

# INTELLIGENT CONTROL OF BALL BALANCER AND HELICOPTER SYSTEM

*Thesis submitted to the*  
Delhi Technological University

*For the award of the*  
Doctor of Philosophy

*by*

RUPAM SINGH  
(2K16/Ph.D./EE/11)

*Under the Supervision of*  
PROF. BHARAT BHUSHAN



DEPARTMENT OF ELECTRICAL ENGINEERING  
DELHI TECHNOLOGICAL UNIVERSITY  
DELHI-110042, INDIA

DECEMBER 2020

# Certificate

This is to certify that the thesis entitled “**Intelligent Control of Ball Balancer and Helicopter System**” submitted for the award of the Doctor of Philosophy is original to the best of my knowledge. The work was carried out by Ms. Rupam Singh under my guidance and has not been submitted in parts or full to this or any other University for award of any degree or diploma. All the assistance and help received during the course of study has been duly acknowledged.

Supervisor



*Prof. Bharat Bhushan*

Department of Electrical Engg.  
Delhi Technological University

Delhi-110042, India

Dated: 30/12/2020

# Acknowledgement

Writing this thesis has been fascinating and extremely rewarding. First of all, I am thankful to God, who has blessed me with the strength to acquire knowledge and learning.

I would like to express my sincere gratitude to my supervisor, **Prof. Bharat Bhushan**, for his invaluable guidance and inspiration to carry out this research work. I thank him for the scholarly inputs and consistent inspiration which I received throughout the research work. During my thesis-writing period, he was immensely helpful. I would also like to thank the reviewers of our papers for their valuable suggestions and comments which motivated me to look into areas which we never thought of venturing into.

I also wish to express my gratefulness to **Prof. Uma Nangia**, *Head, Department of Electrical Engineering, Delhi Technological University, New Delhi*, for facilitating the progress of my research work by providing the necessary facilities in the department. I would like to give special thanks to **Dr. Aniruddha Bhattacharya**, **Mr. Anil Butola**, and **Mr. Karan Singh** for their help with the laboratory setups and moral support during my study.

Besides my supervisor, I would like to thank my Ph.D. colleagues **Ms. Pallavi Verma**, and **Dr. Nikita Gupta**, for their help in accomplishing my work. I would like to extend my thanks to **Mr. V S Bharath**, and **Dr. Mohammed Ali Khan** for their technical and moral support.

I would also like to express my deep sense of gratitude to my parents, sisters, and my brothers for their care and countless calls to enquire about me. I would express my special thanks to all my family and friends for their faith, encouragement, understanding, and unfailing support.



*Rupam Singh*

# Abstract

The operation of robotic systems to perform complex tasks in dynamic environments has been a crucial area of control. Further, the technological advancements in autonomy, artificial intelligence and robotics have broad applications across society, bringing both opportunities and risks. Most of these opportunities are directly related to path tracking, speed control, maneuverability, and balancing control which are highly affected by the complexity and unpredictable dynamics of the surroundings. Besides, efficient path tracking and balancing control are particularly important for the robots, in order to achieve autonomy without any collision and disturbances. Consequently, the parameters of the mechanical and electronic components need to be monitored and optimized for performing multiple tasks and maintain the reliability of the system. In the view of these aspects, this research identified the combination of intelligent approaches and machine learning methods to achieve unprecedented path tracking and balancing control, continuous monitoring, and robustness by relying solely on onboard computing. The approaches are developed based on multiple control algorithms and are implemented with two-degree freedom operation of ball balancer and helicopter benchmark systems.

