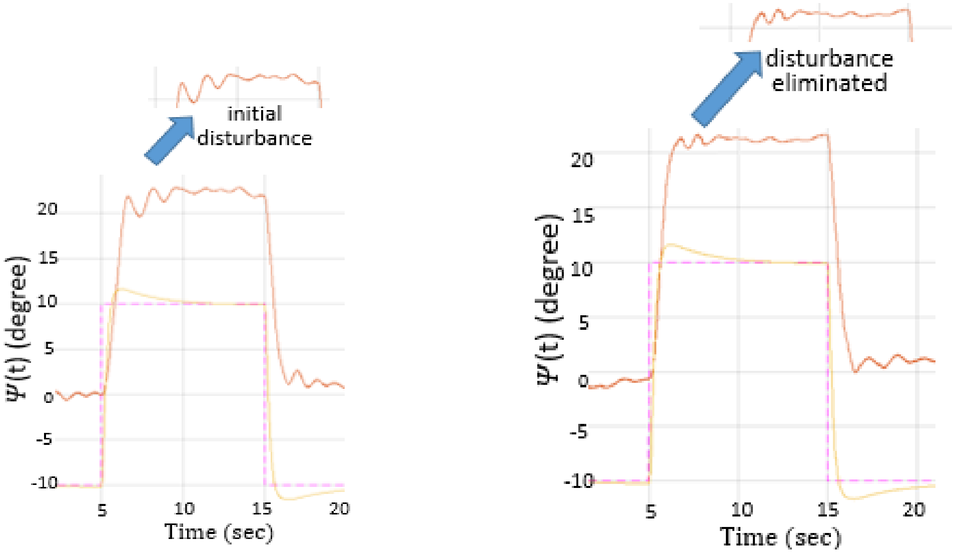
6.4.2 Real-time results

When subjected to turbulence, the pitch angle trajectory is plotted on a graph using the simulation/MATLAB platform using FPID-iTLBO controller & LQR controller. The responses of both controllers are captured with a square reference input signal. The MATLAB simulation platform graphs demonstrate that the FPID-iTLBO controller exhibits superior handling of the pitch trajectory, even in the presence of external disturbances. The comparison between the proposed controller and the default LQR controller for the pitch angle trajectory is depicted in Figure 6.11. In figure 6.11(a), the real-time graph of the pitch trajectory of the nonlinear helicopter system displays the presence of an initial disturbance, whereas in figure 6.11(b), it is evident that the initial disturbance is completely eliminated after implementing the FPID-iTLBO controller. This outcome signifies the reliability of the 2DoF helicopter system's operational model.



(a) (b)

Fig. 6.11 Flight trajectory of pitch angle under the external disturbance using (a) LQR controller and (b) FPID-iTLBO control scheme



(a) (b)

Fig. 6.12 Flight trajectory of yaw angle under the external disturbance using (a) LQR controller and (b) FPID-iTLBO control scheme

Moreover, when subjected to turbulence, the pitch angle trajectory is plotted on a graph using the simulation/MATLAB platform for both the controllers. The responses of both controllers are captured with a square reference input signal. The MATLAB simulation platform graphs demonstrate that the FPID-iTLBO controller exhibits superior handling of the pitch trajectory, even in the presence of external disturbances. The comparison between the FPID-iTLBO controller and the default LQR controller for the yaw angle trajectory is depicted in Figure 6.12. In Figure 6.12(a), the real-time graph of the yaw trajectory of the nonlinear helicopter system displays the presence of an initial disturbance, whereas in Figure 6.12(b), it is evident that the initial disturbance is completely eliminated after implementing the FPID-iTLBO controller.

The simulated results of these graphs are illustrated using table 6.5. In table 6.5, time response analysis are shown for the graphs obtained in fig. 6.11 & fig. 6.12. FPID-iTLBO controller have proved superiority based on settling time (t_s), improved peak time (t_p) and better peak overshoot (M_p), obtained after simulation on MATLAB platform.

Table 6.5: Time response parameters of graphs obtained in Fig. 6.11 & Fig. 6.12

Controller	Pitch Response			Yaw Response		
	t_s (sec)	t_p (sec)	M_p (%)	t_s (sec)	t_p (sec)	M_p (%)
LQR	4.72	2.8	8.99	5.12	2.1	18.3
FPID-iTLBO	2.7	0.9	0.8	3.0	0.6	4.9

6.5 Conclusion

This study focuses on two application of optimized FPID controller in first application, focus is on controlling the flight trajectory of a 2DoF helicopter model under external disturbance induced by two high-speed fans positioned near the helicopter's propellers and in second application, focus is on balancing and positioning a ball over a rectangular plate through servo angle alteration in a 2DoF ball balancer system. To achieve this, a two-step fuzzy-PID controller is employed and optimized using an improved version of the teaching-learning-based optimization algorithm. Two identical controllers are utilized for the pitch axis and the other for the yaw axis of 2DoF helicopter model and the same controller is utilized for servo angle axis control of 2DoF ball balancer. The fuzzy logic component takes error and derivative of error as inputs and generates three distinct constraints as outputs for the second step of the controller, namely the PID controller.

The optimization algorithm developed in previous chapter of this thesis is used to adjust these constraints through iterative calculations while minimizing the error signal effectively. The proposed control scheme demonstrates satisfactory performance, as discussed comprehensively in this chapter. Furthermore, comparison with existing literature results are conducted. The verification and validation of the proposed controller's outcomes are extensively scrutinized. Verification is carried out using MATLAB/Simulink models, while the superiority of the proposed approach is validated by referencing relevant literature. This proves that the optimization algorithms are a good tool to provide stable results while tuning the intelligent as well as classical controller. This gives further scope of use of more strong and quick optimization technique, which may be used to optimize cascaded intelligent-classical controllers. Such optimization techniques may be obtained by hybridizing more than one optimization algorithm.

Hybrid optimization techniques present numerous benefits in addressing complicated optimization problems by merging the strengths of more than one optimization methodologies. Through the integration of more than one algorithms, they effectively navigate solution spaces, resulting in improved convergence rates and solution quality. Such hybrid approaches capitalize on the range of algorithms available to tackle various surfaces of optimization challenges, including exploring and exploiting the search space, managing multi-modal or non-linear functions, and overcoming local optima. Furthermore, they possess the capability to dynamically transition between different algorithms or strategies based on the characteristics of the problem or the progress of convergence, thus ensuring flexibility and adaptability across a wide array of optimization scenarios. Ultimately, hybrid optimization algorithms serve as versatile and efficient tools for addressing complex optimization tasks, offering sharp performance and scalability compared to approaches dependent on a single methodology. One such hybrid optimization technique is developed in next chapter of this thesis, which is further used to handle the balancing and positioning of 2DoF nonlinear systems.

Chapter 7. HGPCTLBO: A hybrid algorithm, based optimized control of 2DoF systems

7.1 Idea of HGPCTLBO

Over time, heuristic and metaheuristic techniques have gradually supplanted classical control methods, primarily because conventional approaches like PID and fuzzy logic controllers (FLCs) demand expertise in parameter tuning from the controller designer. Unlike classical methods, metaheuristics alleviate this requirement. Pairing metaheuristic optimization algorithms with Classical PID has shown enhanced performance compared to traditional PID alone, as discussed deeply in literature survey of this thesis. It's evident that various heuristic and metaheuristic optimization techniques have been adopted to address the challenges posed by nonlinear under-actuated systems over the years. Initially, classical and subsequently intelligent controllers were employed to create efficient environments for multivariable models. TLBO & GPC [49, 50] algorithms are discussed deeply in chapter 3 of this thesis. The TLBO algorithm is based on the interaction dynamics observed between a teacher and a student in a classroom environment, with each role serving as crucial elements of the algorithm. Metaheuristic optimization algorithms rely on parameters specific to each algorithm. Reducing such parameters can lead to decreased complexity and improved efficiency of the algorithm. The Teaching-learning-based-optimization (TLBO) algorithm excels in such contexts. The Giza pyramid construction (GPC) algorithm is a recent addition to the family of metaheuristic algorithms, drawing inspiration from the ancient construction techniques used in building the great pyramids of Egypt. Over time, GPC has demonstrated superior performance on various fitness functions compared to other metaheuristic approaches documented in the literature. GPC operates on a population-based approach, mirroring the movement of workers along ramps to elevate stone blocks.

Eventually, optimization algorithms assumed control scheme optimization, fine-tuning controller parameters to yield optimal local and global solutions. This evolution saw the emergence of techniques such as Cuckoo search, Krill herd, Fire-fly, Jaya, Gray wolf, among others [336 – 340], each contributing optimal solutions as discussed in literature survey. The amalgamation of these theories is becoming increasingly popular as hybrid methods consistently outperform their original counterparts, yielding more reliable results. Hybrid algorithms, leveraging characteristics from both optimizations, are now extensively utilized to enhance the performance of nonlinear systems, effectively addressing convergence time delays and delivering efficient responses. An example of such a hybrid approach is the Hybrid Giza pyramid construction teaching learning based optimization (HGPCTLBO) algorithm, which combines the teaching phase of TLBO with the updating approach of GPC by integrating the best worker fitness from GPC into the TLBO algorithm's student/learner phase.

A hybrid method combines the most effective elements from both original algorithms. HGPCTLBO adopts a productive approach inspired by both the GPC and TLBO algorithms. Its inspiration stems from the construction techniques employed by workers in building the Giza pyramids, as well as the collaborative nature observed among students in a classroom setting. In this hybrid approach, workers, who are focused on the collective goal of placing all stone blocks at designated locations, initially adhere to commands from a designated authority, such as the Pharaoh, and subsequently follow fellow workers to reach their destination. This methodology is structured into two phases: the “command phase” and the “cooperate phase”, as depicted in Figure 7.1 using a flowchart and elaborated upon in the subsequent subsections.

7.1.1 Command phase

During the initial phase, all workers will adhere to the guidance of a designated agent representing the Pharaoh. This agent is a specially selected worker positioned optimally relative to the destination. Workers will transport stone blocks according to the instructions provided by the Pharaoh's agent. The manner in which workers execute these instructions will define the control strategy employed by the current algorithm during this phase.

The workers are tasked with relocating the stone blocks and transferring them to the ultimate destination via a ramp with friction. This displacement process is defined by:

$$d = \frac{v_0^2}{2g(\sin\theta + \mu_{kf}\cos\theta)} \tag{7.1}$$

The initial positions of the blocks, along with their associated costs, are already determined. The workers aim to locate the most optimal position to efficiently relocate the stone block. The displacement of the workers is expressed as:

$$x = \frac{v_0^2}{2g\, sin\, \theta} \tag{7.2}$$

The derivation of equation (7.1) and equation (7.2) has been thoroughly explained in chapter 3 of this thesis. The updated position or solution of the algorithm will be determined based on the displacement of the workers and the solution or position of the Pharaoh's agent, as described by the following equation:

$$\overrightarrow{W_{new}} = \left(\overrightarrow{W_p} + d\right)x \times (\overrightarrow{W_p} - \overrightarrow{W_\iota})\, \overrightarrow{\epsilon_\iota} \tag{7.3}$$

Here, $\overrightarrow{W_{new}}$ represents the new position of the worker while $\overrightarrow{W_p}$ & $\overrightarrow{W_\iota}$ represents the Pharaoh’s agent position and initial position of worker respectively. After this, possibility of substitution will be investigated as discussed in sub-section 3.2 and given here as:

$$\zeta = \begin{cases} \psi, if\ rand[0,1] \le\ 0.5 \\ \varphi, otherwise \end{cases} \tag{7.4}$$

At the conclusion of this stage, the workers' positions will have been adjusted as instructed by the representative of Pharaoh. In the subsequent phase, the workers will collaborate with one another.

7.1.2 Co-operate phase

During this stage, workers will assist one another in reaching the ultimate goal. By the conclusion of the command phase, each worker will have been assigned specific positions, resulting in each group having a distinct solution. Some groups may possess a greater quantity of solutions, while others may have fewer. This distribution will serve as a benchmark for the cooperative phase.

Here, a worker will look to a nearby colleague who is engaged in working on a different stone block. If the cost function solution of the colleague is superior to that of the current worker, the latter will accompany the former to advance closer to the final destination. This process will update the current worker's position according to the following equation:

$$\overrightarrow{W_{new}} = \overrightarrow{W_\imath} + (\overrightarrow{W_f} - \overrightarrow{W_\imath})\, \overrightarrow{\epsilon_\imath} \tag{7.5}$$

$$\overrightarrow{W_{new}} = \overrightarrow{W_\imath} - (\overrightarrow{W_f} - \overrightarrow{W_\imath})\, \overrightarrow{\epsilon_\imath} \tag{7.6}$$

$\overrightarrow{W_f}$ and $\overrightarrow{W_\imath}$ represents the current position of fellow worker and current worker respectively.

The choice of the fellow worker is made randomly, and the criteria for selecting Equation (7.5) and Equation (7.6) depend on whether the problem is of the maximizing or minimizing type. At the conclusion of this phase, the worker's position is adjusted by referencing the position of their co-worker. The worker with the best position thus far will then assume the role of the new Pharaoh's agent, signalling the termination of the algorithm for the next iteration. The appropriate pseudo code for implementing the aforementioned two phases is detailed in the following subsection. This entire process is conducted with the following assumptions in consideration:

- The ramp is linear and possesses a smooth, level surface.
- A single ramp is employed.
- The inclination relative to the horizontal must not exceed 15 degrees.
- The outcome is influenced by both the stone block and the worker's position.
- Friction is taken into account solely for the stone blocks, not the workers.
- The workers are interchangeable and can be relocated to different positions.

7.2 Algorithm flow of Hybrid GPC-TLBO

Step by step flow chart of HGPCTLBO algorithm is detailed in figure 7.1 and the pseudo code developed following the command phase and co-operate phase of HGPCTLBO algorithm are mentioned here:

```
Step 1    Generate initial population array as number of Workers
Step 2    Set maximum number of iterations
Step 3    Declare ISE, IAE and ITAE as cost function using Simulink platform
          Command Phase start:
Step 4    For iteration 1 to maximum number of iterations do
Step 5    For worker 1 to maximum number of workers do
Step 6     Calculate position of Workers (cost) by using cost function
Step 7     Declare best worker as Pharaoh's agent
Step 8     Calculate displacement of stone blocks by using equation no. (7.1)
Step 9     Update the cost function and current position of workers using eq. (7.3)
Step 10    Track movement of workers by using equation no. (7.2)
Step 11    Investigate possibility of substitution using eq. (7.4)
           Co-operate Phase start:
Step 12    Select a worker and his fellow worker randomly
Step 13    If position of fellow worker is better than position of current worker
Step 14    Update the cost function and position of current worker using equation
           (7.5) & equation (7.6)
           End if
           End for loop (for workers)
           End for loop (for iteration)
```

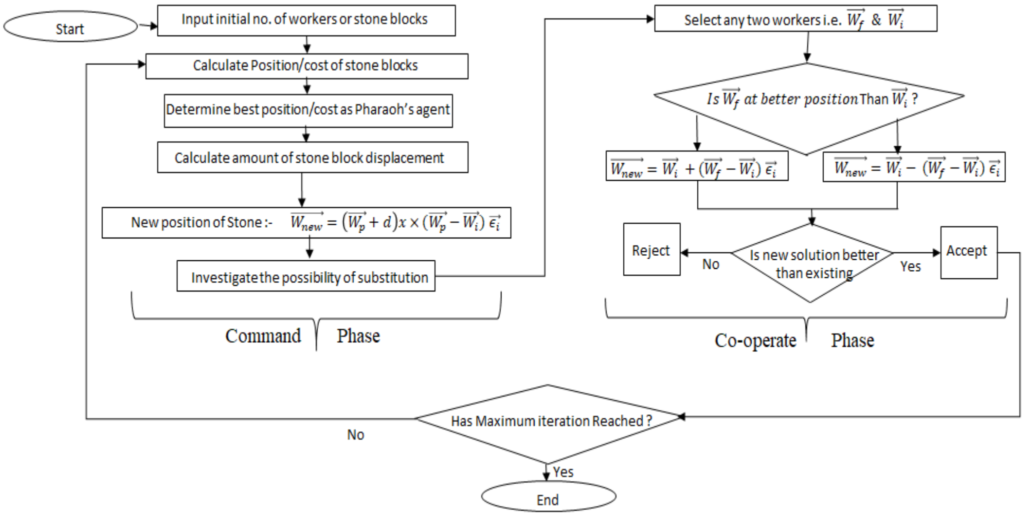

Start
Input initial no. of workers or stone blocks
Calculate Position/cost of stone blocks
Determine best position/cost as Pharaoh's agent
Calculate amount of stone block displacement
New position of Stone :- $\overrightarrow{W_{new}} = (\overrightarrow{W_p} + d)x \times (\overrightarrow{W_p} - \overrightarrow{W_i})\,\overrightarrow{\epsilon_i}$
Investigate the possibility of substitution
Command Phase
Select any two workers i.e. $\overrightarrow{W_f}$ & $\overrightarrow{W_i}$
Is $\overrightarrow{W_f}$ at better position Than $\overrightarrow{W_i}$?
$\overrightarrow{W_{new}} = \overrightarrow{W_i} + (\overrightarrow{W_f} - \overrightarrow{W_i})\,\overrightarrow{\epsilon_i}$
$\overrightarrow{W_{new}} = \overrightarrow{W_i} - (\overrightarrow{W_f} - \overrightarrow{W_i})\,\overrightarrow{\epsilon_i}$
Reject
No
Is new solution better than existing
Yes
Accept
Co-operate Phase
Has Maximum iteration Reached ?
No
Yes
End

Fig. 7.1 Flow chart of HGPCTLBO algorithm

7.3 HGPCTLBO-PID based control of 2DoF ball balancer system

The PID controller is integrated into the outer loop of the balancer system model, as discussed in chapter 3. This integration facilitates the placement of poles to ensure observable decay and controllable time constants. Initially, the PID controller parameters are computed using the methods outlined in Section 2, employing the Ziegler-Nichols (Z-N) method. Subsequently, the controller is applied to algorithmic functions. The optimization algorithms employed for this implementation include TLBO, GPC, and the proposed HGPCTLBO, enabling a comprehensive comparative analysis of the proposed approach. The primary objective of these algorithms is to determine the ball position and generate a graph illustrating the expected and computed values of the ball position along the axis. Variation in the parameters of the PID controller, utilized as the control term in the balancer model, governs these ball positions. The effectiveness of the optimization algorithms is assessed based on minimizing the potential error, measured using various objective functions. Tracking error is employed to validate the effectiveness of the controller utilized in this control arrangement. The objective functions of the feedback controller utilized to validate the attributes of tracking error include the Integral of the absolute error, the Integral of square error, and the Integral of the time-weighted absolute error, represented as:

$$\text{IAE} = \int e(t)\, dt \tag{7.7}$$

$$\text{ITAE} = \int t|e(t)|\, dt \tag{7.8}$$

$$\text{ISE} = \int e^2(t)\, dt \tag{7.9}$$

In this context, e(t) represents the tracking error derived on the Simulink platform after comparing the reference trajectory with the actual trajectory of the ball position on the plate surface. To assess the enhancement in tracking error control, the tracking errors obtained during the PID controller operation for each of the three algorithms (i.e., GPC algorithm, TLBO algorithm, and proposed HGPCTLBO) are individually acquired and subsequently subjected to the objective functions mentioned above during simulation analysis. The tuning mechanism used in this work is discussed in Fig. 7.2 below.