The controllers are synthesized by first applying the theory of Feedback Linearization and then enhancing their robustness properties. A brief overview of the physical interpretation of the dynamic equations, which is important to the control system designer's understanding of the system, is given. This is followed by further mathematical descriptions of the robust techniques used to augment the basic control law. The research identified the difficulties associated with current control practices and the potential improvements achievable by using nonlinear control. Initially, an intelligent approach for ball balancer position control, and unmanned helicopter trajectory tracking using wavelet fuzzy and evolving type2 quantum fuzzy neural network are developed. The wavelet transform based fuzzy controller overcomes the drawbacks of transparent interpretation of choosing fuzzy rules with techniques available in the literature. Besides, the evolving type-2 quantum fuzzy neural network is targeted at developing self-organizing and rule growing scenarios with quantum membership functions to overcome the effects of parametric uncertainties in the benchmark nonlinear system. Further, the development of probabilistic control approaches with randomized algorithms and stochastic

approximations are carried out to estimate the operation of the benchmark systems under random uncertainty conditions. Furthermore, the fault classification based reconfigurable control methods are developed by adapting wavelet transform, machine learning, and intelligent control approaches for both the systems. Finally, the reinforcement learning algorithm-based control approaches are adapted to develop model free controllers with linear quadratic regulator and neural network techniques to perform temporal feedback-based control and interleaved control respectively with the helicopter and ball balancer systems. The performance of all the developed controllers is validated through simulation studies and real-time analysis for both the ball balancer and helicopter benchmark systems.

# Contents

Certificate.....	i
Acknowledgement .....	ii
Abstract.....	iii
Contents .....	v
List of Figures .....	viii
List of Tables .....	xii
Acronyms.....	xv
Nomenclature.....	xvii
Chapter 1. Introduction .....	1
1.1 Overview .....	1
1.2 Background and existing challenges.....	1
1.2.1 Nonlinear systems.....	2
1.2.2 Cascade nonlinear systems .....	4
1.2.3 Underactuated systems.....	4
1.3 Motivation.....	6
1.4 Objectives.....	9
1.5 Methodology of the research work .....	10
1.6 Conclusion .....	13
1.7 Outline.....	13
Chapter 2. Literature Review .....	15
2.1 Modelling of 2 DoF Helicopter.....	16
2.1.1 Mathematical Modelling.....	17
2.1.2 Problem Formulation .....	20
2.2 Review of control techniques for 2DoF Helicopter System .....	22
2.3 Modelling of 2DoF Ball Balancer systems .....	31
2.3.1 Mathematical Modelling of Ball Balancer Setup.....	32
2.3.2 Problem Formulation .....	35
2.4 Review of control techniques for 2DoF Ball Balancer systems .....	36
2.5 Intelligent control of 2DoF systems.....	44
2.5.1 Neural-integrated-fuzzy controller.....	44
2.5.2 Architecture of Neural integrated Fuzzy controller .....	46
2.5.3 NiF controller for trajectory tracking of helicopter .....	47
2.5.4 NiF controller for ball balancer system .....	54
2.6 Conclusion .....	58

Chapter 3. Optimized Intelligent Control for 2DoF Systems .....	59
3.1 Wavelet Fuzzy based control system .....	59
3.1.1 Wavelet Transform .....	59
3.1.2 Wavelet Fuzzy Logic Controller.....	62
3.1.3 Stability Analysis .....	63
3.1.4 Wavelet Fuzzy Control of Helicopter system.....	67
3.1.5 Wavelet Fuzzy Control of Ball Balancer system.....	72
3.2 Evolving Fuzzy based control system.....	76
3.2.1 Basic Control Scheme.....	77
3.2.2 Evolving type-2 quantum fuzzy neural network architecture.....	78
3.2.3 Stability analysis .....	88
3.2.4 Evolving Fuzzy Control for Helicopter system .....	91
3.2.5 Evolving Fuzzy Control for Ball Balancer System.....	97
3.3 Conclusion .....	100
Chapter 4. Probabilistic Algorithms for Control of 2DoF systems .....	102
4.1 Randomized probabilistic algorithm based control development for 2DoF helicopter system .....	103
4.1.1 Randomized Algorithms .....	103
4.1.2 Randomized Algorithm for Probabilistic Controller Synthesis ...	104
4.1.3 Reachability Analysis .....	107
4.1.4 Numerical Simulation .....	109
4.1.5 Real-time Experiment .....	115
4.2 Simultaneous perturbation and stochastic approximation-based control system for ball balancer system .....	117
4.2.1 Simultaneous Perturbation Stochastic Approximation .....	117
4.2.2 Stability Analysis .....	124
4.2.3 Simulation.....	125
4.2.4 Real-Time Results.....	129
4.3 Conclusion .....	131
Chapter 5. Fault Classification-based Reconfigurable Control for 2DoF Systems .....	132
5.1 Fault classification-based control for helicopter system.....	132
5.1.2 Support vector data descriptor classifier.....	135
5.1.3 Classifier training .....	137
5.1.4 Fault-tolerant control .....	142
5.1.5 Experimental testing .....	144
5.2 Fault classification-based control for ball balancer systems.....	147
5.2.1 Problem formulation in ball balancer system.....	148
5.2.2 Support Vector Machine Classifier.....	149
5.2.3 Fault based control process.....	155
5.2.4 Real-Time Results.....	156
5.3 Conclusion .....	158
Chapter 6. Reinforcement Learning with 2DoF Systems .....	159
6.1 Reinforcement Learning Algorithms .....	159