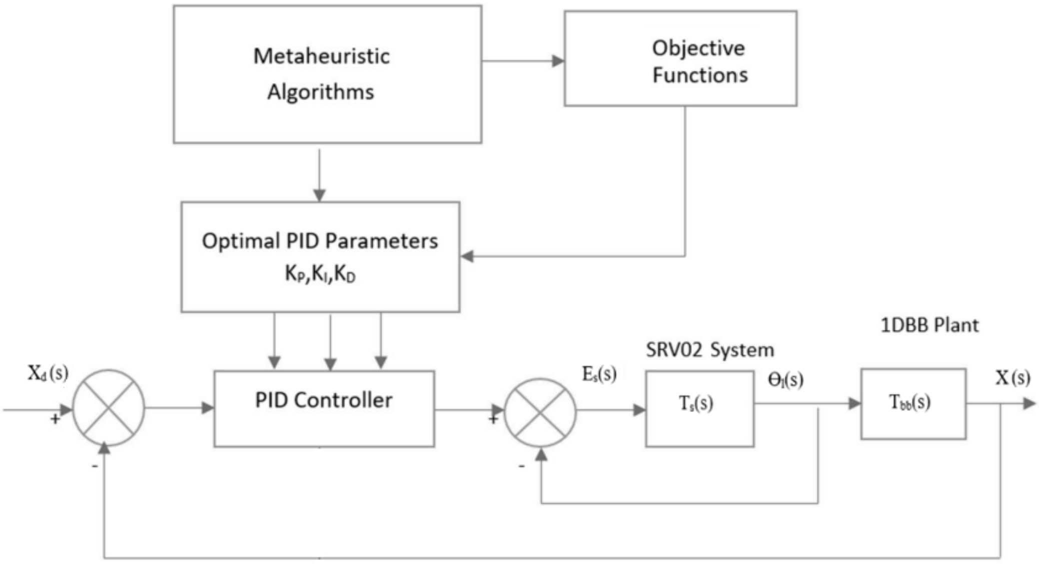


Fig. 7.2 Block diagram showing the tuning mechanism used for constraints of PID controller

7.3.1 Numerical analysis with objective functions

The algorithm utilizes ISE, IAE, and ITAE as the objective functions and will continue iterating until the maximum number of iterations is reached. This process will yield data for analysing the closed-loop stability and relative stability across different optimization methodologies employed. The detailed steps for implementing the proposed hybrid optimization algorithms have already been discussed in previous section of this chapter. TLBO and GPC are executed following their respective methodologies as outlined in references [49] and [50] respectively. Upon reaching the maximum iteration limit of the optimization algorithms, the positions of the ball are tracked and plotted based on the parameter values obtained. The controller outputs continuously adjust the plate angle through a servo motor feedback mechanism to achieve optimal ball position and balanced plate angle. This motion is monitored and plotted on the simulation platform. Figures 7.3, 7.4, 7.5, and 7.6 depict the observed ball tracking during simulation for the PID controller using classical PID, TLBO-PID, GPC-PID and the proposed HGPCTLBO-PID optimization approaches, respectively.

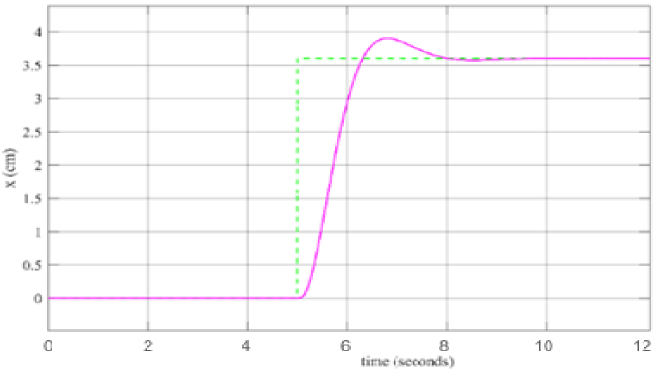


Fig. 7.3 The ball position w.r.t. reference step signal for classical PID controller

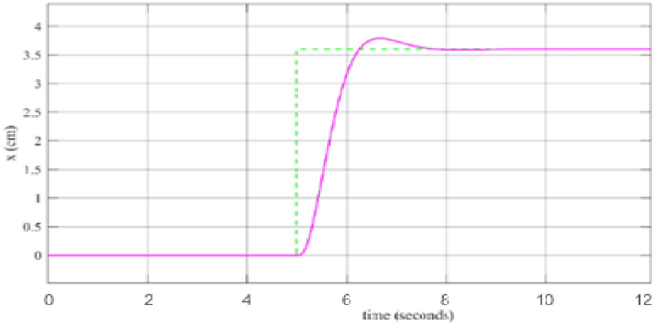


Fig. 7.4 The ball position w.r.t. reference step signal for GPC-PID controller

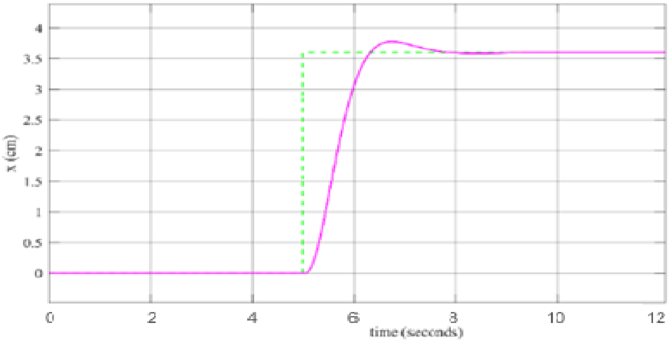


Fig. 7.5 The ball position w.r.t. reference step signal for TLBO-PID controller

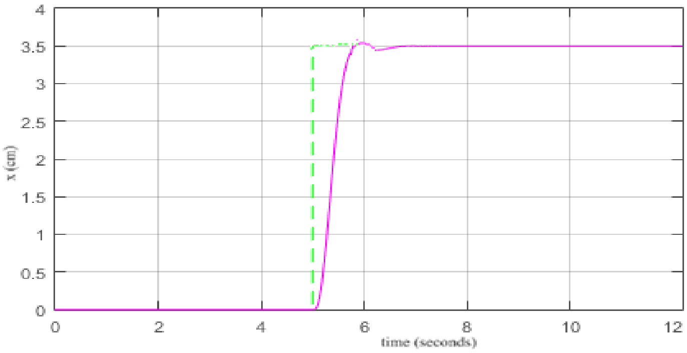


Fig. 7.6 The ball position w.r.t. reference step signal for proposed HGPCTLBO-PID controller

Additionally, the time-domain specifications of the simulation results obtained during the optimization process are detailed alongside the corresponding graphical outcomes of each algorithm. Figure 7.3 illustrates the simulation outcomes for the classical PID controller approach applied to the ball balancer system. These results are optimized since parameter calculations are conducted using the traditional Ziegler-Nichols (ZN) method. In this method, the feedback mechanism's behaviour is not adequately considered over the required duration, resulting in significantly higher observed values for Rise time (T_r), Settling time (T_s) and Peak overshoot (M_p) than expected. Figures 7.4 and 7.5 display the variations in ball positions observed on the simulation platform when the unconstrained parameters of the PID controller are optimized using the GPC and TLBO algorithms, respectively. The time-domain results for the HGPCTLBO-PID control mechanism, depicted in Figure 7.6 and elaborated upon in Table 7.1, demonstrate the superiority of the proposed algorithm over both the classical controller and the latest optimization algorithms. Although the optimization manages to address the settling time and peak time issues of the classical controller to some extent, peak overshoot remains problematic with the TLBO and GPC algorithms. This concern is effectively addressed when employing the hybrid optimization, as indicated in Table 7.1, with improved peak time and settling time, the improved peak overshoot is also obtained to a satisfactory extent.

Table 7.1: Time domain specifications comparison of all optimization algorithms used above

Controller	T_r (sec)	T_s (sec)	M_p (%)
Classical PID	2.40	2.76	22.90
GPC-PID	0.83	1.8	12.33
TLBO-PID	0.93	1.98	14.1
HGPTLBO-PID	0.33	1.05	0.033

The errors noted during this simulation are evaluated through the objective functions, and the corresponding results are presented in Table 7.2 and discussed below. The Integral of Absolute Error (IAE), Integral of Square of Error (ISE), and Integral of Time-Weighted Absolute Error (ITAE) are computed, revealing notable improvements in the objective functions when employing the proposed hybrid optimization process. Specifically, with the HGPCTLBO-PID controller, the ITAE decreases to 0.0003, the IAE decreases to 0.00211, and the ISE decreases to 0.00109, indicating significantly reduced error values. The objective function responses for all applied optimization algorithms are depicted in Figure 7.9.

Table 7.2: Objective function values for experiments conducted

Controller	ISE	IAE	ITAE
Classical PID	0.0219	0.03533	0.0699
GPC-PID	0.0033	0.0054	0.0066
TLBO-PID	0.0031	0.0049	0.0050
HGPTLBO-PID	0.00109	0.00211	0.0003

Moreover, the validity of the aforementioned arguments is demonstrated through simulation analyses of the ball balancer model using both the classical PID controller and the HGPCTLBO-PID controller with a continuous signal generator under unity feedback conditions. In this setup, the servo motor continuously receives voltage adjustments based on variations in the ball's output position on the rectangular plate. This analysis is illustrated in Figures 7.7 and 7.8.

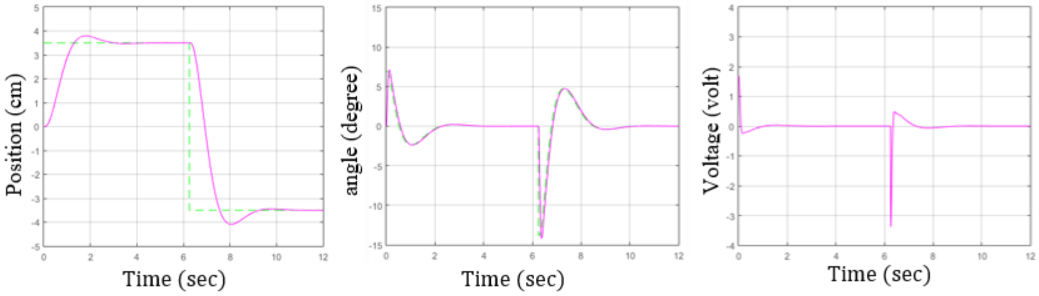


Fig. 7.7 The simulation outcomes of the ball balancer model using the traditional PID controller illustrate (from left to right): (a) the ball's tracking response on the square plate, (b) the variations in servo angle during operation, and (c) the feedback voltage supplied to the servo motor during closed-loop feedback operation

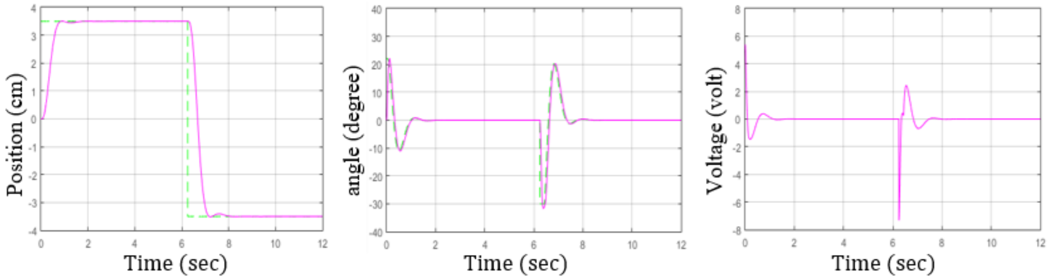


Fig. 7.8 The simulation outcomes of the ball balancer model using the HGPCTLBO-PID controller depict (from left to right): (a) the ball's tracking response on the square plate, (b) the variations in servo angle during operation, and (c) the feedback voltage supplied to the servo motor during closed-loop feedback operation

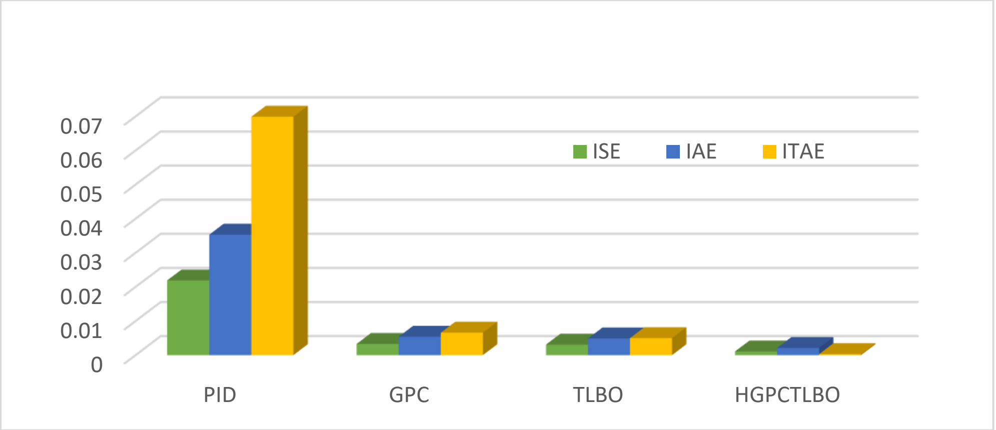


Fig. 7.9 Comparison of ISE, IAE & ITAE response of different objective functions while using classical PID, GPC-PID, TLBO-PID & HGPCTLBO-PID controllers

Figure 7.7 displays the simulation response obtained using the traditional PID controller, while Figure 7.8 exhibits the response achieved through the proposed optimization approach in the closed-loop feedback system of the balancer model. Upon comparing the initial images of both figures, it is evident that the tracking response of the ball position shows reduced oscillations, and the settling time is noticeably diminished. Furthermore, in the second and third images, employing the proposed algorithm results in smoother servo angle variations, accompanied by a reduced settling period. Additionally, the servo voltage, received via the feedback signal, exhibits a quicker response. These analyses serve to confirm that the time response of the ball balancer model is significantly enhanced by employing the proposed HGPCTLBO algorithm.

7.3.2 Real-time results with stability analysis

During the real-time operation, the control action is executed using the HGPCTLBO-PID controller. Initially, the traditional PID controller was assessed on the balancer system, followed by the application of the proposed method to optimize the parameters of this PID controller to achieve a reliable steady-state position. The D controller, along with the proposed algorithm, is then implemented on the hardware model, and the graphical results obtained in this process are documented and displayed in Figure 7.10 and Figure 7.11.

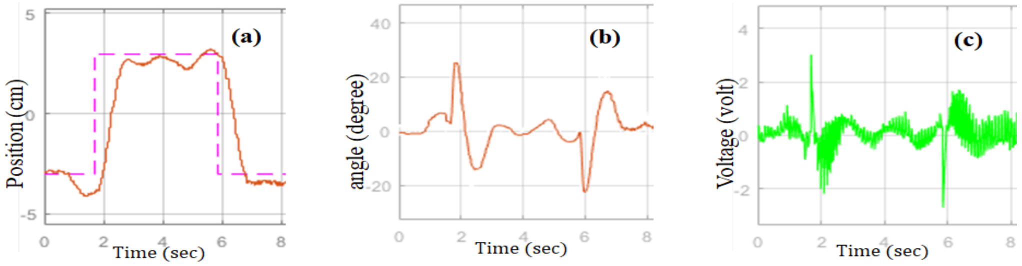


Fig. 7.10 Real time responses of (a) ball position (b) servo angle (c) voltage experienced by servo motor of ball balancer system using PID controller

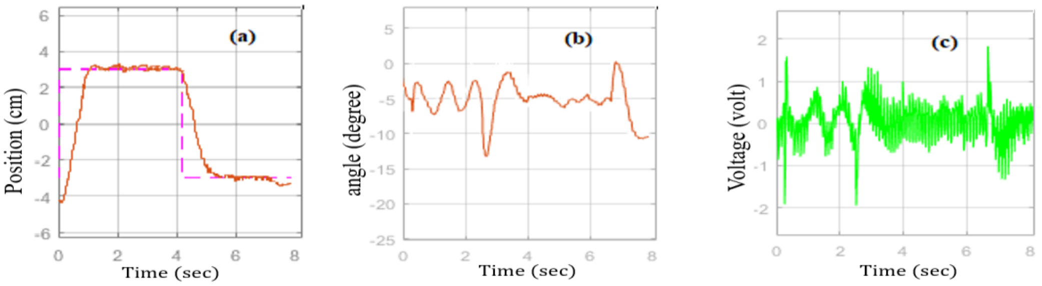


Fig. 7.11 Real time responses of (a) ball position (b) servo angle and (c) voltage experienced by servo motor of ball balancer model using HGPCTLBO –PID controller

The error is minimized while attaining the desired ball position trajectory on the hardware model. The graphs illustrate (a) variations in real-time ball trajectory, (b) observed servo angle variations during real-time operation, and (c) the voltage signal received by the servo motor during the operation of the ball balancer model using both the PID controller and the HGPCTLBO-optimized PID controller. After graphical analysis, it is evident that the ball position exhibits reduced oscillations in Figure 7.11(a) compared to Figure 7.10(a), indicating the enhanced reliability of the proposed controller. Furthermore, the observed servo angle variations reveal a narrower range in Figure 7.11(b) (-17 to 1) compared to Figure 7.10(b) (-25 to 25), indicating faster stability in the ball position over the square plate. Additionally, a comparison between

Figure 7.10(c) and Figure 7.11(c) shows less variation in the feedback signal received by the servo motor from (-1.8 to 1.5 volts) to (-2.2 to 2.2 volts) when using the proposed HGPCTLBO optimization technique. This clearly indicates reduced distortion in the real-time hardware model when employing the HGPCTLBO optimization technique, as evidenced by the improved graphical analysis.

7.4 HGPCTLBO-PID based control of 2DoF helicopter system

The MATLAB platform is employed to implement closed-loop control methods for both pitch and yaw axes through matrix equations. It's crucial to minimize deviations in system parameters to prevent significant variations in pitch and yaw angles, which are highly undesirable outcomes. To address this issue, a novel optimization approach is introduced. Additionally, real-time weather conditions are simulated using high-speed fans within the laboratory setting. This necessitates a more advanced controller capable of handling external disturbances, which is precisely what the proposed control scheme offers. This scheme utilizes an optimized controller, incorporating a fuzzy logic controller (FLC) that takes error and error deviation as inputs and generates three distinct outputs. These outputs serve as constraint parameters for a PID controller. Through the proposed hybridized optimization algorithm, these parameters are fine-tuned to minimize pitch and yaw errors. Figure 2 illustrates this control strategy, where error and its derivative are fed into the FLC, and the resulting outputs dictate the constraints for the PID controller, completing the fuzzy-PID implementation. Finally, the hybridized algorithm is applied to optimize these constraints.

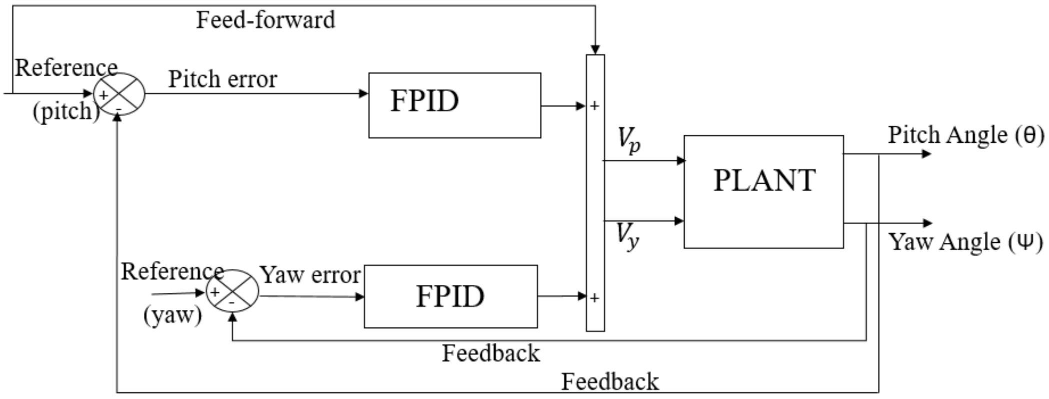


Fig. 7.12 Control mechanism used for handling the pitch and yaw angles of 2DoF helicopter system

FPID controller and the optimization mechanism for 2DoF helicopter:

Figure 7.13 illustrates the control strategy employed. The FPID controller is utilized for both the pitch and yaw axes. The variable "e" represents the pitch and yaw errors, while "e_d" signifies the derivative of these errors. These signals are quantized to form fuzzy subsets, followed by fuzzification to create linguistic values. A fuzzy rule base is then established to facilitate fuzzy inference. Subsequently, the signal is defuzzified into three output signals, which are given as input into the s-function block within MATLAB. This s-function block integrates the proposed hybridized HGPCTLBO algorithm. The process iterates until the pitch and yaw errors are minimized. Seven distinct sets, namely POSITIVE_BIG (PB), NEGATIVE_BIG (NB), POSITIVE_MEDIUM (PM), NEGATIVE_MEDIUM (NM), POSITIVE_LOW (PL), NEGATIVE_LOW (NL), and ZERO, are utilized to determine linguistic values for both input and output fuzzy subsets. In total, 49 fuzzy rules are applied using Gaussian membership functions with degrees ranging from 0 to 1, as discussed in previous chapter of this thesis, employing IF-THEN logic. This approach imbues the control action of the helicopter model with intelligent decision-making capabilities. The linguistic level control operates as an automatic control scheme, utilizing experimental data to implement the FLC rule base for Gaussian membership functions on both the pitch and yaw axes. The three outputs from this arrangement serve as constraints for the PID controller.

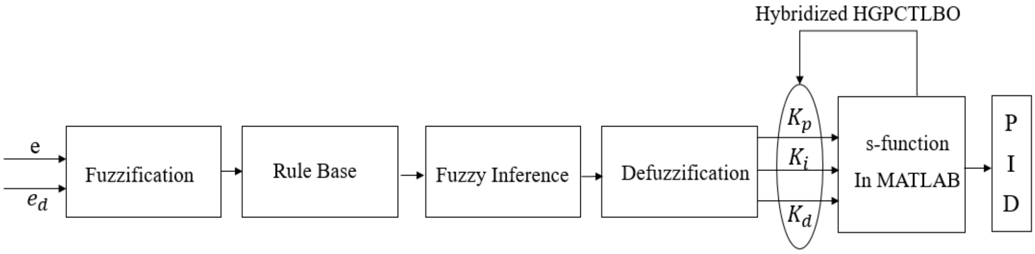


Fig. 7.13 Fuzzy-HGPCTLBO-PID controller mechanism used to handle flight angles of 2DoF helicopter

7.4.1 Numerical analysis and objective functions

The propellers exhibit cross-coupling characteristics, adding complexity to the tuning process. To address this challenge, fuzzy logic is employed, offering intelligent tuning to establish initial parameters for the PID controller. Once these initial values are determined, they are inputted into an optimization algorithm. The TLBO, GPC, and proposed HGPCTLBO algorithms are utilized to optimize the controller parameters. This sequential use of optimization algorithms, starting with TLBO and GPC before employing HGPCTLBO, allows for a comparative analysis of the tuning process. The pitch and yaw trajectories are then tracked using these three control algorithms with the objective of following as reference trajectory while ensuring stability in the operation of the helicopter model. Errors observed in pitch and yaw angles, as illustrated in Figure 7.12, serve as different objective functions to evaluate the performance of the control approaches. Which are mentioned as follows:

$$\text{ISE} = \int e^2(t)\, dt \tag{7.10}$$

$$\text{IAE} = \int |e(t)|\, dt \tag{7.11}$$

$$\text{ITSE} = \int t e^2(t)\, dt \tag{7.12}$$

The objective functions are utilized until the maximum number of iterations is reached. In the tuning process, three distinct algorithms—GPC, TLBO, and HGPCTLBO—are employed. The data collected post-implementation of the optimization algorithms is examined, and the three algorithms are compared through time response analysis. The pitch and yaw trajectories are monitored relative to the reference trajectories, and simulation results are generated. The parameters of the

FPID controllers are continually adjusted until the maximum number of iterations is reached, employing all three algorithms. Once this maximum iteration limit is reached, the final parameter values are utilized to compute the time response parameters. Graphs depicting the movement of pitch and yaw angles are plotted, as depicted in figures 7.14 to 7.19.

Figures 7.14 to 7.16 illustrate the yaw response observed on the simulation platform following the utilization of three different optimization algorithms. To facilitate comparison, Rise time (T_r), Peak time (T_p), Settling time (T_s) and Peak overshoot (M_p) are considered from the time domain. In Figure 7.14, the values for T_r, T_p & T_s are determined to be 0.5 sec, 0.95 sec, and 4.01 sec, respectively. These values are adjusted in Figure 7.15, where the time response parameters are measured at 0.45 sec, 0.81 sec, and 3.94 sec, respectively. Notably, in Figure 7.16, these parameters exhibit further enhancement. The final optimized parameters achieved through the proposed HGPCTLBO algorithm are found to be 0.41 sec, 0.58 sec, and 2.8 sec. These time response parameters, observed following analysis of the simulation results for the yaw axis of the helicopter model, are summarized in Table 7.3.

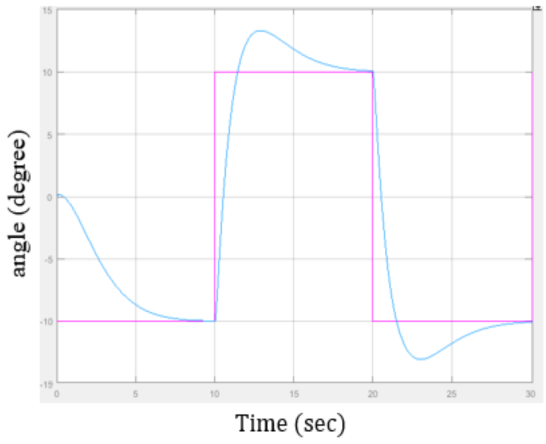


Fig. 7.14 Yaw angle vs time response when using TLBO optimization algorithm

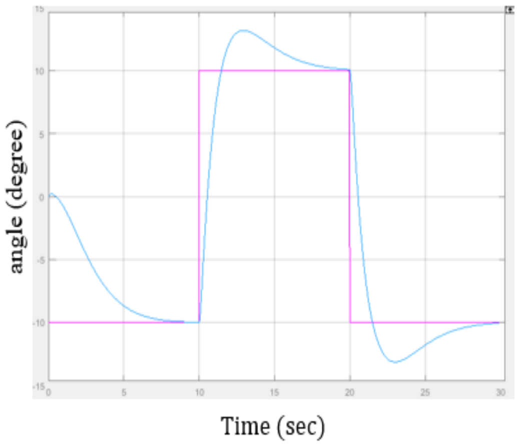


Fig. 7.15 Yaw angle vs time response when using GPC optimization algorithm

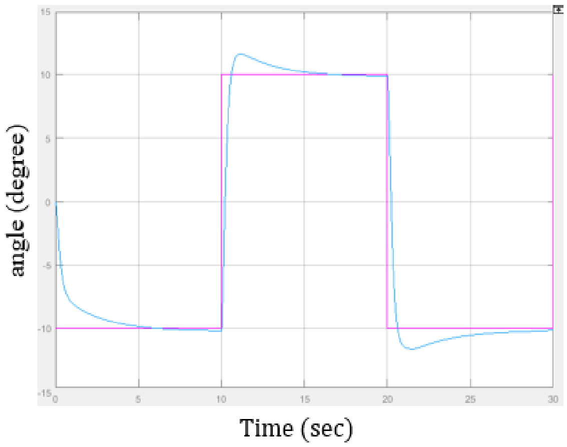


Fig. 7.16 Yaw angle vs time response when using HGPCTLBO optimization algorithm

Table 7.3: Response observed from yaw axis for various optimization methods

Method	Yaw Response			
	t_r (sec)	t_p (sec)	M_p (%)	t_s (sec)
TLBO	0.5	0.95	8.19	4.01
GPC	0.45	0.81	8.21	3.94
H-GPC-TLBO	0.41	0.58	4.6	2.8

Moreover, the simulation of the pitch axis is explored using the TLBO, GPC, and HGPCTLBO algorithms. These three optimization techniques are applied to optimize the fuzzy-PID controller, and the resultant outcomes are depicted in Figures 7.17 to 7.19.

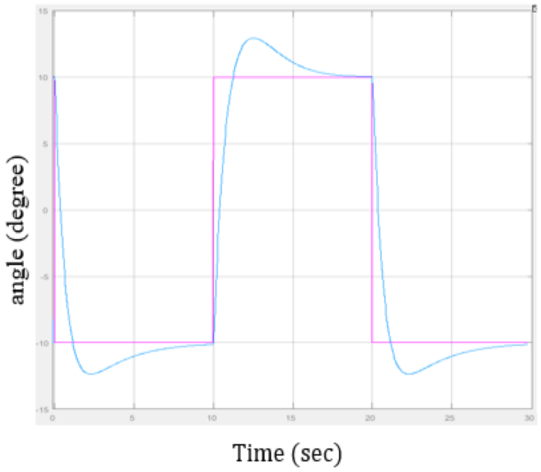


Fig. 7.17 Pitch angle vs time response when using TLBO optimization algorithm

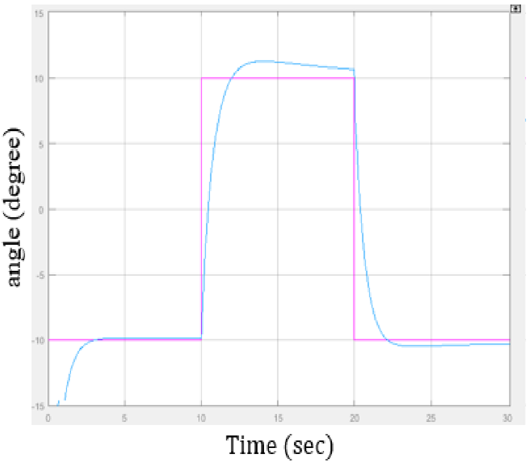


Fig. 7.18 Pitch angle vs time response when using GPC optimization algorithm

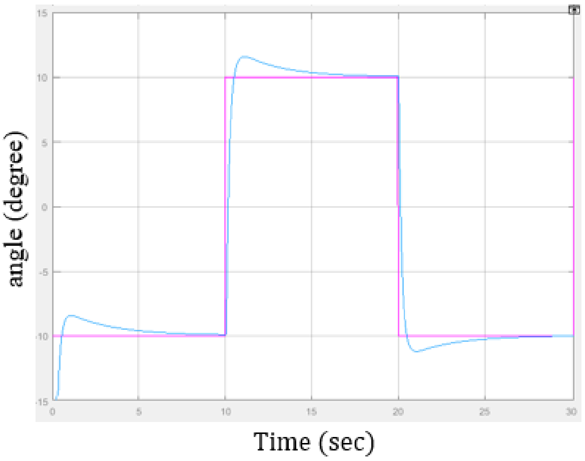


Fig. 7.19 Pitch angle vs time response when using HGPCTLBO optimization algorithm

In Figure 7.17, the pitch response following the implementation of the TLBO algorithm is illustrated. Here, the values for T_r, T_p & T_s are determined to be 0.66 sec, 1.1 sec, and 3.99 sec, respectively. Figure 7.18 illustrates the pitch angle response obtained after employing the GPC optimization technique on the fuzzy-PID controller. In this case, the values for T_r, T_p & T_s are measured at 0.58 sec, 1.1 sec, and 11.11 sec, respectively. It is evident that the settling time after utilizing GPC is unacceptable, indicating instability. This issue of undesirable time response parameters is resolved through the HGPCTLBO algorithm, which yields values of 0.46 sec, 0.9 sec, and 2.7 sec for T_r, T_p & T_s respectively as shown in figure 7.19. A detailed comparison of the time response parameters calculated for the pitch axis is presented in Table 7.4.

Table 7.4: Response observed from pitch axis for various optimization methods

Method	Pitch Response			
	t_r (sec)	t_p (sec)	M_p (%)	t_s (sec)
TLBO	0.66	1.1	4.18	3.99
GPC	0.58	1.1	4.30	11.11
H-GPC-TLBO	0.46	0.9	3.33	2.7

The above discussed findings are sufficient to determine the superiority of the hybridized method over the individual algorithms. The rise time, peak time, peak overshoot, and settling time of both the pitch and yaw axes are effectively managed through the utilization of the HGPCTLBO algorithm. When employing the hybridized optimization algorithm, issues such as undesired oscillations and prolonged stability time are effectively addressed. Various derivatives of error serve as objective functions during this simulation analysis. Specifically, Integral Square Error (ISE), Integral Absolute Error (IAE), and Integral Time Square Error (ITSE) are the three error objectives employed to illustrate the reduction of errors during the implementation of optimization algorithms, as shown below in table 7.5.

Table 7.5: Minimized error values after applying various optimization algorithms

Controller	IAE	ISE	ITSE
GPC	0.7717	0.1046	1.616
TLBO	0.8396	0.11	1.714
H-GPC-TLBO	0.7236	0.09869	1.5

For the initial objective function regarding error, namely ISE, its value stands at 0.1046 and 0.11 when employing GPC and TLBO, respectively. However, this figure significantly diminishes to 0.09869 with the application of the hybridized algorithm. Similarly, for the second objective function of error, IAE, its value registers at 0.7717 and 0.8396 when optimized using GPC and TLBO, respectively. This value is further reduced to 0.7236 with the use of the hybridized algorithm. Finally, the third objective function, ITSE, also sees improvement to 1.5 when the hybridized algorithm is utilized, compared to its earlier values of 1.616 and 1.714 with GPC and TLBO optimization algorithms, respectively. All the values of different error functions obtained after implementing various optimization algorithms are presented in table 3. A comparative analysis of Integral Square Error (ISE), Integral Absolute Error (IAE) and Integral Time Square Error (ITSE) is depicted using bar charts in Figures 7.20, 7.21 and 7.22, respectively.

7.4.2 Real-time results with stability analysis

Real-time control is achieved by employing optimization algorithms to adjust the parameters of the FPID controller. These optimization algorithms include GPC, TLBO, and the hybridized GPC-TLBO method. The changes in pitch and yaw angles over time are depicted graphically. The graphical representations of the pitch trajectory and yaw trajectory are presented in Figures 7.23 and 7.24, respectively. Initially, the traditional FPID controller is tuned using the GPC algorithm, and the resulting pitch response is graphically recorded, as shown in Figure 7.23(a). It's observed that the trajectory exhibits oscillations and fails to reach the reference amplitude within the complete cycle, indicating an unreliable controller. Subsequently, the same FPID controller is tuned using the TLBO algorithm, resulting in even larger oscillations and an unstable system response, as depicted in Figure 7.23(b). In contrast, Figure 7.23(c) illustrates the response obtained using the H-GPC-TLBO algorithm on the same FPID controller.

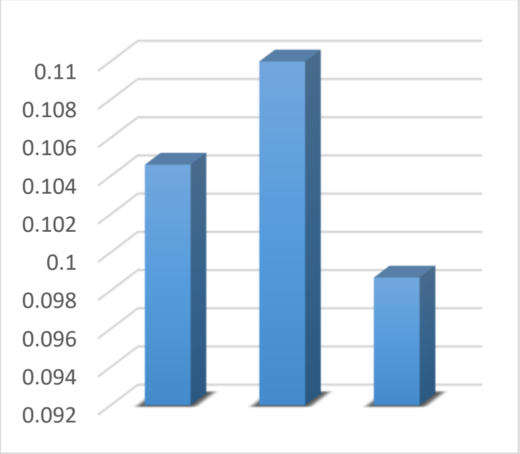


Fig. 7.20 Integral of Square error obtained after using GPC, TLBO & HGPCTLBO algorithms

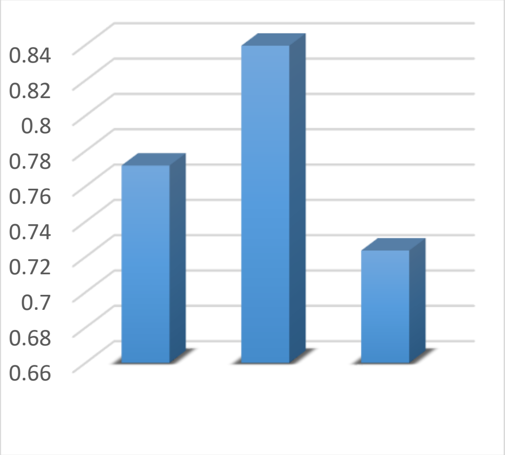


Fig. 7.21 Integral of absolute error obtained after using GPC, TLBO & HGPCTLBO algorithms

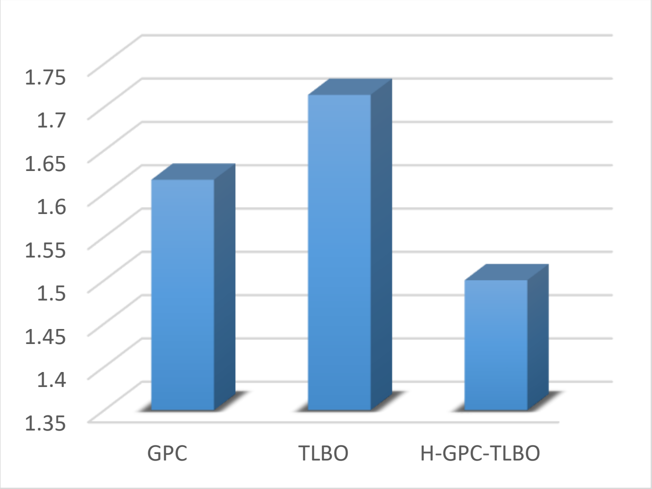
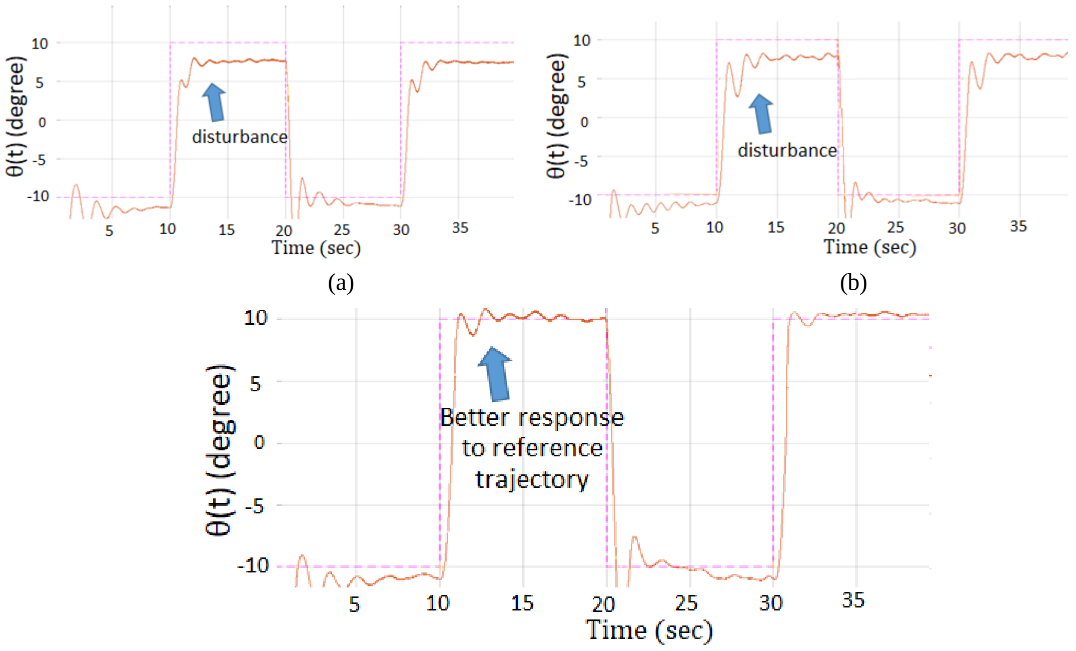


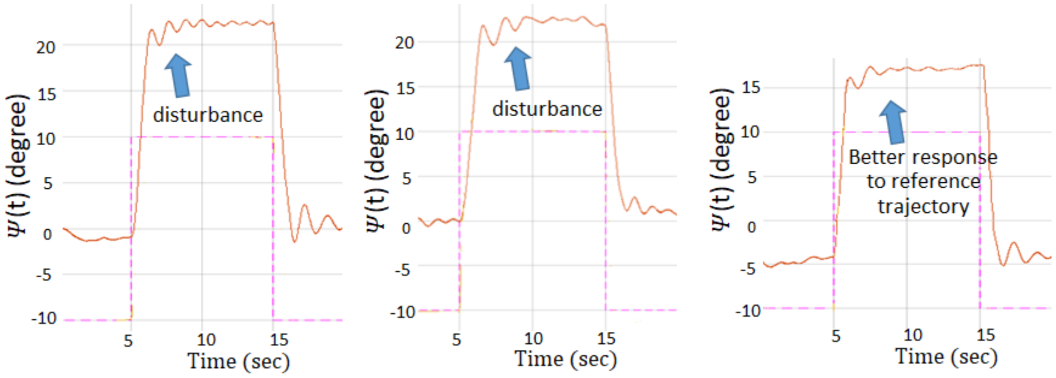
Fig. 7.22 Integral of time square error obtained after using GPC, TLBO & HGPCTLBO algorithms



(a) (b)

(c)

Fig. 7.23 The evaluation of the pitch axis trajectory of a real-time helicopter model under the influence of external disturbances caused by high-speed fans subsequent to the optimization algorithms including (a) GPC (b) TLBO and (c) the hybridized-GPC-TLBO



(a) (b) (c)

Fig. 7.24 The evaluation of the yaw axis trajectory of a real-time helicopter model under the influence of external disturbances caused by high-speed fans subsequent to the optimization algorithms including (a) GPC (b) TLBO and (c) the hybridized-GPC-TLBO

The pitch angle trajectory closely follows the signal reference amplitude with fewer oscillations, settling down by the end of the signal cycle. This demonstrates the reliability and efficiency of the H-GPC-TLBO algorithm in tracking the pitch reference signal, even in the presence of external disturbances, minimizing errors. Similarly, these three algorithms are applied to the yaw error signal. Initially, a GPC-tuned FPID controller is utilized to control the yaw trajectory of the helicopter model, resulting in a trajectory with a significant offset and oscillations, as shown in Figure 7.24(a). The same FPID controller is then tuned using the TLBO algorithm, yielding a yaw angle trajectory with similar large offset and oscillations, as depicted in Figure

7.24(b). Finally, the hybridized GPC-TLBO algorithm is employed to tune the FPID controller parameters, resulting in a less oscillatory response that settles by the end of the reference signal cycle, as shown in Figure 7.24(c). This demonstrates the distortion-less behaviour of the helicopter model achieved through the proposed optimization technique. The improved graphical results presented in Figures 7.23 and 7.24 suffice to demonstrate the superiority of the proposed algorithm in mitigating external disturbances on the helicopter model.