6.2 Learning algorithm-based linear model free control .....	160
6.2.1 Learning algorithm with state measurements .....	161
6.2.2 Learning algorithm with feedback measurements .....	162
6.2.3 LQR learning algorithm with temporal difference .....	164
6.2.4 Linear model free control for trajectory tracking of helicopter system .....	171
6.3 Learning algorithm-based nonlinear model free control.....	175
6.3.1 Interleaved Learning with state measurements.....	176
6.3.2 Nonlinear model free control for ball balancer system.....	179
6.4 Conclusion .....	183
Chapter 7. Conclusion and Future Scope of Work .....	184
7.1 Introduction .....	184
7.2 Main contributions of the work.....	184
7.3 Future scope of work.....	188
List of publications .....	190
References.....	192
Curriculum vitae .....	228



# List of Figures

Figure 2.1. Experimental setup of a helicopter system.....	17
Figure 2.2. Free body diagram of 2DoF helicopter model with main and tail rotor.	17
Figure 2.3. Block diagram of couplings in the twin-rotor helicopter system .....	18
Figure 2.4. Generalized closed-loop control of pitch and yaw motors of a 2DoF helicopter System.....	22
Figure 2.5. Laboratory setup of ball balancer system.....	32
Figure 2.6. Typical wiring diagram between different components of Quanser 2DoF ball balancer system.....	32
Figure 2.7. Schematic of 2DoF ball balancer system .....	33
Figure 2.8. Open-loop block diagram of 2DoF ball balancer system.....	35
Figure 2.9. General closed-loop block diagram for ball balancer system .....	35
Figure 2.10. Schematic diagram of neural integrated fuzzy control.....	46
Figure 2.11. Block diagram of NiF controller for helicopter system.....	49
Figure 2.12. NiF membership function of the Pitch and Yaw errors and their derivatives after training .....	50
Figure 2.13. Pitch and yaw output with neural integrated fuzzy control of helicopter in Simulink.....	51
Figure 2.14. Pitch and yaw output with neural integrated fuzzy control of helicopter in real-time .....	53
Figure 2.15. Block diagram for NiF controller for ball balancer operation.....	54
Figure 2.16. NiF membership function of the position error and position rate after training .....	55
Figure 2.17. Ball balancer output with neural intgerated fuzzy control in Simulink	56
Figure 2.18. Ball balancer output with neural integaretd fuzzy control in real-time	57
Figure 3.1. Three-level discrete wavelet decomposition .....	61
Figure 3.2. Three-level discrete wavelet transform based fuzzy logic controller.....	63