7.5 Conclusion

The hybridized algorithm is employed to manage external disturbances in the 2DoF helicopter system through the integration of a fuzzy-proportional-integral-derivative (FPID) controller and the same algorithm is employed to manage the servo angle variations of a 2DoF ball balancer system under unknown but bounded disturbance using proportional-integral-derivative controller. The hybridized-GPC-TLBO algorithm yields a more stable and reliable response, even when faced with external disturbances such as unfavourable wind speeds caused by high-speed fans in helicopter experiment and nonlinearity created by dead zone in ball balancer experiment. This hybrid optimization technique has demonstrated remarkable efficiency and superiority compared to individual algorithm results. Both simulation and real-time tuning are conducted on benchmark 2DoF helicopter model and 2DoF ball balancer system, with extensive numerical and graphical comparisons performed to showcase the performance of the developed control technique. External disturbances are diminished by minimizing error signals experienced by the pitch and yaw axes of the 2DoF helicopter model. The position error is diminished by minimizing the error signal experienced by servo axis of 2DoF ball balancer model. To achieve these error reductions, Integral Square Error (ISE), Integral Absolute Error (IAE) and Integral Time Square Error (ITSE) are employed as objective functions. The MATLAB/Simulink platform is utilized to demonstrate the effective control efficacy and adaptability of the developed control scheme in ensuring stability for both the 2DoF systems.

Chapter 8. Conclusion and Future Scope of Work

8.1 Introduction

This chapter offers concluding reflections on different control strategies explored in the thesis, drawing from observed simulation and real-time outcomes. In this thesis, a new metaheuristic algorithms i.e. iTLBO & HGPCTLBO are introduced and used to optimize the classical and intelligent controllers on different benchmark nonlinear control problems to showcase their efficacy and performance. Initially, challenges within mechanical systems were pinpointed through modelling benchmark systems like an aerial vehicle and robotic balancer. Subsequently, these challenges were addressed using diverse nonlinear control theory approaches. The primary goal of developing controllers was to enable path tracking and position control for the benchmark systems, mitigating issues arising from uncertainty and outside laboratory environment. This chapter delivers a conclusion along with an exploration of future prospects in this domain.

8.2 Contributions of the work

The summary of work presented in this thesis is briefed chapter-wise as follows:

Chapter 1 offers an introduction to nonlinear systems control, covering fundamental principles of nonlinear control theory. It outlines the motivations driving advancements in this field and formulates the necessary objectives. It mentions that several optimization techniques are used to handle nonlinear benchmark problems. These versatile optimization techniques have found applications across diverse fields of science, technology and engineering, facilitating the selection of optimal solutions from a variety of possible alternatives. Additionally, it emphasizes the significant contributions of the thesis and provides an overview of the subsequent chapters.

Chapter 2 provides a literature survey on a broad spectrum of topics in nonlinear control theory, ranging from contemporary optimization algorithms to the fusion of multiple classical and intelligent control techniques. It includes surveys, comparisons, and analyses of performance. The review reveals that while linear controllers offer a straightforward means of designing closed-loop control systems for these benchmarks, the intricate nonlinear dynamics of the systems pose challenges in providing viable solutions and limit the general applicability of control laws. Linearization of nonlinear systems is attempted but adversely affects system response speed, prompting the development of several nonlinear control techniques to address underactuated system issues. Furthermore, it explores variations of control techniques and their applications in 2DoF systems. In light of drawbacks and limitations caught up in literature, basic control schemes integrating fuzzy logic are developed for the helicopter and ball balancer systems. Evaluation results shed light on the performance of these control techniques and underscore their vulnerabilities to external disturbances and parametric uncertainties. These problems are handled using various classical and intelligent control techniques, as discussed in different chapters of this thesis.

Chapter 3 provides background information on the 2-degree-of-freedom (2DoF) helicopter and ball balancer benchmark systems, outlining fundamental concepts in their mechanics. The complete mathematical model is discussed in detail in this chapter. The control equations for these benchmark systems are obtained, which are further used in upcoming chapters of this thesis. The benchmark helicopter model is introduced with external disturbance using two high speed fans, so that it will experience turbulence like situations. The ball balancer system is discussed with dead zone linearity with bounded but unknown disturbances. These systems are further controlled using various control schemes in chapter 4 to chapter 7.

Chapter 4 presents optimized classical control techniques, which are applied on two nonlinear systems i.e. 2DoF ball balancer system and 2DoF helicopter system, employing TLBO and GPC as the optimization approaches. Different cost functions are established in this chapter and the same are employed to reduce the system errors observed during the simulation of 2DoF benchmark systems. Time response evaluations are performed on both systems, representing the inaugural utilization of these algorithms on such nonlinear systems. The stable responses attained highlight the originality of this work. All the results are validated through graphical as well as numerical results in detail.

Chapter 5 presents the improvement in teaching learning based optimization algorithm, namely iTLBO, through mathematical and conceptual evidence. A data-driven solution is provided to update the teaching factor in the execution of TLBO algorithm. The provided solution showcase the superiority of iTLBO over TLBO algorithm. Various validity criteria, including traditional benchmark functions, CEC functions and standard error derivatives of objective function values are obtained across multiple runs. These criteria examined the effectiveness of iTLBO algorithm. Simulation and real time results indicate that the iTLBO algorithm exhibits faster convergence compared to other optimization techniques in literature on linear and nonlinear benchmark functions. When applied to CEC test functions, iTLBO outperforms other algorithms, effectively balancing diversification and intensification to locate global optima. Finally iTLBO is used to constraints of PID controller, which is implemented on benchmark nonlinear systems i.e. 2DoF ball balancer system and 2DoF helicopter system. It was observed through graphical as well as numerical results that the iTLBO-PID control method is providing stable results to both the nonlinear systems. Algorithm development and implementation is discussed at length in this chapter.

Chapter 6 outlines the hybridization of intelligent Fuzzy Logic controller with classical PID controller to optimize the angle variation and axis variations of benchmark systems. The hybrid FPID controller is optimized using the iTLBO algorithm, which is developed in chapter 5 of this thesis. The optimized FPID control scheme for coupled helicopter system and decoupled ball balancer system is discussed at length in this chapter. The iTLBO algorithm showcases the improved time response analysis on simulation model of these benchmark systems. The real time graphs of both the benchmark systems are enough to prove the improved stability and balancing under the unfavourable condition also. Results obtained using this control scheme are compared with results available using earlier control methods and the dominance of iTLBO-FPID controller is proved. In the results, the position error, angle error and

ITAE error are reduced while using the proposed hybridized intelligent-classical control scheme.

Chapter 7 presents mathematical and conceptual description of a developed hybrid optimization algorithm, namely HGPCTLBO. The optimization algorithm introduces “command phase” and “Co-operate phase” to the GPC algorithm. The Co-operate phase is inspired from the learner phase of TLBO algorithm. From this, the hybridization of GPC and TLBO is obtained and named as hybrid Giza pyramid construction teaching learning based optimization (HGPCTLBO) algorithm. The algorithm flow of HGPCTLBO is discussed in detail. This algorithm is used to optimize the classical PID controller as well as the intelligent Fuzzy-PID controller. The classical HGPCTLBO-PID control mechanism is developed for 2DoF ball balancer system. Here, the servo angle variations of square plate of ball balancer is controlled and the ball position over this plate is maintained successfully. The intelligent HGPCTLBO-Fuzzy-PID control mechanism is employed on 2DoF helicopter system. Here, the pitch angle axis and the yaw angle axis are controlled using various controllers while maintaining the minimal error using square reference trajectory.

HGPCTLBO proved to be efficient and faster when the time response analysis is done for both the 2DoF nonlinear systems. The real – time graphical results are also showing the reduced fluctuations and improved error for position control as well as balancing of nonlinear systems. The IAE, ITAE and ISE cost functions are showing minimized error when HGPCTLBO optimization algorithm is implemented on PID controller as well as FPID controller of benchmark systems.

8.3 Suggestion for future scope of work

This section outlines potential avenues for future research building upon the current study. For trajectory tracking employing intelligent and classical controllers, enhancements can focus on improving tracking efficiency while maintaining steady-state performance and accommodating multiple degrees of freedom. Exploring the operation of underactuated systems in both strong and weak fields, which involve zero dynamics, presents another promising direction. To enhance its effectiveness, the HGPCTLBO & iTLBO method could be integrated with additional optimization algorithms. Ultimately, an assessment of HGPCTLBO & iTLBO efficacy across diverse real-world engineering optimization assignments may be conducted.

Additionally, incorporating migration strategy-based algorithms such as biogeography-based optimization (BBO), biologically inspired mechanisms like bacteria foraging algorithm (BFA), population-based evolutionary methods like invasive weed optimization-based algorithm (IWO) and other metaheuristic approaches like Jaya algorithm, Honey Badger Algorithm, the Dingo Optimizer and Artificial Gorilla Troops Optimizer with nonlinear systems may provide further research in this field. The methods like SVM, deep learning and block chain algorithms is also available to explore deep in this area. Furthermore, conducting reliability analyses of components within mechanical systems and investigating the impact of various control approaches could serve as valuable extensions.

8.4 Societal impact of the work

Enhanced Disaster Response and Humanitarian Efforts: Autonomous systems with strong control mechanisms can be vital in disaster response. Drones and robots with intelligent controllers can be sent to areas affected by earthquakes, collapsed buildings or floods to search for survivors, assess damage and deliver critical supplies. Their ability to navigate challenging terrain with little human oversight allows for quicker responses, improving the chances of saving lives and reducing the risks faced by emergency responders in dangerous situations.

Advancement in Autonomous Systems and Robotics: This research improves the accuracy, stability, and flexibility of autonomous robots, helping them operate better in changing and unpredictable environments. With better path tracking and balance control, robots and drones can efficiently handle tasks like warehouse management, security monitoring, and industrial inspections. Their ability to move independently without errors reduces the need for human involvement, cutting risks and costs while making the system more reliable.

Safer and More Efficient Transportation: The intelligent control methods in this research can be used in smart transportation to improve the safety and efficiency of self-driving cars, drones, and automated trains. Better path tracking allows these vehicles to adjust in real time, reducing accidents and improving route planning. This can help ease traffic and decrease fuel consumption, supporting a more sustainable and eco-friendly transportation system.

Breakthroughs in Aerospace and UAV Technology: This research offers improved control methods for aerial vehicles, helping them manage turbulence and external disturbances, which is essential for UAVs and future air mobility. For example, autonomous drones with advanced balance and navigation can be used in remote areas for delivering medical supplies, emergency surveillance or search-and-rescue operations.

Improved Industrial Automation and Smart Manufacturing: With intelligent and optimized control systems, industrial automation can be more precise, efficient and adaptable. Manufacturing robots with advanced algorithms can perform complex tasks like assembling delicate parts, conducting quality checks and adjusting to unexpected changes in real time. This can boost production efficiency, reduce operational costs and improve workplace safety by minimizing human exposure to dangerous environments.

Appendix #1

#All steps involved in the optimization process of real time automatic tuning of PID gain values through TLBO algorithm are explained herewith:

The constrained parameters of the algorithm are defined along with the variable sizes and boundaries: -

```
nVar = 3; % Number of Unknown Variables which are namely P, I and D

VarSize = [1 nVar]; % Unknown Variables Matrix Size

VarMin = 0; % Unknown Variables Lower Bound

VarMax = 10; % Unknown Variables Upper Bound
```

Then cost function is defined through a script named error:

```
CostFunction = @error: %the error script is called from its function which is defined herein
```

Then the parameters of TLBO algorithm are defined and initialized as follows: -

```
MaxIt = 1000; % Maximum Number of Iterations

nPop = 50; % Population Size

% Empty Structure for Individuals

empty_individual.Position = [];

empty_individual.Cost = [];

% Initialize Population Array

pop = repmat(empty_individual, nPop, 1);

% Initialize Best Solution

BestSol.Cost = inf;

% Initialize Population Members

for i=1:nPop

pop(i).Position = unifrnd(VarMin, VarMax, VarSize);

pop(i).Cost = CostFunction(pop(i).Position);

if pop(i).Cost < BestSol.Cost

BestSol = pop(i);

end end
```

Then the cost function for optimal PID is defined alongside the ISE, IAE, ITSE and ITAE.

```
function [C,fval]=optimPID(G,ctype,idx)
```

```
% Inputs: G: Model of 2DoF System, ctype: Controler type (1 = P, 2* = PI, 3 = PID)
% idx: Performance criterion, 1 – ISE 2 – IAE 3 - ITSE 4 - ITAE
% Outputs: C: PID transfer function, fval: optimal paramters
C1=optimPID(G,3,1); % PID-Control, ISE index
C2=optimPID(G,3,2); % PID-Control, IAE index
C3=optimPID(G,3,3); % PID-Control, ITSE index
C4=optimPID(G,3,4); % PID-Control, ITAE index
K=znpidtuning(G,3); % Ziegler-Nichols stability margin tuning
t=0:0.1:30;
y1=step(feedback(C1*G,1),t);
y2=step(feedback(C2*G,1),t);
y3=step(feedback(C3*G,1),t);
y4=step(feedback(C4*G,1),t);
y=step(feedback(G*(K.kc*(1+tf(1,[K.ti 0])+tf([K.td 0],1))),1),t);
[Gm,Pm,Wcg]=margin(G); % Initial parameters using stability based tuning
pu=2*pi/Wcg;
ku=Gm;
x=ku/2;
den=1;
if ctype==2
x=ku/2.2*[1 1.2/pu];
den=[1 0];
elseif ctype==3
x=ku*2/pu/1.7*[pu/8 1 2/pu];
den=[1 0]; end
```

Then the response for initial tuning is done to trace the initial parameters: -

```
[y,t]=step(feedback(tf(x,den)*G,1));
cost = @error(x,G,den,t,dt,idx); % redefine cost function to facilitate optimization
function J=error(x,G,den,t,dt,idx)
```

```
e=1-step(feedback(G*tf(x,den),1),t); % control error of step response
```

Then the performance calculation for ISE, IAE, ITSE, ITAE are evaluated as following: -

```
switch idx
case 1 % ISE J=e'*e*dt;
case 2 % IAE J=sum(abs(e)*dt);
case 3 % ITSE J=(t.*e'*dt)*e;
case 4 % ITAE J=sum(t'.*abs(e)*dt);
end
```

Now Initialize the Best result for the current iteration:

```
BestCosts = zeros(MaxIt,1);
```

Now start the TLBO algorithm Main Loop

```
for it=1:MaxIt
```

Calculation of Population Mean is done as follows

```
Mean = 0;
for i=1:nPop
Mean = Mean + pop(i).Position;
end
Mean = Mean/nPop;
```

The best solution is selected as Teacher as follows: -

```
Teacher = pop(1);
for i=2:nPop
if pop(i).Cost < Teacher.Cost
Teacher = pop(i);
end
end
```

Teacher Phase main loop starts as: -

```
for i=1:nPop
```

Create Empty Solution

```
newsol = empty_individual;
```

Teaching phase is moving towards teacher as: -

```
newsol.Position = pop(i).Position + rand(VarSize).*(Teacher.Position - TF*Mean);
```

Evaluation of new position for cost function: -

```
newsol.Cost = CostFunction(newsol.Position);
```

Comparison for the required greedy solution

```
if newsol.Cost<pop(i).Cost
pop(i) = newsol;
if pop(i).Cost < BestSol.Cost
BestSol = pop(i);
end end end
```

```
for i=1:nPop
A = 1:nPop;
A(i)=[];
j = A(randi(nPop-1));
Step = pop(i).Position - pop(j).Position;
if pop(j).Cost < pop(i).Cost
Step = -Step; end
```

The above-mentioned equations will be repeated until the maximum iterations are reached.

```
newsol.Position = pop(i).Position + rand(VarSize).*Step;
newsol.Cost = CostFunction(newsol.Position);
if newsol.Cost<pop(i).Cost
pop(i) = newsol;
if pop(i).Cost < BestSol.Cost
BestSol = pop(i);
end end end
```

Appendix #2

#All steps involved in the optimization process of real time automatic tuning of PID gain values through GPC algorithm are explained herewith:

The parameters of the algorithm are defined along with the variable sizes and boundaries: -

```
nVar=3; % Number of Decision Variables i.e. P, I and D.
VarSize=[1 nVar]; % Decision Variables Matrix Size
VarMin=0; % Decision Variables Lower Bound
VarMax=10; % Decision Variables Upper Bound
```

Algorithm specific Giza Pyramids Construction (GPC) Parameters

```
MaxIteration=1000; % Maximum Number of Iterations (Days of work)
nPop=50; % Number of workers
G = 9.8; % Gravity
Tetha = 14; % Angle of Ramp
MuMin = 1; % Minimum Friction
MuMax = 10; % Maximum Friction
pSS= 0.5; % Substitution Probability
```

Then cost function is defined through a script named error:

```
CostFunction = @error: %the error script is called from its function
```

Then the cost function for optimal PID is defined alongside the ISE, IAE, ITSE and ITAE.

```
function [C,fval]=optimPID(G,ctype,idx)
C1=optimPID(G,3,1); % PID-Control, ISE index
C2=optimPID(G,3,2); % PID-Control, IAE index
C3=optimPID(G,3,3); % PID-Control, ITSE index
C4=optimPID(G,3,4); % PID-Control, ITAE index
K=znpidtuning(G,3); % Ziegler-Nichols stability margin tuning
t=0:0.1:30;
y1=step(feedback(C1*G,1),t);
y2=step(feedback(C2*G,1),t);
y3=step(feedback(C3*G,1),t);
```

```
y4=step(feedback(C4*G,1),t);
y=step(feedback(G*(K.kc*(1+tf(1,[K.ti 0])+tf([K.td 0],1))),1),t);
[Gm,Pm,Wcg]=margin(G); % Initial parameters using stability based tuning
pu=2*pi/Wcg;
ku=Gm;
x=ku/2;
den=1;
if ctype==2
x=ku/2.2*[1 1.2/pu];
den=[1 0];
elseif ctype==3
x=ku*2/pu/1.7*[pu/8 1 2/pu];
den=[1 0];
end
```

Then the response for initial tuning is done to trace the initial parameters: -

```
[y,t]=step(feedback(tf(x,den)*G,1));
cost = @error(x,G,den,t,dt,idx); % redefine cost function to facilitate optimization
opt=optimset('display','off','TolX',1e-9,'TolFun',1e-9,'LargeScale','off');
flag=0; while ~flag % if flag=0 restart optimization from current solution
[x,fval,flag]=fminunc(cost,x,opt);
end
function J=error(x,G,den,t,dt,idx)
e=1-step(feedback(G*tf(x,den),1),t); % control error of step response
```

Then the performance calculation for ISE, IAE, ITSE, ITAE are evaluated as following: -

```
switch idx
case 1 % ISE J=e'*e*dt;
case 2 % IAE J=sum(abs(e)*dt);
case 3 % ITSE J=(t.*e'*dt)*e;
case 4 % ITAE J=sum(t'.*abs(e)*dt);  end
```

Initialization of the algorithm through individual steps one after one as: -

```
stone.Position=[];
stone.Cost=[];
pop=repmat(stone,nPop,1);
best_worker.Cost=inf;
for i=1:nPop
pop(i).Position=unifrnd(VarMin,VarMax,VarSize);
pop(i).Cost=CostFunction(pop(i).Position);
if pop(i).Cost<=best_worker.Cost
best_worker=pop(i); % Pharaoh's special agent is designated here
end end
```

Giza Pyramids Construction (GPC) Algorithm Main Loop

```
for it=1:MaxIteration
newpop=repmat(stone,nPop,1);
for i=1:nPop
newpop(i).Cost = inf;
V0= rand(0,1); % Initial Velocity
Mu= MuMin+(MuMax-MuMin)*rand(1,10); % Friction
d = (V0^2)/((2*G)*(sind(Tetha)+(Mu*cosd(Tetha)))); % Stone Destination
x = (V0^2)/((2*G)*(sind(Tetha))); % Worker Movement
epsilon=unifrnd(-((VarMax-VarMin)/2),((VarMax-VarMin)/2),VarSize); % Epsilon
newsol.Position = (pop(i).Position+d).*(x*epsilon); % Position of Stone and Worker
newsol.Position=max(newsol.Position,VarMin);
newsol.Position=min(newsol.Position,VarMax);
```

Substitution steps followed in each iteration as: -

```
z=zeros(size(pop(i).Position));
k0=randi([1 numel(pop(i).Position)]);
for k=1:numel(pop(i).Position)
if k==k0 || rand<=pSS
```

```
z(k)=newsol.Position(k);
else z(k)=pop(i).Position(k);
end end
newsol.Position=z;
newsol.Cost=CostFunction(newsol.Position);
if newsol.Cost <= newpop(i).Cost
newpop(i) = newsol;
if newpop(i).Cost<=best_worker.Cost
best_worker=newpop(i);
end end end
```

Best cost value at the end of all iteration is stored as: -

```
        BestCost(it)=pop(1).Cost;
end
```

REFERENCES

1. O. Nelles, “Nonlinear System Identification” Berlin, Springer Berlin Heidelberg, 2001.
2. K. Hussain, M. N. Mohd Salleh, S. Cheng and Y. Shi, “Metaheuristic research: a comprehensive survey,” Artif. Intell. Rev., vol. 52, no. 4, pp. 2191–2233, 2019. doi: 10.1007/s10462-017-9605-z
3. W. G. Wolpert, D. H. Macready, “No free lunch theorems for optimization,” IEEE Trans. Evol. Comput., vol. 1, pp. 67–82, 1997.
4. S. Desale, A. Rasool, S. Andhale and P. Rane, “Heuristic and Meta-Heuristic Algorithms and Their Relevance to the Real World: A Survey,” Int. J. Comput. Eng. Res. Trends, vol. 351, no. 5, pp. 2349–7084, 2015.
5. X. S. Yang, “Engineering Optimization: An Introduction with Metaheuristic Applications,” John Wiley and Sons, Hoboken, NJ, USA. 2010.
6. A. H. Gandomi, X. S. Yang, S. Talatahari and A. H. Alavi, “Metaheuristic Algorithms in Modeling and Optimization,” no. 1, pp. 1 – 24, December. 2013.
7. W. G. Wolpert, D.H., Macready, “No Free Lunch Theorems for Optimization,” IEEE Trans. Evol. Comput., vol. 1, no. 1, pp. 67–82, 1997.
8. A. Megretski and A. Rantzer, “System analysis via integral quadratic constraints,” IEEE Trans. Automat. Contr., vol. 42, no. 6, pp. 819–830, Jun. 1997, doi: 10.1109/9.587335.
9. F. Albertini and E. D. Sontag, “State observability in recurrent neural networks,” Syst. Control Lett., vol. 22, no. 4, pp. 235–244, Apr. 1994, doi: 10.1016/0167-6911(94)90054-X.
10. N. Shimkin, “Nonlinear Control Systems,” in Encyclopedia of Neuroscience, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 2886–2889.
11. E. D. Sontag, “Smooth stabilization implies coprime factorization,” IEEE Trans. Automat. Contr., vol. 34, no. 4, pp. 435–443, Apr. 1989, doi: 10.1109/9.28018.
12. M. A. Ajwad et. al. “A systematic review of current and emergent manipulator control approaches.,” Front Mech Eng China, vol. 10, pp. 198–210, 2015.
13. T. Kara and I. Eker, “Nonlinear modeling and identification of a DC motor for bidirectional operation with real time experiments,” Energy Convers. Manag., vol. 45, no. 7–8, pp. 1087–1106, 2004, doi: 10.1016/j.enconman.2003.08.005.
14. A. C. Menini et al, “Current trends in nonlinear systems and control” - In Honor of Petar Kokotovic and Turi Nicosia: Springer. 2006.
15. M. W. Spong and L. Praly, “Control of underactuated mechanical systems using switching and saturation,” in Control Using Logic-Based Switching, London: SpringerVerlag, pp. 162–172, April 2006.
16. M. W. Spong, “Energy Based Control of a Class of Underactuated Mechanical Systems,” IFAC Proc. Vol., vol. 29, no. 1, pp. 2828–2832, Jun. 1996, doi: 10.1016/S1474-6670(17)58105-7.
17. S. A. Bortoff and M. W. Spong, “Pseudolinearization of the acrobot using spline functions,” in [1992] Proceedings of the 31st IEEE Conference on Decision and Control, pp. 593–598, doi: 10.1109/CDC.1992.371658.
18. J. Hauser, S. Sastry and P. Kokotovic, “Nonlinear control via approximate input-output linearization: the ball and beam example,” IEEE Trans. Automat. Contr., vol. 37, no. 3, pp. 392–398, Mar. 1992, doi: 10.1109/9.119645.
19. A. Teel and L. Praly “Tools for Semiglobal Stabilization by Partial State and Output Feedback,” SIAM J. Control Optim., vol. 33, no. 5, pp. 1443–1488, Sep. 1995, doi: 10.1137/S0363012992241430.

20. J. Hauser, S. Sastry and G. Meyer, "Nonlinear control design for slightly non-minimum phase systems: Application to V/STOL aircraft," Automatica, vol. 28, no. 4, pp. 665–679, Jul. 1992, doi: 10.1016/0005-1098(92)90029-F.
21. N. H. McClamroch and I. Kolmanovsky, "A hybrid switched mode control approach for V/STOL flight control problems," in Proceedings of 35th IEEE Conference on Decision and Control, vol. 3, pp. 2648–2653, doi: 10.1109/CDC.1996.573503.
22. O. Egeland, M. Dalsmo and O. J. Sørdalen, "Feedback Control of a Nonholonomic Underwater Vehicle With a Constant Desired Configuration," Int. J. Rob. Res., vol. 15, no. 1, pp. 24–35, Feb. 1996, doi: 10.1177/027836499601500102.
23. K. Y. Pettersen and O. Egeland, "Exponential stabilization of an underactuated surface vessel," in Proceedings of 35th IEEE Conference on Decision and Control, vol. 1, pp. 67–972, doi: 10.1109/CDC.1996.574602.
24. P. Morin and C. Samson, "Time-varying exponential stabilization of the attitude of a rigid spacecraft with two controls," in Proceedings of 1995 34th IEEE Conference on Decision and Control, vol. 4, pp. 3988–3993, doi: 10.1109/CDC.1995.479228.
25. K. Y. Wichlund, O. J. Sordalen and O. Egeland, "Control of vehicles with second-order nonholonomic constraints: Underactuated vehicles," in 3rd European Control Conference, 1995, pp. 3086–3091.
26. R. W. Brockett, "Asymptotic stability and feedback stabilization," Differ. Geom. Control Theory, pp. 181–191, 1983.
27. M. Reyhanoglu, A. van der Schaft, N. H. Mcclamroch and I. Kolmanovsky, "Dynamics and control of a class of underactuated mechanical systems," IEEE Trans. Automat. Contr., vol. 44, no. 9, pp. 1663–1671, 1999, doi: 10.1109/9.788533.
28. A. M. Bloch, M. Reyhanoglu and N. H. McClamroch, "Control and stabilization of nonholonomic dynamic systems," IEEE Trans. Automat. Contr., vol. 37, no. 11, pp. 1746–1757, 1992, doi: 10.1109/9.173144.
29. H. J. Sussmann, "A General Theorem on Local Controllability," SIAM J. Control Optim., vol. 25, no. 1, pp. 158–194, Jan. 1987, doi: 10.1137/0325011.
30. M. Reyhanoglu, Sangbum Cho, N. H. McClamroch and I. Kolmanovsky, "Discontinuous feedback control of a planar rigid body with an unactuated degree of freedom," in Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No.98CH36171), vol. 1, pp. 433–438, doi: 10.1109/CDC.1998.760714.
31. You-Liang Gu, "A direct adaptive control scheme for under-actuated dynamic systems," in Proceedings of 32nd IEEE Conference on Decision and Control, pp. 1625–1627, doi:10.1109/CDC.1993.325463.
32. C.-Y. Su and Y. Stepanenko, "Sliding Mode Control of Nonholonomic Mechanical Systems: Underactuated Manipulators Case," IFAC Proc. Vol., vol. 28, no. 14, pp. 565–569, Jun. 1995, doi: 10.1016/S1474-6670(17)46888-1.
33. A. de Luca and B. Siciliano, "Trajectory control of a non-linear one-link flexible arm," Int. J. Control, vol. 50, no. 5, pp. 1699–1715, Nov. 1989, doi: 10.1080/00207178908953460.
34. M. Vidyasagar and B. D. O. Anderson, "Approximation and stabilization of distributed systems by lumped systems," Syst. Control Lett., vol. 12, no. 2, pp. 95–101, Feb. 1989, doi: 10.1016/0167-6911(89)90001-7.
35. A. Arisoy, M. Gokasan and O. S. Bogosyan, "Partial feedback linearization control of a single flexible link robot manipulator," in Proceedings of 2nd International Conference on Recent Advances in Space Technologies, 2005. RAST 2005, pp. 282–287, doi: 10.1109/RAST.2005.1512577.
36. A. M. Bloch, N. E. Leonard and J. E. Marsden, "Stabilization of the pendulum on a rotor arm by the method of controlled Lagrangians," in Proceedings 1999 IEEE

International Conference on Robotics and Automation (Cat. No.99CH36288C), vol. 1, pp. 500–505, doi: 10.1109/ROBOT.1999.770026.

37. A. S. R. Masadeh and A. Alzaqebah, “Whale Optimization Algorithm for Solving the Maximum Flow Problem,” J. Theor. Appl. Inf. Technol., vol. 96, no. 8, pp. 2208–2220, 2018.
38. M. S. K. P. Sindhuja and P. Ramamoorthy, “A Brief Survey on Nature Inspired Algorithms: Clever Algorithms for Optimization,” Asian J. Comput. Sci. Technol., vol. 7, no. 1, pp. 27–32, 2018.
39. P. R. and J. F. Miller, “A Review of Hyper-Heuristic Frameworks,” in Electronic Village Online - Evo20 Workshop, American International School of Bucharest (AISB), 2014, pp. 1–7.
40. N. K. T. El-Omari, “Sea Lion Optimization Algorithm for Solving the Maximum Flow Problem,” IJCSNS Int. J. Comput. Sci. Netw. Secur., vol. 20, no. 8, pp. 30–67, 2020.
41. X. S. Yang, “Nature-Inspired Metaheuristic Algorithms,” First Edition, Luniver Press, UK. 2008.
42. M. Gavrilas, “Heuristic and metaheuristic optimization techniques with application to power systems,” Int. Conf. Math. Methods Comput. Tech. Electr. Eng. - Proc., pp. 95–103, 2010.
43. Polya, “Reviewed Work: How to Solve It by G. Pólya,” The Mathematical Gazette, vol. 30. p. 181, 1945, doi: 10.2307/3609122.
44. V. N. Drozdov, V. A. Kim and L. B. Lazebnik, “Modern approach to the prevention and treatment of NSAID gastropathy,” no. 2. 2011.
45. Y. Wang, “A Socio psychological Perspective on Collective Intelligence in Metaheuristic Computing,” Int. J. Appl. Metaheuristic Comput., vol. 1, no. 1, pp. 110–128, 2010, doi:10.4018/jamc.2010102606
46. J. H. Holland, “Adaptation in Natural and Artificial Systems,” University of Michigan Press, Ann Arbor. 2nd Edition, MIT Press, 1992.
47. M. Dorigo, M. Birattari and T. Stutzle, "Ant colony optimization," in IEEE Computational Intelligence Magazine, vol. 1, no. 4, pp. 28-39, Nov. 2006, doi: 10.1109/MCI.2006.329691.
48. Kennedy, J. and Eberhart, R. “Particle Swarm Optimization,” Proceedings of the IEEE International Conference on Neural Networks, 4, 1942-1948. http://dx.doi.org/10.1109/ICNN.1995.488968
49. S. Harifi et al. “Giza Pyramids Construction: an ancient-inspired metaheuristic algorithm for optimization.” Evol. Intel. 14, 1743–1761 (2021). https://doi.org/10.1007/s12065-020-00451-3
50. R.V. Rao, et al. “Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems,” Computer-Aided Design, Volume 43, Issue 3, 2011, Pages 303-315, ISSN 0010-4485, https://doi.org/10.1016/j.cad.2010.12.015
51. Rao et al. “Optimization of advanced finishing processes using teaching-learning based optimization algorithm,” Nano finishing Science and Technology (Ed. V.K.Jain), New York: Taylor and Francis, 2015.
52. Rao et al. “Multi-objective design optimization of a robot gripper using TLBO technique,” Proceedings of the Second Indo-Russian Joint Workshop on Computational Intelligence, Modern Heuristics in Automation and Robotics, Novosibirsk State Technical University, Russia, 10-13 September, 184-188, 2011.
53. Baghlani et al. “Teaching-learning-based optimization algorithm for shape and size optimization of truss structures with dynamic frequency constraints.” IJST Transactions of Civil Engineering, 37(1), 409-421, 2013.

54. Biswas et al. “Cooperative co-evolutionary teaching-learning based algorithm with a modified exploration strategy for large scale global optimization.” Swarm, Evolutionary, and Memetic Computing, Lecture Notes in Computer Science, 7677, 467-475, 2012.
55. Gonzalez-Alvarez et al. “Multi-objective teaching-learning-based optimization (MO-TLBO) for motiffinding.” 13th IEEE International Symposium on Computational Intelligence and Informatics, Budapest, Hungary, 2012. doi:10.1109/cinti.2012.6496749.
56. Kundu et al. “Selective teaching-learning based niching technique with local diversification strategy.” Swarm, Evolutionary and Memetic Computing, Lecture Notes in Computer Science, 7677, 160-168, 2012.
57. Mohapatra et al “Optimal placement of capacitors in distribution networks using a modified teaching-learning based algorithm,” Swarm, Evolutionary, and Memetic Computing, Lecture Notes in Computer Science, 7677, 484-490, 2012.
58. Rajasekhar et al. “Elitist teaching learning opposition based algorithm for global optimization,” Proceedings of IEEE International Conference on Systems, Man and Cybernetics, Seoul, 2012. doi:10.1109/icsmc.2012.6377882.
59. Rao et al. “An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems,” International Journal of Industrial Engineering Computations, 3(4), 535-560, 2012.
60. V. Toğan, “Design of planar steel frames using teaching-learning based optimization,” Engineering Structures, 34, 225-234, 2012.
61. Y. H. Cheng, “A novel optimization method for picking PCR oligonucleotide primers.” International Journal of Computer Science and Electronics Engineering, 1(4), 518-523, 2013.
62. Daljit et al. “A design of IIR based digital hearing aids using teaching-learning-based optimization.” International Journal of Computer Engineering & Applications, 3(2-3), 180-190, 2013.
63. T. Dede, “Optimum design of grillage structures to LRFD-AISC with teaching-learning based optimization,” Structure and Multidisciplinary Optimization, 48(5), 955-964, 2013
64. Degertekin et al. “Sizing truss structures using teaching learning based optimization. Computers & Structures,” 119(1), 177-188, 2013.
65. García et al. “Optimal distributed generation location and size using a modified teaching-learning-based optimization algorithm,” International Journal of Electrical Power & Energy Systems, 50, 65-75, 2013.
66. X. Jiang & J. Zhou “Hybrid DE-TLBO algorithm for solving short term hydro-thermal optimal scheduling with incommensurable Objectives,” Proceedings of IEEE 32nd Chinese Control Conference, 26-28 July, Xi'an, 2474-2479, 2013.
67. Krishnanand et al. “Optimal power flow solution using self–evolving brain–storming inclusive teaching–learning–based algorithm,” In: Proceeding of International Conference on Swarm Intelligence, Lecture Notes in Computer Science, 7928, 338-345, 2013.
68. V.R. Kumar, “Teaching learning based optimization applied to mechanical constrained design problems,” M. Tech. Thesis, NIT Rourkela, India, 2013.
69. Li et al. “Model NOx emissions by least squares support vector machine with tuning based on ameliorated teaching–learning-based optimization” Chemo metrics and Intelligent Laboratory Systems, 126, 11-20, 2013.
70. Mandal et al “Optimal reactive power dispatch using quasi oppositional teaching learning based optimization,” Electrical Power and Energy Systems,53, 123-134, 2013.

71. M. A. Medina et al. “Reactive Power Handling by a Multi-objective Teaching Learning Optimizer Based on Decomposition,” IEEE Transactions on Power Systems, 28(4), 3629-3637, 2013
72. T. Niknam et al. “A new modified teaching-learning algorithm for reserve constrained dynamic economic dispatch.” IEEE Transactions on Power Systems, 28(2), 749-763, 2013
73. P. J. Pawar & R. V. Rao, “Parameter optimization of machining processes using teaching–learning-based optimization algorithm,” International Journal of Advanced Manufacturing Technology, 67(5-8), 995-1006, 2013.
74. R.V. Rao, V.D. Kalyankar, “Multi-pass turning process parameter optimization using teaching–learning-based optimization algorithm,” Scientia Iranica, 20(3), 967–974, 2013.
75. P. K. Roy, “Teaching learning based optimization for short-term hydrothermal scheduling problem considering valve point effect and prohibited discharge constraint,” International Journal of Electrical Power & Energy Systems, 53, 10-19, 2013.
76. S. C. Satapathy et al. “A teaching learning based optimization based on orthogonal design for solving global optimization problems” Springer Plus, 2(130), 1–12, 2013.
77. Satapathy et al. “Weighted teaching-learning-based optimization for global function optimization.” Applied Mathematics, 4, 429-439, 2013
78. R. Singh et al. “Teaching-learning-based optimization algorithm for parameter identification in the design of IIR filters.” Journal of The Institution of Engineers (India): Series B, 94(4), 285-294, 2013.
79. D. Tang, “An improved teaching-learning-based optimization algorithm with memetic method for global optimization.” International Journal of Advancements in Computing Technology, 5(9) 942-949, 2013.
80. Theja et al. “An optimal design of coordinated PI based PSS with TCSC controller using modified teaching learning based optimization.” Proceedings of World Congress on Nature and Biologically Inspired Computing, 2013. doi:10.1109/NaBIC.2013.6617845.
81. S. Tuo et al. “An improved harmony search based on teaching-learning strategy for unconstrained optimization problems,” Mathematical Problems in Engineering, 2013. http://dx.doi.org/10.1155/2013/413565.
82. Waghmare G., “A Note on Teaching Learning Based Optimization Algorithm” Information Sciences, 229, 159-169, 2013
83. Wang et al. “Teaching-learning-based optimization algorithm for dealing with real-parameter optimization problems,” Applied Mechanics and Materials, 380-384, 1342-1345, 2013
84. Zou et al. “Multiobjective optimization using teaching learning-based optimization algorithm,” Engineering Applications of Artificial Intelligence,26, 1291- 1300, 2013.
85. M. Abirami et al. “Source and transmission line maintenance outage scheduling in a power system using teaching learning based optimization algorithm.” Applied Soft Computing, 21, 72-83, 2014.
86. Alneamy et al. “Heart disease diagnosis utilizing hybrid fuzzy wavelet neural network and teaching learning based optimization algorithm.” Advances in Artificial Neural Systems, 2014. http://dx.doi.org/10.1155/2014/796323.
87. Baykasoğlu et al. “Testing the performance of teaching-learning based optimization (TLBO) algorithm on combinatorial problems: Flowshop and job shop scheduling cases.” Information Sciences, 276(20), 204-218, 2014.
88. Bhattacharjee et al. “Teaching-learning-based optimization for different economic dispatch problems.” Scientia Iranica, 21(3), 870-884, 2014.