Figure 3.3. Wavelet fuzzy logic controller for helicopter control .....	68
Figure 3.4. Pitch and yaw output with wavelet fuzzy control of helicopter in Simulink.....	69
Figure 3.5. Pitch and yaw output with wavelet fuzzy control of helicopter in real-time .....	71
Figure 3.6. Wavelet fuzzy control for plate angle and ball position control .....	73
Figure 3.7. Ball balancer output with wavelet fuzzy control in Simulink .....	74
Figure 3.8. Ball balancer output with wavelet fuzzy control in real-time .....	75
Figure 3.9. Architecture of evolving type-2 quantum fuzzy neural network.....	78
Figure 3.10. Interval type-2 quantum membership functions with 3 grades .....	80
Figure 3.11. Comparison of interval type-2 quantum membership functions and Gaussian membership functions .....	83
Figure 3.12. Evolving type 2 fuzzy neural network controller for helicopter control .....	92
Figure 3.13. Pitch and yaw output with evolving type 2 quantum fuzzy neural network control of helicopter in Simulink.....	93
Figure 3.14. Pitch and yaw output with evolving type 2 quantum fuzzy neural network control of helicopter in real-time .....	95
Figure 3.15. Evolving type 2 fuzzy neural network control for ball balancer system .....	97
Figure 3.16. Ball balancer output with evolving type 2 quantum fuzzy neural network control in Simulink .....	98
Figure 3.17. Ball balancer output with evolving type 2 quantum fuzzy neural network control in real-time.....	99
Figure 4.1. Generalized M- $\Delta$ model for uncertainty representation .....	103
Figure 4.2. Controller design for a fixed plant.....	104
Figure 4.3. Probabilistic Control for Helicopter System. ....	109
Figure 4.4. Pitch and yaw output with randomised algorithm based probabilistic control of helicopter in Simulink .....	111
Figure 4.5. Reachability analysis for 2DoF helicopter system with LQR controller	113

Figure 4.6. Reachability analysis for 2DoF helicopter system with RABPC controller .....	114
Figure 4.7. Pitch and yaw output with randomised algorithm based probabilistic control of helicopter in real-time .....	115
Figure 4.8. Flow process of parameter estimation with simultaneous perturbation and stochastic approximation.....	119
Figure 4.9. Closed loop control of ball balancer system.....	126
Figure 4.10. Overall structure of ball balancer system with SPSA algorithm and estimation evaluation for optimal PID controller .....	127
Figure 4.11. Ball balancer output with SPSA-PID control in Simulink .....	128
Figure 4.12. Ball balancer output with SPSA-PID control in real-time .....	130
Figure 5.1. Output characteristics of helicopter system for normal and failure conditions.....	139
Figure 5.2. Parameter selection and decision boundary formation for training and testing process of the helicopter system fault classifier.....	141
Figure 5.3. Confusion matrix of various faults and trained fault classes of helicopter system classifier .....	141
Figure 5.4. Schematic diagram of neural network for neural integrated fuzzy control of helicopter system.....	143
Figure 5.5. Implementation of neural integrated fuzzy control with the helicopter system .....	144
Figure 5.6. General scheme of fault-tolerant controller for helicopter system.....	145
Figure 5.7. Pitch and yaw actuator response of helicopter during fault and after fault removal conditions in real-time.....	147
Figure 5.8. Three axial dynamics of ball balancer system representing the translational and rotational movements in XYZ plane .....	148
Figure 5.9. Different faults and operating states of ball balancer system.....	151
Figure 5.10. Feature space between peak and power features of ball balancer operating conditions.....	152
Figure 5.11. Confusion plot for prediction rate of the ball balancer fault classifier.....	153
Figure 5.12. Receiver operating characteristics – Area under the curve for ball balancer fault classifier .....	154

Figure 5.13. Discrete wavelet transform based reconfigurable fuzzy controller for ball balancer system.....	155
Figure 5.14. Response of ball balancer during fault and after fault removal conditions in real-time .....	158
Figure 6.1. Response of helicopter pitch and yaw motor operation for different trajectories.....	173
Figure 6.2. Pitch and yaw output with reinforcement learning-linear quadratic regulator control of helicopter in real-time for sine trajectory .....	174
Figure 6.3. Actor-Critic architecture.....	176
Figure 6.4. Reinforcement learning-neural network architecture for ball position and plate angle control of ball balancer system .....	180
Figure 6.5. Response of ball balancer for ball movement in different trajectories.	181
Figure 6.6. Ball balancer output with reinforcement learning-neural network controller in real-time .....	182

# List of Tables

Table 2.1. Time response analysis for neural integrated fuzzy control of helicopter in Simulink.....	51
Table 2.2. Root mean square error for neural integrated fuzzy control of helicopter in Simulink.....	52
Table 2.3. Time response analysis for neural integrated fuzzy control of helicopter in real-time .....	53
Table 2.4. Root mean square error for neural integrated fuzzy control of helicopter in real-time .....	54
Table 2.5. Time response analysis for neural integrated fuzzy control of ball balancer in Simulink .....	56
Table 2.6. Root mean square error for neural integrated fuzzy control of ball balancer in Simulink .....	56
Table 2.7. Time response analysis for neural integrated fuzzy control of ball balancer in real-time .....	57
Table 2.8. Root mean square error for neural integrated fuzzy control of ball balancer in real-time .....	58
Table 3.1. The computational complexity of transform analysis.....	60
Table 3.2. Root mean square error for wavelet fuzzy control of helicopter in Simulink.....	70
Table 3.3. Time response analysis for wavelet fuzzy control of helicopter in Simulink.....	70
Table 3.4. Root mean square error for wavelet fuzzy control of helicopter in real-time .....	72
Table 3.5. Time response analysis for wavelet fuzzy control of helicopter in real-time .....	72
Table 3.6. Time response analysis for wavelet fuzzy control of ball balancer in Simulink.....	74
Table 3.7. Root mean square error for wavelet fuzzy control of ball balancer in Simulink.....	74

Table 3.8. Time response analysis for wavelet fuzzy control of ball balancer in real-time .....	76
Table 3.9. Root mean square error for wavelet fuzzy control of ball balancer in real-time .....	76
Table 3.10. Root mean square error for evolving type 2 quantum fuzzy neural network control of helicopter in Simulink .....	94
Table 3.11. Time response analysis for evolving type 2 quantum fuzzy neural network control of helicopter in Simulink .....	94
Table 3.12. Root mean square error for evolving type 2 quantum fuzzy neural network control of helicopter in real-time .....	96
Table 3.13. Time response analysis for evolving type 2 quantum fuzzy neural network control of helicopter in real-time .....	96
Table 3.14. Time response analysis for evolving type 2 quantum fuzzy neural network control of ball balancer in Simulink .....	98
Table 3.15. Root mean square error for evolving type 2 quantum fuzzy neural network control of ball balancer in Simulink .....	99
Table 3.16. Time response analysis for evolving type 2 quantum fuzzy neural network control of ball balancer in real-time.....	100
Table 3.17. Root mean square error for evolving type 2 quantum fuzzy neural network control of ball balancer in real-time.....	100
Table 4.1. Root mean square error for randomised algorithm based probabilistic control of helicopter in Simulink .....	112
Table 4.2. Time response analysis for randomised algorithm based probabilistic control of helicopter in Simulink .....	112
Table 4.3. Root mean square error for randomised algorithm based probabilistic control of helicopter in real-time .....	116
Table 4.4. Time response analysis for randomised algorithm based probabilistic control of helicopter in real-time .....	116
Table 4.5. Time response analysis for SPSA-PID control of ball balancer in Simulink.....	129
Table 4.6. Time response analysis for SPSA-PID control of ball balancer in real-time .....	130
Table 5.1. Simulated conditions on helicopter system for generating data for developing fault classifier .....	137

Table 5.2. Comparison of classification parameters for helicopter system fault classifiers.....	142
Table 5.3. Parameters of NiF controller for helicopter system.....	145
Table 5.4. Sample data analysed in 2DoF ball balancer for developing the fault classifier .....	150
Table 5.5. Comparison of classification parameters for ball balancer fault classifiers.....	154
Table 5.6. Testing efficiency of the classifier for different faults in ball balancer system .....	156
Table 5.7. Time response analysis for reconfigurable control of ball balancer system .....	157
Table 5.8. Root mean square error for reconfigurable control of ball balancer system .....	157
Table 6.1. Root mean square error for reinforcement learning-linear quadratic regulator control of helicopter in real-time .....	175
Table 6.2. Time response analysis for reinforcement learning-linear quadratic regulator control of helicopter in real-time .....	175
Table 6.3. Time response analysis for reinforcement learning-neural network control of ball balancer in real-time.....	182