89. Bouchekara et al. “Optimal power flow using teaching learning-based optimization technique.” Electric Power Systems Research, 114, 49-59, 2014.
90. T. Dede “Application of teaching-learning-based-optimization algorithm for the discrete optimization of truss structures.” KSCE Journal of Civil Engineering,18(6), 1759-1767, 2014.
91. Ghasemi et al. “Modified teaching learning algorithm and double differential evolution algorithm for optimal reactive power dispatch problem: A comparative study.” Information Sciences, 278, 231-249, 2014.
92. González-Álvarez et al. “Predicting DNA motifs by using evolutionary multiobjective optimization,” IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, 1–13, 2014.
93. Hoseini et al. “A new multi objective optimization approach in distribution systems.” Optimization Letters, 8, 181-199, 2014.
94. W. H. Lim & N.A.M. Isa, “Bidirectional teaching and peer-learning particle swarm optimization.” Information Sciences, 280, 111-134, 2014.
95. Lin et al. “Multiobjective teaching–learning-based optimization algorithm for reducing carbon emissions and operation time in turning operations.” Engineering Optimization, 2014 doi:10.1080/0305215X.2014.928818
96. M. Mardaneh & F. Golestaneh, “Harmonic optimization of diode-clamped multilevel inverter using teaching-learning-based optimization algorithm.” IETE Journal of Research, 59(1), 9-16, 2014.
97. Patel et al. “Extraction of solar cell parameters from a single current–voltage characteristic using teaching learning based optimization algorithm.” Applied Energy, 119, 384-393, 2014.
98. Pholdee et al. “Efficient hybrid evolutionary algorithm for optimization of a strip coiling process.” Engineering Optimization, 47(4), 521-532, 2014.
99. R.V. et al. “Advanced optimal tolerance design of machine elements using teachinglearning-based optimization algorithm.” Production & Manufacturing Research: An Open Access Journal, 2(1), 71-94, 2014.
100. Rao et al. “Parameters optimization of selected casting processes using teaching–learning-based optimization algorithm.” Applied Mathematical Modelling, 38, 5592-5608, 2014.
101. Rao R.V. & Patel V. “A multiobjective improved teaching-learning based optimization algorithm for unconstrained and constrained optimization problems.” International Journal of Industrial Engineering Computations 5(1), 1-22, 2014.
102. Roy et al. “Solution of unit commitment problem using quasi-oppositional teaching learning based algorithm.” Electrical Power and Energy Systems, 60, 96-106, 2014.
103. Roy et al. “Oppositional teaching learning based optimization approach for combined heat and power dispatch.” Electrical Power and Energy Systems, 57, 392-403, 2014.
104. Satapathy et al. “Modified teaching-learning-based optimization algorithm for global numerical optimization — A comparative study.” Swarm and Evolutionary Computation, 16, 28-37, 2014.
105. Shabanpour-Haghighi et al. “A modified teaching learning based optimization for multiobjective optimal power flow problem.” Energy Conversion and Management, 77, 597-607, 2014.
106. Wang et al. “A hybridization of teaching–learning-based optimization and differential evolution for chaotic time series prediction.” Neural Computing and Applications, 25(6), 1407-1422, 2014.

107. Xiao et al. “Application of modified teachinglearning algorithm in coordination optimization of TCSC and SVC.” Pattern Recognition, Communications in Computer and Information Sciences, 483, 44-53, 2014.
108. Xia et al. “Disassembly sequence planning using a simplified teaching–learning-based optimization algorithm.” Advanced Engineering Informatics, 28, 518-527, 2014.
109. B. Xing & W. J. Gao, “Teaching-learning-based optimization algorithm. Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms.” Intelligent Systems Reference Library, 62, 211-216, 2014
110. Yang et al. “A new compact teaching-learning-based optimization method.” Intelligent Computing Methodologies, Lecture Notes in Computer Science, 8589, 717-726, 2014.
111. Wang et al. “An improved teaching-learning-based optimization algorithm for numerical and engineering optimization problems.” Journal of Intelligent Manufacturing, 2014 doi:10.1007/s10845-014-0918-3.
112. Chen et al. “A teaching–learning-based optimization algorithm with producer–scrounger model for global optimization.” Soft Computing, 19, 745-762, 2015.
113. S.P. Das & S. Padhy, “A novel hybrid model using teaching–learning-based optimization and a support vector machine for commodity futures index forecasting.” International Journal of Machine Learning and Cybernetics, 2015. doi:10.1007/s13042-015-0359-0.fs.
114. Durai et al. “Improved parameters for economic dispatch problems by teaching-learning-based optimization.” International Journal of Electrical Power & Energy Systems, 67, 11-24, 2015.
115. Huang et al. “An effective teaching-learning-based cuckoo search algorithm for parameter optimization problems in structure designing and machining processes.” Applied Soft Computing, 36, 349-356, 2015.
116. H. Hosseinpour & B. Bastaee, “Optimal placement of on-load tap changers in distribution networks using SA-TLBO method.” International Journal of Electrical Power & Energy Systems, 64, 1119-1128, 2015.
117. Kanwar et al. “Simultaneous allocation of distributed resources using improved teaching learning based optimization.” Energy Conversion and Management, 103, 387-400, 2015.
118. Kumar “Parametric appraisal and optimization in machining of CFRP composites by using TLBO (teaching–learning based optimization algorithm).” Journal of Intelligent Manufacturing, 2015. doi:10.1007/s10845-015-1050-8.
119. Kurada et al. “Automatic teaching–learning-based optimization: A novel clustering method for gene functional enrichments.” Computational Intelligence Techniques for Comparative Genomics, Springer Briefs in Applied Sciences and Technology, 2015. doi:10.1007/978- 981-287-338-5_217-35.
120. Mukhopadhyay et al. “Optimal location of TCSCusing opposition teaching learning based optimization.” International Journal of Energy Optimization and Engineering, 4(1), 708-224, 2015.
121. Rao R.V. & More K.C. “Optimal design of the heat pipe using TLBO (teaching-learning-based optimization) algorithm.” Energy, 80, 535-544, 2015.
122. Rao R.V. & Waghmare G. “Design optimization of robot grippers using teaching-learning based optimization algorithm.” Advanced Robotics, 29(6), 431-447, 2015.

123. Rao R.V. & Waghmare G. "Optimization of thermal performance of a smooth flat-plate solar air heater using teaching–learning-based optimization algorithm." Cogent Engineering, 2(1), 1-28, 2015.
124. Rao R.V. & Waghmare G. "Multiobjective design optimization of a plate-fin heat sink using a teaching-learning-based optimization algorithm." Applied Thermal Engineering, 76, 521-529, 2015.
125. Sahu et al. "Automatic generation control of multi-area power systems with diverse energy sources using teaching learning based optimization algorithm." Engineering Science and Technology, An International Journal, 2015. doi:10.1016.j.jestech.2015.07.011.
126. O. E. Turgut, M.T. Coban, :Optimal proton exchange membrane fuel cell modelling based on hybrid Teaching Learning Based Optimization – Differential Evolution algorithm." Ain Shams Engineering Journal, 2015. doi:10.1016.j.asej.2015.05.003.
127. Wang et al. "An effective teaching–learning-based optimization algorithm for the flexible job-shop scheduling problem with fuzzy processing time." Neuro-computing, 148, 26-268, 2015.
128. Zhang et al. "An integrated approach for real-time model-based state-of-charge estimation of lithium-ion batteries." Journal of Power Sources, 283(1), 24-36, 2015.
129. B. Brogliato, & R. Lozano, "Adaptive control of first-order nonlinear systems with reduced knowledge of the plant parameters." IEEE Trans. Autom. Control 39, 1764–1768, 1994.
130. Grimm et al. "Nominally robust model predictive control with state constraints." IEEE Trans. Autom. Control 52, 1856–1870, 2007.
131. J. Kaloust, Z. Qu, "Robust control design for nonlinear uncertain systems with an unknown time-varying control direction." IEEE Trans. Autom. Control 42, 393–399, 1997.
132. M. Lazar, W. Heemels, "Predictive control of hybrid systems: input-to-state stability results for sub-optimal solutions." Automatica 45, 180–185, 2009.
133. Rakovic et al. "Simple robust control invariant tubes for some classes of nonlinear discrete time systems." In: Proceedings of the 45th IEEE Conference on Decision and Control. IEEE, pp. 6397–6402, 2006.
134. Bharat et al. "A review on tuning methods for PID controller." Asian J. Converg. Technol, 2019
135. E. Abbasi, N. Naghavi, "Offline auto-tuning of a PID controller using extended classifier system (XCS) algorithm." J. Adv. Comput. Eng. Technol. 3, 41–44, 2017
136. M. Kushwah, A. Patra, "Tuning PID controller for speed control of DC motor using soft computing techniques-a review." Adv. Electron. Electr. Eng. 4, 141–148, 2014
137. J. Ziegler, N. Nichols, "Optimum settings for automatic controllers." 1993.
138. J. Ziegler, N. Nichols, "Optimum settings for automatic controllers." Trans. Am. Soc. Mech. Eng. 64, 1942.
139. K.J. Åström, T. Hägglund, "PID Controllers: Theory, Design, and Tuning," vol. 2. Instrument Society of America Research, Triangle Park, NC, 1992.
140. R. Lozano, A. Valera, P. Albertos, S. Arimoto, T. Nakayama, "PD control of robot manipulators with joint flexibility, actuators dynamics and friction." Automatica 35, 1697–1700, 1999.
141. M. Johnson, M. Moradi, "PID Control." Springer, London, 2005.

142. M. Moradi, “New techniques for PID controller design.” In: Proceedings of 2003 IEEE Conference on Control Applications. CCA 2003, vol. 2. IEEE, pp. 903–908.
143. Bansal et al. “PID controller tuning techniques: a review.” J. Control Eng. Technol. 2, 168–176, 2012.
144. K. Latha, V. Rajinikanth, P.M. Surekha, “PSO-based PID controller design for a class of stable and unstable systems.” In: ISRN Artificial Intelligence, 2013.
145. L. F. Fraga-Gonzalez, R. Q. Fuentes-Aguilar, A. Garcia-Gonzalez, G. Sanchez-Ante, G., “Adaptive simulated annealing for tuning PID controllers.” AI Commun. 30, 347–362, 2017.
146. K. Krishnakumar, D.E. Goldberg, “Control system optimization using genetic algorithms." J. Guid. Control Dyn. 15, 735–740, 1992.
147. Tang et al. "Genetic algorithms: concepts and applications [in engineering design].” IEEE Trans. Ind. Electron. 43, 519–534, 1996.
148. A. T. El-Deen, A.A.H. Mahmoud, A.R. El-Sawi, “Optimal PID tuning for DC motor speed controller based on genetic algorithm.” Int. Rev. Autom. Control 8, 80–85, 2015.
149. S. B. Joseph, G.D. Emmanuel, “Automatic tuning of proportional integral derivative controller using genetic algorithm." Pac. J. Sci. Technol. 19, 51–57, 2018.
150. Mohammed et al. “Tuning of PID controller of synchronous generators using genetic algorithm.” In: 2014 IEEE International Conference on Mechatronics and Automation. IEEE, pp. 1544–1548, 2014.
151. N. Patil, G. Lakhekar, “Design of PID controller for cascade control process using genetic algorithm.” In: 2017 International Conference on Intelligent Computing and Control Systems. ICICCS. IEEE, pp. 1089–1095, 2017.
152. J.M. Amaral, R. Tanscheit, M.A. Pacheco, “Tuning PID controllers through genetic algorithms.” Complex Syst. 2, 3, 2018.
153. M. Dorigo, “Optimization, learning and natural algorithms.” PhD thesis. Politecnico di Milano, Italy, 1992.
154. Sandoval et al. “Control of direct current motor using ant colony optimization.” In: 2015 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies. CHILECON. IEEE, pp. 79–82, 2015.
155. Priyambodo et al. “Optimizing control based on ant colony logic for quadrotor stabilization.” In: 2015 IEEE International Conference on Aerospace Electronics and Remote Sensing Technology. ICARES. IEEE, pp. 1–4, 2015.
156. M. Aabid, A. Elakkary, N. Sefiani, “PID parameters optimization using ant-colony algorithm for human heart control.” In: 2017 23rd International Conference on Automation and Computing. ICAC. IEEE, pp. 1–6, 2017.
157. Kaliannan et al. “Ant colony optimization algorithm based PID controller for LFC of single area power system with non-linearity and boiler dynamics.” World J. Model. Simul. 12, 3–14, 2016.
158. Herlambang et al. “Particle swarm optimization (PSO) and ant colony optimization (ACO) for optimizing PID parameters on autonomous underwater vehicle (AUV) control system.” In: Journal of Physics: Conference Series. In: IOP Publishing, vol. 1211. 012039, 2019
159. A.S. Yunus, M.R. Djalal, “Optimal tuning of PID control on single machine infinite bus using ant colony optimization.” In: 2019 International Conference on Technologies and Policies in Electric Power & Energy. IEEE, pp. 1–6, 2019.
160. Sim et al. “Optimization of PID parameters using ant colony algorithm for position control of DC motor.” In: 2019 8th International Conference on Renewable Energy Research and Applications. ICRERA. IEEE, pp. 1047–1051, 2019.

161. Chiha et al. “Tuning PID controller using multiobjective ant colony optimization.” Appl. Comput. Intell. Soft Comput. 2012.
162. D. Karaboga, “An idea based on honey bee swarm for numerical optimization.” Technical Report Technical report-tr06. Erciyes University, Engineering Faculty, 2005.
163. D. Karaboga, “Artificial bee colony algorithm." Scholarpedia 5, 6915, 2010.
164. Abachizadeh et al. “Optimal tuning of PID controllers using artificial bee colony algorithm.” In: 2010 IEEE/ASME International Conference on Advanced Intelligent Mechatronics. IEEE, pp. 379–384, 2010.
165. M.E. El-Telbany, “Tuning PID controller for DC motor: an artificial bees optimization approach.” Int. J. Comput. Appl. 77, 2013.
166. A. Bagis, H. Senberber, “ABC algorithm based PID controller design for higher order oscillatory systems.” Elektron. Elektrotech. 23, 3–9, 2017.
167. H. Senberber, A. Bagis, “Fractional PID controller design for fractional order systems using ABC algorithm.” In: 2017 Electronics. IEEE, pp. 1–7, 2017.
168. Annisa et al. “Implementation of PID based controller tuned by evolutionary algorithm for double link flexible robotic manipulator.” In: 2018 International Conference on Computational Approach in Smart Systems Design and Applications. ICASSDA. IEEE, pp. 1–5, 2018.
169. S. K. Valluru, M. Singh, “Investigation of NARMA L-2 and artificial bee colony tuned PID controllers for bench scaled nonlinear dynamical system.” Int. J. Control Theory Appl. 10, 363–374, 2017.
170. D. Zhi, “Optimization of PID controller for single phase inverter based on ABC.” In: 2019 5th International Conference on Control, Automation and Robotics. ICCAR. IEEE, pp. 501–504, 2019.
171. X.S. Yang, “A new metaheuristic bat-inspired algorithm. In: Nature Inspired Cooperative Strategies for Optimization.” NICSO 2010. Springer, pp 65-74, 2010.
172. A.H. Gandomi, X.S. Yang, “Chaotic bat algorithm.” J. Comput. Sci. 5, 224–232, 2014.
173. N. Katal, P. Kumar, S. Narayan, “Optimal PID controller for coupled-tank liquid level control system using bat algorithm.” In: 2014 International Conference on Power, Control and Embedded Systems. ICPCES. IEEE, pp. 1–4, 2014
174. R. Kotteeswaran, L. Sivakumar, “Optimal partial-retuning of decentralised PI controller of coal gasifier using bat algorithm.” In: International Conference on Swarm, Evolutionary, and Memetic Computing. Springer, pp. 750–761, 2013.
175. K. Singh, P. Vasant, I. Elamvazuthi, R. Kannan, “PID tuning of servo motor using bat algorithm.” Proc. Comput. Sci. 60, 1798–1808, 2015.
176. K. Premkumar, B. Manikandan, “Bat algorithm optimized fuzzy pd based speed controller for brushless direct current motor.” Int. J. Eng. Sci. Technol. 19, 818–840, 2016.
177. M. Rahmani, A. Ghanbari, M.M. Ettefagh, “Robust adaptive control of a bioinspired robot manipulator using bat algorithm.” Expert Syst. Appl. 56, 164–176, 2016.
178. Rahmani et al. “Optimal novel super twisting PID sliding mode control of a MEMS gyroscope based on multi-objective bat algorithm.” Microsyst. Technol. 24, 2835–2846, 2018
179. B.S.G. de Almeida, V.C. Leite, “Particle swarm optimization: a powerful technique for solving engineering problems.” In: Ser, J.D., Villar, E., Osaba, E. (Eds.), Swarm Intelligence. IntechOpen, Rijeka. Chapter 3, 2019

180. M. Eshtay, H. Faris, N. Obeid, “Improving extreme learning machine by competitive swarm optimization and its application for medical diagnosis problems.” Expert Syst. Appl. 104, 134–152, 2018.
181. Shie et al. “Prediction of corporate financial distress: an application of the America banking industry.” Neural Comput. Appl. 21, 1687–1696, 2012.
182. K.Y. Lee, J.B. Park, “Application of particle swarm optimization to economic dispatch problem: advantages and disadvantages.” In: 2006 IEEE PES Power Systems Conference and Exposition. IEEE, pp. 188–192, 2006.
183. Cho et al. “Optimizing tactical military manets with a specialized pso.” In: IEEE Congress on Evolutionary Computation. IEEE, pp. 1–6, 2010.
184. A. Khare, S. Rangnekar, “A review of particle swarm optimization and its applications in solar photovoltaic system.” Appl. Soft Comput. 13, 2997–3006, 2013.
185. M. Hajihassani, D.J. Armaghani, R. Kalatehjari, “Applications of particle swarm optimization in geotechnical engineering: a comprehensive review.” Geotech. Geolog. Eng. 36, 705–722, 2018.
186. S. Xia, L. Li, “Application of greenhouse temperature prediction based on pso-rbf neural network.” Comput. Eng. Des. 38, 744–748, 2017
187. X.S. Yang, S. Deb, “Cuckoo search via Lévy flights.” In: 2009 World Congress on Nature & Biologically Inspired Computing. NaBIC. IEEE, pp. 210–214, 2009.
188. P. Kumar, S. Nema, P. Padhy, “PID controller for nonlinear system using cuckoo optimization.” In: 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies. ICCICCT. IEEE, pp. 711–716.
189. Z. Bingul, O. Karahan, “A novel performance criterion approach to optimum design of PID controller using cuckoo search algorithm for avr system.” J. Franklin Inst. 355, 5534–5559, 2018.
190. P. Verma, N. Patel, N. Nair, A. Sikander, “Design of PID controller using cuckoo search algorithm for buck-boost converter of led driver circuit.” In: 2016 IEEE 2nd Annual Southern Power Electronics Conference. SPEC. IEEE, pp. 1–4.
191. K. Nimisha, R. Senthilkumar, “A survey on optimal tuning of PID controller for buck-boost converter using cuckoo-search algorithm.” In: 2018 International Conference on Control, Power, Communication and Computing Technologies. ICCPCCT. IEEE, pp. 216–221.
192. S. Mirjalili, S.M. Mirjalili, A. Lewis, “Grey wolf optimizer.” Adv. Eng. Softw. 69, 46–61, 2014.
193. J.A. Goldbogen, A.S. Friedlaender, J. Calambokidis, M.F. Mckenna, M. Simon, D.P. Nowacek, “Integrative approaches to the study of baleen whale diving behaviour, feeding performance, and foraging ecology.” BioScience 63, 90–100, 2013
194. Li et al. “An enhanced grey wolf optimization based feature selection wrapped kernel extreme learning machine for medical diagnosis.” Comput. Math. Methods Med. 2017.
195. Yadav et al. “Optimized PID controller for magnetic levitation system,” IFAC-PapersOnLine 49, 778–782, 2016.
196. Das et al. “Optimal tuning of PID controller using GWO algorithm for speed control in DC motor.” In: 2015 International Conference on Soft Computing Techniques and Implementations. ICSCTI. IEEE, pp. 108–112.
197. M.A. Sen, M. Kalyoncu, “Optimal tuning of PID controller using grey wolf optimizer algorithm for quadruped robot.” Balkan J. Electr. Comput. Eng. 6, 29–35, 2018.
198. Agarwal et al. “Analysis of grey wolf optimizer based fractional order PID controller in speed control of DC motor.” Microsyst. Technol. 24, 4997–5006, 2018.

199. Yadav et al. “Tuning of parameters of PID controller using grey wolf optimizer." Available at SSRN 3575432, 2020.
200. S.K. Verma, Nagar, “Design and optimization of fractional order PIλ Dμ controller using grey wolf optimizer for automatic voltage regulator system." Recent Adv. Electr. Electron. Eng. 11, 217–226, 2018
201. Sule et al. “Optimal tuning of proportional integral controller for fixed-speed wind turbine using grey wolf optimizer.” Int. J. Electr. Comput. Eng. 10, 5251–5261, 2020.
202. S. Yadav, S.K. Verma, S.K. Nagar, “Performance enhancement of magnetic levitation system using teaching learning based optimization.” Alex. Eng. J. 57, 2427–2433, 2018.
203. V. Rajinikanth, S.C. Satapathy, “Design of controller for automatic voltage regulator using teaching learning based optimization.” Proc. Technol. 21, 295–302, 2015.
204. V.K. Patel, V.J. Savsani, “A multi-objective improved teaching–learning based optimization algorithm (MO-ITLBO).” Inf. Sci. 357, 182–200, 2016.
205. Chen et al. “Quadratic interpolation based teaching-learning-based optimization for chemical dynamic system optimization.” Knowl.-Based Syst. 145, 250–263, 2018.
206. Chen et al. “Teaching-learning-based optimization with learning enthusiasm mechanism and its application in chemical engineering.” J. Appl. Math. 2018.
207. L. Keviczky and C. Bányász, “Two-Degree-of-Freedom Control Systems.” Elsevier, 2015.
208. S. Krafes, Z. Chalh and A. Saka, “A Review on the Control of Second Order Underactuated Mechanical Systems,” Complexity, vol. 2018, pp. 1–17, Dec. 2018, doi: 10.1155/2018/9573514.
209. A. A. Aly and H. Ohuchi, “A Cross-Coupled Control of a Two-DOF Robot System Using Fuzzy Logic Technique,” Int. J. Control. Autom. Syst., vol. 4, no. 1, 2015. http://www.researchpub.org/journal/jac/jac.html.
210. T. Shen, C. A. Nelson and J. Bradley, “Design of a Model-Free Cross-Coupled Controller with Application to Robotic NOTES,” J. Intell. Robot. Syst., vol. 95, no. 2, pp. 473–489, Aug. 2019, doi: 10.1007/s10846-018-0836-2.
211. J. Yang, J. Xu, and Z. Li, “Two-Degree-of-Freedom Based Cross-Coupled Control for 196 High-Accuracy Tracking Systems,” in 2007 IEEE International Conference on Automation Science and Engineering, Sep. 2007, pp. 950–955, doi: 10.1109/COASE.2007.4341672.
212. P. Ouyang, Y. Hu, W. Yue, and D. Liu, “Cross-Coupled Contouring Control of Multi DOF Robotic Manipulator,” Algorithms, vol. 9, no. 4, p. 81, Nov. 2016, doi: 10.3390/a9040081.
213. P. R. Ouyang, V. Pano, and J. Acob, “Contour tracking control for multi-DOF robotic manipulators,” in 2013 10th IEEE International Conference on Control and Automation (ICCA), Jun. 2013, pp. 1491–1496, doi: 10.1109/ICCA.2013.6564888.
214. K. L. Barton and A. G. Alleyne, “A Cross-Coupled Iterative Learning Control Design for Precision Motion Control,” IEEE Trans. Control Syst. Technol., vol. 16, no. 6, pp. 1218–1231, Nov. 2008, doi: 10.1109/TCST.2008.919433.
215. M. A. Khanesar and E. Kayacan, “Controlling the Pitch and Yaw Angles of a 2-DOF Helicopter Using Interval Type-2 Fuzzy Neural Networks,” 2015, pp. 349–370.
216. G. Joselin Retna Kumar, N. Showme, M. Aravind, and R. Akshay, “Design and control of ball on plate system,” Int. J. Control Theory Appl., vol. 9, no. 34, pp. 765–778, 2016.

217. H. Wang, C. Vasseur, V. Koncar, A. Chamroo, and N. Christov, “Modelling and trajectory tracking control of a 2-DOF vision based inverted pendulum,” Control Eng. Appl. Informatics, vol. 12, no. 3, pp. 59–66, 2010.
218. W. Fan, H. Lu, X. Zhang, Y. Zhang, R. Zeng, and Q. Liu, “Two-degree-of-freedom dynamic model-based terminal sliding mode control with observer for dual-driving feed stage,” Symmetry (Basel)., vol. 10, no. 10, 2018, doi: 10.3390/sym10100488.
219. R. Da Silveira Castro, J. V. Flores, A. T. Salton, and L. F. A. Pereira, “A comparative analysis of repetitive and resonant controllers to a servo-vision ball and plate system,” IFAC Proc. Vol., vol. 19, pp. 1120–1125, 2014, doi: 10.3182/20140824-6-ZA1003.01074.
220. J. Farzana Pasha and S. J. Mija, “Asymptotic Stabilization and Trajectory Tracking of 4 DOF Ball Balancer using LQR,” IEEE Reg. 10 Annu. Int. Conf. Proceedings/TENCON, vol. 2019-Octob, pp. 1411–1415, 2019, doi: 10.1109/TENCON.2019.8929327.
221. D. Choi, M. Kim, H. Kim, J. Choe, and M. C. Nah, “Motion Planning of Autonomous Personal Transporter using Model Predictive Control for Minimizing Non-minimum Phase Behavior,” 2018 15th Int. Conf. Ubiquitous Robot. UR 2018, pp. 362–368, 2018, doi: 10.1109/URAI.2018.8442211.
222. M. Palanisamy, H. Modares, F. L. Lewis, and M. Aurangzeb, “Continuous-Time Q Learning for Infinite-Horizon Discounted Cost Linear Quadratic Regulator Problems,” IEEE Trans. Cybern., vol. 45, no. 2, pp. 165–176, Feb. 2015, doi: 10.1109/TCYB.2014.2322116.
223. M. Kazim, A. Zaidi, L. Zhang, and A. Mehmood, “Robust backstepping control with disturbance rejection for a class of underactuated systems,” Proc. IECON 2017 - 43rd Annu. Conf. IEEE Ind. Electron. Soc., vol. 2017-Janua, pp. 3104–3109, 2017, doi: 10.1109/IECON.2017.8216524.
224. S. Rudra, R. K. Barai, and M. Maitra, “Block backstepping design of nonlinear state feedback control law for underactuated mechanical systems,” Block Backstepping Control Underactuated Mech. Syst., pp. 1–171, 2016, doi: 10.1007/978-981-10-1956-2.
225. A. Wang, X. Li, and X. Cao, “Backstepping-based robust h∞ tracking controller design for ball and plate system,” Chinese Control Conf. CCC, vol. 2019-July, no. 1, pp. 523–528, 2019, doi: 10.23919/ChiCC.2019.8866332.
226. S. A. Moezi, E. Zakeri, and M. Eghtesad, “Optimal adaptive interval type-2 fuzzy fractional-order backstepping sliding mode control method for some classes of nonlinear systems,” ISA Trans., vol. 93, pp. 23–39, 2019, doi: 10.1016/j.isatra.2019.03.006.
227. R. Singh, B. Bhushan, “Real-time control of ball balancer using neural integrated fuzzy controller.” Artif Intell Rev 53, 351–368, 2020. https://doi.org/10.1007/s10462-018-9658-7.
228. S. Sun and L. Li, “The Study of Ball and Plate System Based on Non-linear PID,” Appl. Mech. Mater. Vol. 187 pp 134-137, vol. 187, pp. 134–137, 2012, doi: 10.4028/www.scientific.net/AMM.187.134.
229. A. R. Teel, “Using Saturation to Stabilize a Class of Single-Input Partially Linear Composite Systems,” IFAC Proc. Vol., vol. 25, no. 13, pp. 379–384, Jun. 1992, doi: 10.1016/S1474-6670(17)52311-3.
230. R. Singh, B. Bhushan, “Improved ant colony optimization for achieving self-balancing and position control for balancer systems.” J Ambient Intell Human Comput 12, 8339–8356, 2021. https://doi.org/10.1007/s12652-020-02566-y

231. R. Singh, B. Bhushan, "Improving Self-Balancing and Position Tracking Control for Ball Balancer Application with Discrete Wavelet Transform-Based Fuzzy Logic Controller." Int. J. Fuzzy Syst. 23, 27–41, 2021. https://doi.org/10.1007/s40815-020-00994-8.

232. R. Singh, B. Bhushan, "Application of Stochastic Approximation for Self-tuning of PID in Unmanned Surface Vehicles." In: International Conference on Innovative Computing and Communications. Advances in Intelligent Systems and Computing, vol 1165. Springer, Singapore, 2021. https://doi.org/10.1007/978-981-15-5113-0_81.

233. H. Feng et al., "Robotic excavator trajectory control using an improved GA based PID controller," Mech. Syst. Signal Process., vol. 105, pp. 153–168, May 2018, doi: 10.1016/j.ymssp.2017.12.014.

234. Y.-Y. Hou, "Design and implementation of EP-based PID controller for chaos synchronization of Rikitake circuit systems," ISA Trans., vol. 70, pp. 260–268, Sep. 2017, doi: 10.1016/j.isatra.2017.04.016.

235. H. Liu, G. Lu, and Y. Zhong, "Robust LQR Attitude Control of a 3-DOF Laboratory Helicopter for Aggressive Maneuvers," IEEE Trans. Ind. Electron., vol. 60, no. 10, pp. 4627–4636, Oct. 2013, doi: 10.1109/TIE.2012.2216233.

236. Shaun Sawyer, "Gain-Scheduled Control of a Quadcopter UAV," University of Waterloo, 2015.

237. A. Phillips and F. Sahin, "Optimal control of a twin rotor MIMO system using LQR with integral action," in 2014 World Automation Congress (WAC), Aug. 2014, pp. 14–119, doi: 10.1109/WAC.2014.6935709.

238. A. Bayrak, F. Dogan, E. Tatlicioglu, and B. Ozdemirel, "Design of an experimental twin-rotor multi-input multi-output system," Comput. Appl. Eng. Educ., vol. 23, no. 4, pp. 578–586, Jul. 2015, doi: 10.1002/cae.21628.

239. O. U. Rehman, I. R. Petersen, and B. Fidan, "Minimax linear quadratic Gaussian control of nonlinear MIMO system with time varying uncertainties," 2012 2nd Aust. Control Conf. AUCC 2012, no. November, pp. 138–143, 2012.

240. J.-H. Yang and W.-C. Hsu, "Adaptive backstepping control for electrically driven unmanned helicopter," Control Eng. Pract., vol. 17, no. 8, pp. 903–913, Aug. 2009, doi: 10.1016/j.conengprac.2009.02.012.

241. A. Haruna, Z. Mohamed, M. Ö. Efe, and M. A. M. Basri, "Dual boundary conditional integral backstepping control of a twin rotor MIMO system," J. Franklin Inst., vol. 354, no. 15, pp. 6831–6854, Oct. 2017, doi: 10.1016/j.jfranklin.2017.08.050.

242. A. Haruna, Z. Mohamed, M. Efe, and M. A. M. Basri, "Dual boundary conditional integral backstepping control of a twin rotor MIMO system," J. Franklin Inst., vol. 354, no. 15, pp. 6831–6854, 2017, doi: 10.1016/j.jfranklin.2017.08.050.

243. K. Lee, Y. Choi, and J. Park, "Backstepping Based Formation Control of Quadrotors with the State Transformation Technique," Appl. Sci., vol. 7, no. 11, p. 1170, Nov. 2017, doi: 10.3390/app7111170.

244. B. B. Alagoz, A. Ates, and C. Yeroglu, "Auto-tuning of PID controller according to fractional-order reference model approximation for DC rotor control," Mechatronics, vol. 23, no. 7, pp. 789–797, Oct. 2013, doi: 10.1016/j.mechatronics.2013.05.001.

245. A. P. S. Ramalakshmi and P. S. Manoharan, "Non-linear modeling and PID control of twin rotor MIMO system," in 2012 IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), Aug. 2012, pp. 366–369, doi: 10.1109/ICACCCT.2012.6320804.

246. Jih-Gau Juang, Ming-Te Huang, and Wen-Kai Liu, “PID Control Using Presearched Genetic Algorithms for a MIMO System,” IEEE Trans. Syst. Man, Cybern. Part C Applications Rev., vol. 38, no. 5, pp. 716–727, Sep. 2008, doi: 10.1109/TSMCC.2008.923890.
247. P. Wen and T.-W. Lu, “Decoupling control of a twin rotor MIMO system using robust deadbeat control technique,” IET Control Theory Appl., vol. 2, no. 11, pp. 999–1007, Nov. 2008, doi: 10.1049/iet-cta:20070335.
248. F. M. Aldbrez, M. S. Alam, and M. O. Tokhi, “Input-Shaping with GA-Tuned PID for Target Tracking and Vibration Reduction,” in Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation Intelligent Control, 2005., pp. 485–490, doi: 10.1109/.2005.1467063.
249. P. E. I. Pounds and A. M. Dollar, “Stability of Helicopters in Compliant Contact Under PD-PID Control,” IEEE Trans. Robot., vol. 30, no. 6, pp. 1472–1486, Dec. 2014, doi: 10.1109/TRO.2014.2363371.
250. J. Y. Hung, W. Gao, and J. C. Hung, “Variable structure control: a survey,” IEEE Trans. Ind. Electron., vol. 40, no. 1, pp. 2–22, 1993, doi: 10.1109/41.184817.
251. V. I. Utkin and K. D. Yang, “Methods for construction of discontinuity planes in multidimensional variable structure systems,” Autom. Remote Control, vol. 39, pp. 1466–1470, 1978, doi: 10.18287/0134-2452-2015-39-4-459-461.
252. Z. Liu, X. Liu, J. Chen and C. Fang, “Altitude Control for Variable Load Quadrotor via Learning Rate Based Robust Sliding Mode Controller,” IEEE Access, vol. 7, pp. 9736–9744, 2019, doi: 10.1109/ACCESS.2018.2890450.
253. R. Rashad, A. El-Badawy, and A. Aboudonia, “Sliding mode disturbance observer based control of a twin rotor MIMO system,” ISA Trans., vol. 69, pp. 166–174, Jul. 2017, doi: 10.1016/j.isatra.2017.04.013.
254. N. Khanduja, B. Bhushan, “CSTR Control Using IMC-PID, PSO-PID, and Hybrid BBO-FF-PID Controller.” In: Malik, H., Srivastava, S., Sood, Y., Ahmad, A. (eds) Applications of Artificial Intelligence Techniques in Engineering . Advances in Intelligent Systems and Computing, vol 697. Springer, Singapore. 2019. https://doi.org/10.1007/978-981-13-1822-1_48.
255. S. Mondal and C. Mahanta, “Adaptive second-order sliding mode controller for a twin rotor multi-input–multi-output system,” IET Control Theory Appl., vol. 6, no. 14, pp. 2157–2167, Sep. 2012, doi: 10.1049/iet-cta.2011.0478.
256. A. Ben Brahim, S. Dhahri, F. Ben Hmida, and A. Sellami, “Multiplicative fault estimation-based adaptive sliding mode fault-tolerant control design for nonlinear systems,” Complexity, vol. 2018, 2018, doi: 10.1155/2018/1462594.
257. A. K. Ekbote, N. S. Srinivasan, and A. D. Mahindrakar, “Terminal Sliding Mode Control of a Twin Rotor Multiple-Input Multiple-Output System,” IFAC Proc. Vol., vol. 44, no. 1, pp. 10952–10957, Jan. 2011, doi: 10.3182/20110828-6-IT-1002.00645.
258. S. Singh, S. Janardhanan, and Mashoq-un-Nabi, “Fast terminal sliding mode control for twin rotor multi-input multi-output system,” in 2015 Annual IEEE India Conference (INDICON), Dec. 2015, pp. 1–5, doi: 10.1109/INDICON.2015.7443515.
259. J. Yu, X. Xing, H. Gao, and Z. Li, “Robust output-feedback attitude control of a three degree-of-freedom helicopter via sliding-mode observation technique,” IET Control Theory Appl., vol. 9, no. 11, pp. 1637–1643, Jul. 2015, doi: 10.1049/iet-cta.2014.1068.
260. F. Chen, R. Jiang, K. Zhang, B. Jiang, and G. Tao, “Robust Backstepping Sliding Mode Control and Observer-based Fault Estimation for a Quadrotor UAV,” IEEE Trans. Ind. Electron., pp. 1–1, 2016, doi: 10.1109/TIE.2016.2552151.

261. H. Alwi and C. Edwards, “Fault detection and fault-tolerant control of a civil aircraft using a sliding-mode-based scheme,” IEEE Trans. Control Syst. Technol., vol. 16, no. 3, pp. 499–510, 2008, doi: 10.1109/TCST.2007.906311.

262. S. P. Sadala and B. M. Patre, “A new continuous sliding mode control approach with actuator saturation for control of 2-DOF helicopter system,” ISA Trans., vol. 74, pp. 165–174, 2018, doi: 10.1016/j.isatra.2018.01.027.

263. C. Izaguirre-Espinosa, V. Parra-Vega, P. Castillo, A. J. Muñoz-Vázquez, and A. Sánchez-Orta, “Attitude control of quadrotors based on fractional sliding modes: theory and experiments,” IET Control Theory Appl., vol. 10, no. 7, pp. 825–832, Apr. 2016, doi: 10.1049/iet-cta.2015.1048.

264. A. Rahideh and M. H. Shaheed, “Constrained output feedback model predictive control for nonlinear systems,” Control Eng. Pract., vol. 20, no. 4, pp. 431–443, Apr. 2012, doi: 10.1016/j.conengprac.2011.12.003.

265. A. Rahideh and M. Hasan Shaheed, “Stable model predictive control for a nonlinear system,” J. Franklin Inst., vol. 348, no. 8, pp. 1983–2004, Oct. 2011, doi: 10.1016/j.jfranklin.2011.05.015.

266. C. Greatwood and A. G. Richards, “Reinforcement learning and model predictive control for robust embedded quadrotor guidance and control,” Auton. Robots, vol. 43, no. 7, pp. 1681–1693, Oct. 2019, doi: 10.1007/s10514-019-09829-4.

267. C. Liu, W.-H. Chen, and J. Andrews, “Piecewise constant model predictive control for autonomous helicopters,” Rob. Auton. Syst., vol. 59, no. 7–8, pp. 571–579, Jul. 2011, doi: 10.1016/j.robot.2011.04.004.

268. J. Du, K. Kondak, M. Bernard, Y. Zhang, T. Lü, and G. Hommel, “Model Predictive Control for a Small Scale Unmanned Helicopter,” Int. J. Adv. Robot. Syst., vol. 5, no. 4, pp. 34, Nov. 2008, doi: 10.5772/10583.

269. T. Oktay and C. Sultan, “Model Predictive Control of Maneuvering Helicopters,” Aug. 2012, doi: 10.2514/6.2012-4530.

270. T. D. Ngo and C. Sultan, “Model Predictive Control for Helicopter Shipboard Operations in the Ship Airwakes,” J. Guid. Control. Dyn., vol. 39, no. 3, pp. 574–589, Mar. 2016, doi: 10.2514/1.G001243.

271. J. P. Ortiz, L. I. Minchala, and M. J. Reinoso, “Nonlinear Robust H-Infinity PID Controller for the Multivariable System Quadrotor,” IEEE Lat. Am. Trans., vol. 14, no. 3, pp. 1176–1183, Mar. 2016, doi: 10.1109/TLA.2016.7459596.

272. X. C. Méndez Cubillos and L. C. G. de Souza, “Using of H-Infinity Control Method in Attitude Control System of Rigid-Flexible Satellite,” Math. Probl. Eng., vol. 2009, pp. 1–9, 2009, doi: 10.1155/2009/173145.

273. P. K. Paul and J. Jacob, “H2 Vs H∞ control of TRMS via output error optimization augmenting sensor and control singularities,” Ain Shams Eng. J., Aug. 2019, doi: 10.1016/j.asej.2019.07.001.

274. V. S. Rao, V. I. George, S. Kamath, and C. Shreesha, “Implementation of reliable H infinity observer-controller for TRMS with sensor and actuator failure,” 2015 10th Asian Control Conf. Emerg. Control Tech. a Sustain. World, ASCC 2015, pp. 1–6, 2015, doi: 10.1109/ASCC.2015.7244381.

275. A. Al-Aradi, A. Correia, D. Naiff, G. Jardim, and Y. Saporito, “Solving Nonlinear and High-Dimensional Partial Differential Equations via Deep Learning,” 2018, [Online]. Available: http://arxiv.org/abs/1811.08782.

276. F. Xu, D. Tang, and S. Wang, “Research on parallel nonlinear control system of PD and RBF neural network based on U model,” Automatika, vol. 61, no. 2, pp. 284–294, Apr. 2020, doi: 10.1080/00051144.2020.1731227.

277. G. Li and Z. Zeng, “A Neural-Network Algorithm for Solving Nonlinear Equation Systems,” in 2008 International Conference on Computational Intelligence and Security, Dec. 2008, pp. 20–23, doi: 10.1109/CIS.2008.65.

278. A. Rahideh, A. H. Bajodah, and M. H. Shaheed, “Real time adaptive nonlinear model inversion control of a twin rotor MIMO system using neural networks,” Eng. Appl. Artif. Intell., vol. 25, no. 6, pp. 1289–1297, Sep. 2012, doi: 10.1016/j.engappai.2011.12.006.

279. F. A. Shaik, S. Purwar, and B. Pratap, “Real-time implementation of Chebyshev neural network observer for twin rotor control system,” Expert Syst. Appl., vol. 38, no. 10, pp. 13043–13049, Sep. 2011, doi: 10.1016/j.eswa.2011.04.107.

280. Y. Ouyang, L. Dong, L. Xue, and C. Sun, “Adaptive control based on neural networks for an uncertain 2-DOF helicopter system with input deadzone and output constraints,” IEEE/CAA J. Autom. Sin., vol. 6, no. 3, pp. 807–815, 2019, doi: 10.1109/JAS.2019.1911495.

281. L. Hager, K. R. Uren, G. van Schoor, and A. J. van Rensburg, “Adaptive Neural network control of a helicopter system with optimal observer and actor-critic design,” Neurocomputing, vol. 302, pp. 75–90, Aug. 2018, doi: 10.1016/j.neucom.2018.04.004.

282. S. S. Ge and K.-P. Tee, “Adaptive Neural Network Control of Helicopters,” 2006, pp. 82–87.

283. C. Fu, A. Sarabakha, E. Kayacan, C. Wagner, R. John, and J. M. Garibaldi, “Input Uncertainty Sensitivity Enhanced Nonsingleton Fuzzy Logic Controllers for Long-Term Navigation of Quadrotor UAVs,” IEEE/ASME Trans. Mechatronics, vol. 23, no. 2, pp. 725–734, Apr. 2018, doi: 10.1109/TMECH.2018.2810947.

284. S. R. S. Abdullah, M. M. Mustafa, R. A. Rahman, T. O. S. Imm, and H. A. Hassan, “A fuzzy logic controller of two-position pump with time-delay in heavy metal precipitation process,” in 2011 International Conference on Pattern Analysis and Intelligence Robotics, Jun. 2011, pp. 171–176, doi: 10.1109/ICPAIR.2011.5976890.

285. R. M. Hilloowala and A. M. Sharaf, “A rule-based fuzzy logic controller for a PWM inverter in a stand alone wind energy conversion scheme,” in Conference Record of the 1993 IEEE Industry Applications Conference Twenty-Eighth IAS Annual Meeting, pp. 2066–2073, doi: 10.1109/IAS.1993.299150.

286. M. Marin-Perianu, S. Bosch, R. Marin-Perianu, H. Scholten, and P. Havinga, “Autonomous vehicle coordination with wireless sensor and actuator networks,” ACM Trans. Auton. Adapt. Syst., vol. 5, no. 4, 2010, doi: 10.1145/1867713.1867714.

287. M. Jahed and M. Farrokhi, “Robust adaptive fuzzy control of twin rotor MIMO system,” Soft Comput., vol. 17, no. 10, pp. 1847–1860, 2013, doi: 10.1007/s00500-013-1026-6.

288. A. Bounemeur, M. Chemachema, and N. Essounbouli, “Indirect adaptive fuzzy fault tolerant tracking control for MIMO nonlinear systems with actuator and sensor failures,” ISA Trans., vol. 79, no. April, pp. 45–61, 2018, doi: 10.1016/j.isatra.2018.04.014.

289. O. Castillo, L. Amador-Angulo, J. R. Castro, and M. Garcia-Valdez, “A comparative study of type-1 fuzzy logic systems, interval type-2 fuzzy logic systems and generalized type-2 fuzzy logic systems in control problems,” Inf. Sci. (Ny)., vol. 354, pp. 257–274, Aug. 2016, doi: 10.1016/j.ins.2016.03.026.

290. W. W. Tan and T. W. Chua, “Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions (Mendel, J.M.; 2001) [book review],” IEEE Comput. Intell. Mag., vol. 2, no. 1, pp. 72–73, Feb. 2007, doi: 10.1109/MCI.2007.357196.

291. J. M. Mendel and R. I. B. John, “Type-2 fuzzy sets made simple,” IEEE Trans. Fuzzy Syst., vol. 10, no. 2, pp. 117–127, Apr. 2002, doi: 10.1109/91.995115.
292. H. A. Hagras, “A Hierarchical Type-2 Fuzzy Logic Control Architecture for Autonomous Mobile Robots,” IEEE Trans. Fuzzy Syst., vol. 12, no. 4, pp. 524–539, Aug. 2004, doi: 10.1109/TFUZZ.2004.832538.
293. M. A. Sanchez, O. Castillo, and J. R. Castro, “Generalized Type-2 Fuzzy Systems for controlling a mobile robot and a performance comparison with Interval Type-2 and Type-1 Fuzzy Systems,” Expert Syst. Appl., vol. 42, no. 14, pp. 5904–5914, Aug. 2015, doi: 10.1016/j.eswa.2015.03.024.
294. O. Castillo, L. Cervantes, J. Soria, M. Sanchez, and J. R. Castro, “A generalized type-2 fuzzy granular approach with applications to aerospace,” Inf. Sci. (Ny)., vol. 354, pp. 165–177, Aug. 2016, doi: 10.1016/j.ins.2016.03.001.
295. L. Cervantes and O. Castillo, “Type-2 fuzzy logic aggregation of multiple fuzzy controllers for airplane flight control,” Inf. Sci. (Ny)., vol. 324, pp. 247–256, Dec. 2015, doi: 10.1016/j.ins.2015.06.047.
296. H. Chaoui and W. Gueaieb, “Type-2 Fuzzy Logic Control of a Flexible-Joint Manipulator,” J. Intell. Robot. Syst., vol. 51, no. 2, pp. 159–186, Feb. 2008, doi: 10.1007/s10846-007-9185-2.
297. W. Gao and R. R. Selmic, “Neural Network Control of a Class of Nonlinear Systems With Actuator Saturation,” IEEE Trans. Neural Networks, vol. 17, no. 1, pp. 147–156, Jan. 2006, doi: 10.1109/TNN.2005.863416.
298. J. I. Mulero-Martinez, “Robust GRBF Static Neurocontroller With Switch Logic for Control of Robot Manipulators,” IEEE Trans. Neural Networks Learn. Syst., vol. 23, no. 7, pp. 1053–1064, Jul. 2012, doi: 10.1109/TNNLS.2012.2196053.
299. F. Gul, W. Rahiman, and S. S. Nazli Alhady, “A comprehensive study for robot navigation techniques,” Cogent Eng., vol. 6, no. 1, pp. 1–25, 2019, doi: 10.1080/23311916.2019.1632046.
300. Faa-Jeng Lin and Po-Huan Chou, “Adaptive Control of Two-Axis Motion Control System Using Interval Type-2 Fuzzy Neural Network,” IEEE Trans. Ind. Electron., vol. 56, no. 1, pp. 178–193, Jan. 2009, doi: 10.1109/TIE.2008.927225.
301. R. J. Oentaryo, M. J. Er, S. Linn, and X. Li, “Online probabilistic learning for fuzzy inference system,” Expert Syst. Appl., vol. 41, no. 11, pp. 5082–5096, Sep. 2014, doi: 10.1016/j.eswa.2014.01.034.
302. C.-H. Chen, C.-J. Lin, and C.-T. Lin, “An efficient quantum neuro-fuzzy classifier based on fuzzy entropy and compensatory operation,” Soft Comput., vol. 12, no. 6, pp. 567–583, Apr. 2008, doi: 10.1007/s00500-007-0229-0.
303. E. Asarin, T. Dang, and A. Girard, “Hybridization methods for the analysis of nonlinear systems,” Acta Inform., vol. 43, no. 7, pp. 451–476, Jan. 2007, doi: 10.1007/s00236-006-0035-7.
304. R. Singh, B. Bhushan, “Adaptive Neuro-Fuzzy-PID and Fuzzy-PID-Based Controller Design for Helicopter System.” In: Applications of Computing, Automation and Wireless Systems in Electrical Engineering. Lecture Notes in Electrical Engineering, vol 553. Springer, Singapore, 2019. https://doi.org/10.1007/978-981-13-6772-4_25.
305. R. Singh and B. Bhushan, "Data-Driven Technique-Based Fault-Tolerant Control for Pitch and Yaw Motion in Unmanned Helicopters," in IEEE Transactions on Instrumentation and Measurement, vol. 70, pp. 1-11, 2021, Art no. 3502711, doi: 10.1109/TIM.2020.3025656.
306. F. Allouani, D. Boukhetala, and F. Boudjema, “Particle swarm optimization based fuzzy sliding mode controller for the Twin Rotor MIMO system,” in 2012 16th

IEEE Mediterranean Electrotechnical Conference, Mar. 2012, pp. 1063–1066, doi: 10.1109/MELCON.2012.6196611.
307. N. Bouarroudj, A. Djari, D. Boukhetala, and F. Boudjema, “Tuning of decentralized fuzzy logic sliding mode controller using PSO algorithm for nonlinear Twin Rotor Mimo System,” in 2017 6th International Conference on Systems and Control (ICSC), May 2017, pp. 45–50, doi: 10.1109/ICoSC.2017.7958641.
308. J. Juang, M. Huang, and W. Liu, “PID Control Using Presearched Genetic Algorithms for a MIMO System,” vol. 38, no. 5, pp. 716–727, 2008.
309. J. G. Juang, M. Te Huang, and W. K. Liu, “PID control using presearched genetic algorithms for a MIMO system,” IEEE Trans. Syst. Man Cybern. Part C Appl. Rev., vol. 38, no. 5, pp. 716–727, 2008, doi: 10.1109/TSMCC.2008.923890.
310. J.-G. Juang, R.-W. Lin, and W.-K. Liu, “Comparison of classical control and intelligent control for a MIMO system,” Appl. Math. Comput., vol. 205, no. 2, pp. 778–791, Nov. 2008, doi: 10.1016/j.amc.2008.05.061.
311. C.-W. Tao, J.-S. Taur, Y.-H. Chang, and C.-W. Chang, “A Novel Fuzzy-Sliding and Fuzzy-Integral-Sliding Controller for the Twin-Rotor Multi-Input Multi Output System,” IEEE Trans. Fuzzy Syst., vol. 18, no. 5, pp. 893–905, Oct. 2010, doi: 10.1109/TFUZZ.2010.2051447.
312. C. W. Tao, J. S. Taur, and Y. C. Chen, “Design of a parallel distributed fuzzy LQR controller for the twin rotor multi-input multi-output system,” Fuzzy Sets Syst., vol. 161, no. 15, pp. 2081–2103, Aug. 2010, doi: 10.1016/j.fss.2009.12.007.
313. M. Jahed and M. Farrokhi, “Robust adaptive fuzzy control of twin rotor MIMO system,” Soft Comput., vol. 17, no. 10, pp. 1847–1860, Oct. 2013, doi: 10.1007/s00500-013-1026-6.
314. A. S. Huaman Loayza and C. G. Perez Zuniga, “Design of a Fuzzy Sliding Mode Controller for the Autonomous Path-Following of a Quadrotor,” IEEE Lat. Am. Trans., vol. 17, no. 06, pp. 962–971, Jun. 2019, doi: 10.1109/TLA.2019.8896819.
315. M. M. System, C. Tao, S. Member, J. Taur, and Y. Chang, “A Novel Fuzzy-Sliding and Fuzzy-Integral-Sliding Controller for the Twin-Rotor Multi-Input–Multi-Output System,” vol. 18, no. 5, pp. 893–905, 2010.
316. S. Mondal and C. Mahanta, “Adaptive second-order sliding mode controller for a twin rotor multi-input–multi-output system,” IET Control Theory Appl., vol. 6, no. 14, pp. 2157–2167, 2012, doi: 10.1049/iet-cta.2011.0478.
317. M. B. Witczak, Marcin, “An LMI approach to robust fault estimation for a class of nonlinear systems,” Int. J. Robust Nonlinear Control, vol. 18, no. October 2016, pp. 1530–1538, 2015, doi: 10.1002/rnc.
318. S. Behzadimanesh, A. Fatehi, and S. Fakhimi Derakhshan, “Optimal fuzzy controller based on non-monotonic Lyapunov function with a case study on laboratory helicopter,” Int. J. Syst. Sci., vol. 50, no. 3, pp. 652–667, 2019, doi: 10.1080/00207721.2019.1567864.
319. Quanser educational solutions, “2D Ball Balancer User Manual,” Markham, Ontario, 2013.
320. M. Faulhaber, “Electronics: DC-Micromotors” Series 2338.
321. A. K. Pinagapani, G. Mani, K. R. Chandran, and K. Pandian, “Composite Disturbance Rejection Control for Ball Balancer System,” Procedia Comput. Sci., vol. 133, pp. 124–133, 2018, doi: 10.1016/j.procs.2018.07.016.
322. Quanser educational solutions, Quanser 2-DOF Helicopter Manual, Tech. rep., Quanser 2010.
323. A.S.R. Beloli, J.L. Florêncio and MSM Cavalca, “A 2 DoF low cost control workstation for control techniques application” in: 22nd International Congress of Mechanical Engineering (COBEM 2013), Brazil, 2013, pp. 3101–3112.

324. B. Zheng and Y. Zhong, “Robust attitude regulation of a 3-DOF helicopter benchmark: theory and experiments”, IEEE Trans. Ind. Electron. 58 (2) (2011) 660–670.
325. E. Bristol, "On a new measure of interaction for multivariable process control," in IEEE Transactions on Automatic Control, vol. 11, no. 1, pp. 133-134, January 1966, doi:10.1109/TAC.1966.1098266
326. M. M. Sabir and J. A. Khan, “Optimal Design of PID Controller for the Speed Control of DC Motor by Using Metaheuristic Techniques,” Adv. Artif. Neural Syst., vol. 2014, pp. 1–8, 2014, doi: 10.1155/2014/126317.
327. Lawrence, Bansode. “ Tuning of A PID Controller for Optimal Performance of Ball and Beam System.” International Journal of Engineering Research and V9, 2020. 10.17577/IJERTV9IS040009.
328. P.N. Suganthan, N. Hansen, J.J. Liang, K. Deb, Y.P. Chen, A. Auger, S. Tiwari, “Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization.” KanGAL Rep 2005005(2005):2005
329. J Liang, B Qu, P Suganthan, A.G. Herna´ndez-Dı´az, “Problem definitions and evaluation criteria for the cec, (2013) special session and competition on real-parameter optimization.” Technical Report 201212, 2013.
330. Liang et al. “Problem definitions and evaluation criteria for the cec, (2014) special session and competition on single objective real-parameter numerical optimization.” Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore 635, 2014.
331. Wu et al. “Problem definitions and evaluation criteria for the cec 2017 competition on constrained real-parameter optimization.” National University of Defense Technology, Changsha, Hunan, PR China and Kyungpook National University, Daegu, South Korea and Nanyang Technological University, Singapore, Technical Report, 2017
332. Yue et al. “Problem definitions and evaluation criteria for the cec 2020 special session and competition on single objective bound constrained numerical optimization.” Comput Intell Lab, Zhengzhou Univ., Zhengzhou, China, Tech. Rep, vol. 201911, 2020
333. Mohamed et al. “Problem definitions and evaluation criteria for the cec 2021 special session and competition on single objective bound constrained numerical optimization.” In: Tech. Rep, 2020
334. Liang et al. “Problem Definitions and Evaluation Criteria for the CEC”, Special Session on Real-Parameter Optimization. Technical Report 201212, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China, 2020
335. Ravipudi Venkata Rao et al. “An efficient Balanced Teaching-Learning-Based optimization algorithm with Individual restarting strategy for solving global optimization problems,” Information Sciences,Volume 576, 2021, Pages 68-104,ISSN 0020-0255, 2021 https://doi.org/10.1016/j.ins.2021.06.064
336. L. Wang, Y. Zhong, “Cuckoo search algorithm with chaotic maps.” Math Probl Eng 2015:1–14, 2015 https://doi.org/10.1155/2015/ 715635
337. G. Wang, L. Guo, A.H. Gandomi, G. Hao, H. Wang, “Chaotic Krill Herd algorithm.” Inf Sci (ny). 2014 https://doi.org/10.1016/j.ins.2014. 02.123
338. Su et al. “Comparisons of firefly algorithm with chaotic maps.” Comput Model New Technol 18(2):326–332, 2014

339. A. Farah, A. Belazi, “A novel chaotic Jaya algorithm for unconstrained numerical optimization.” Nonlinear Dyn 93(3):1451–1480, 2018. https://doi.org/10.1007/s11071-018-4271-5

340. M.S. Nasrabadi, Y. Sharafi, M. Tayari, “A parallel Grey Wolf Optimizer combined with Opposition based learning” In: Ist conference on swarm intelligence and evolutionary computation (CSIEC), pp 18–23, 2016.

LIST OF PUBLICATIONS:

- A. Chaudhary, B. Bhushan, “An improved teaching learning based optimization method to enrich the flight control of a helicopter system”, Sādhanā 48, 222 (2023). https://doi.org/10.1007/s12046-023-02271-4

- A. Chaudhary, B. Bhushan, “Self-Balancing and Position Control of a Balancer System Using a Pattern-Based Intelligent Optimization Method”, International Journal of Pattern Recognition and Artificial Intelligence, Vol. 37, No. 12, 2357011 (2023). https://doi.org/10.1142/S0218001423570112

- A. Chaudhary, B. Bhushan, “Novel Optimization Approach to Control the Nonlinear Vibrations of a Real Time Balancer System Under Unknown Disturbances” J. Vib. Eng. Technol. 13, 181 (2025). https://doi.org/10.1007/s42417-024-01552-4

- A. Chaudhary and B. Bhushan, "Trajectory Tracking of a 2-DOF Helicopter System using Fuzzy Controller Approach," 2021 International Conference on Emerging Techniques in Computational Intelligence (ICETCI), Hyderabad, India, 2021, pp. 159-164, https://doi.org/10.1109/ICETCI51973.2021.9574049

- A. Chaudhary, "Axis Control of a Nonlinear Helicopter Model Using Intelligent Controller," 2023 International Conference for Advancement in Technology (ICONAT), Goa, India, 2023, pp. 1-6, https://doi.org/10.1109/ICONAT57137.2023.10080707

- A. Chaudhary, “Innovative Control Framework for Nonlinear Vibrations of a Balancer System Through Hybrid FPID Controller” J. Vib. Eng. Technol. 13, 115 (2025). https://doi.org/10.1007/s42417-024-01700-w

- N. Nikita, B. Bhushan and A. Chaudhary, "Comparative Performance Analysis of Mamdani and Sugeno Fuzzy Controller using Application on 2DoF Ball Balancer System," 2023 International Conference on Power, Instrumentation, Energy and Control (PIECON), Aligarh, India, 2023, pp. 1-6, https://doi.org10.1109/PIECON56912.2023.10085809

DETAILS OF PAPERS UNDER REVIEW / PUBLICATION:

S. No.	Title	Journal / Conference	Status
1.	Addressing external disturbances in helicopter control: Enhanced TLBO for FPID tuning	IETE Journal of Research	Under Review
2.	Optimization algorithms are used to control the flight angles of a nonlinear helicopter model	IEEE 2nd International Conference on Cyber Resilience (ICCR) 2024, Dubai, UAE (Feb. 2024)	Accepted & Presented
3.	Conventional and intelligent control schemes for handling a Helicopter's pitch axis	1st International Conference on Advances in Computer Science, Electrical Electronics and Communication Technologies (CE2CT) 2025	Accepted & Presented
4.	Stabilizing a Nonlinear Helicopter Model: Advanced Hybrid Optimization Technique for Controlled Rotor Dynamics and Vibration Minimization Under External Disturbances	Journal of Vibration Engineering and Technologies	Under Review
5.	Optimized control of servo angle balancer through computational algorithms	IEEE Conference GCON 2025, NERIST, Itanagar, Arunachal Pradesh, India	Under Review

CURRICULAM VITAE

ABHISHEK CHAUDHARY

Education:

- (2020 – Pursuing) Ph.D. candidate in control system from Department of Electrical Engineering, Delhi Technological University, New Delhi.

- (2016 – 2018) Master of Technology (M. Tech) in control & instrumentation from Department of Electrical Engineering, Delhi Technological University, New Delhi.

- (2011 – 2015) Bachelor of Engineering (B.E.) in instrumentation & control engineering from ICE department, Netaji Subhas Institute of Technology, New Delhi.

Subjects Taught:

- Instrumentation and Measurement
- Digital Circuit & Systems
- Control System
- Basics of Electrical Engineering
- Instrumentation System
- Numerical Engineering and Optimization Methods

Technical Skills:

MATLAB & Simulink, Human machine interface, Quarc real-time control, Lab VIEW, PLC, C, C++, Two degree of freedom robotic systems.

Dissertation Topics:

- Ph.D. – “Design and Implementation of Intelligent Controllers on Nonlinear Systems”.
- M. Tech – “Sliding mode control on an under-actuated TORA system”.
- B. Tech – “NIR Spectroscopy Glucometer”.

Personal Information:

- Nationality – Indian
- DOB – 7th August 1993
- Language – English & Hindi
- Email – abhishek@dtu.ac.in

I hereby declare that the details mentioned above are true to the best of my knowledge.
ABHISHEK CHAUDHARY