# Acronyms

Linear time invariant	LTI
Control Lyapunov function	CLF
Vertical take-off and landing	VTOL
Two degree of freedom	2DoF
Policy iteration	PI
Value iteration	VI
Twin rotor multi-input multi-output system	TRMS
Data acquisition	DAQ
Linear quadratic regulator	LQR
Linear quadratic gaussian	LQG
Proportional–integral–derivative	PID
Sliding mode controller	SMC
Adaptive second-order sliding mode controller	ASMC
Model predictive controller	MPC
Neural network	NN
Implicit function theorem	IFT
Mean value theorem	MVT
Single-input single-output	SISO
Fuzzy logic controller	FLC
Interval type 2 fuzzy neural networks	IT2FNN
Genetic algorithm	GA
Particle swarm optimization	PSO
Robust adaptive fuzzy controller	RAFC
Principal component analysis	PCA
Self-adaptive noise cancellation	SANC
Discrete-random separation	DRS
Generalized Kalman-Yanukovych-Popov lemma	GKYPL
Fractional-order sliding mode controller	FOSMC
Node decoupled extended Kalman filter	NDEKF
Shuffled frog leaping algorithm	SFLA



Gaussian particle swarm optimization	GPSO
Nonlinear model predictive control	NMPC
Differential evolution particle swarm optimization	DEPSO
Simultaneous perturbation stochastic approximation	SPSA
Adaptive disturbance rejection control	ADRC
Field-programmable gate array	FPGA
Wavelet transforms	WT
Discrete wavelet transforms	DWT
Minimal description length	MDL
Membership functions	MFs
Root mean square error	RMSE
Hardware in loop	HIL
Application programming interface	API
Evolving type-2 quantum fuzzy neural network	eT2QFNN
Quantum membership functions	QMF
Antecedent type-2 and consequent crisp number	A2-C0
Interval type-2 quantum membership function	IT2QMF
Footprint of uncertainties	FOU
Gaussian mixture model	GMM
Probably approximately correct	PAC
Randomized algorithm based probabilistic controller	RABPC
Support vector data descriptor	SVDD
Neural integrated fuzzy	NiF
Support vector machine	SVM
Area under the curve	AUC
Receiver operating characteristics	ROC
Reinforcement learning	RL

# Nomenclature

$\tau_1$	Helicopter main rotor torque
$\tau_2$	Helicopter tail rotor torque
$\psi$	Helicopter pitch angle
$\varphi$	Helicopter yaw angle
$u_1$ & $u_2$	Voltage inputs for operating pitch and yaw motors of helicopter
$a_1, a_2, b_1, b_2$	Static characteristic parameters of helicopter
$I_1$	Moment of inertia of helicopter main rotor
$I_2$	Moment of inertia of helicopter tail rotor
$M_g$	Gravity momentum of the helicopter
$B_{1\psi}$	Friction momentum function parameters for helicopter main rotor
$B_{1\varphi}$	Friction momentum function parameters for helicopter tail rotor
$T_{m1}, T_{m2}, T_{t1}, T_{t2}$	Denominator parameters for main and tail rotor of helicopter
$k_1$	Motor gain for helicopter main rotor
$k_2$	Motor gain for helicopter tail rotor
$k_c$	Cross-reaction momentum gain for helicopter
$\beta_l(s)$	Dynamics among the subsequent angle of the motor load gear in ball balancer system
$V_m(s)$	Motor input voltage for servo unit of ball balancer system
$A(s)$	Ball position on the ball balancer system
$W_{ss}(s)$	Ratio of dynamics of the ball position to the load gear angle
$M_{ball}$	Mass of the ball on the ball balancer system
$J_{ball}$	Moment of inertia of the ball on the ball balancer system
$R_{ball}$	Radius of the ball on the ball balancer system
$R_{arm}$	Distance between the servo motor and output gear shaft in the ball balancer system
$g$	Gravitational constant
$L_p$	Length of the plate in the ball balancer system
$K_m$	Motor speed constant for servo unit of ball balancer system
$K_g$	Static gain of the ball balancer system

$t_s$	Settling time for the characteristics of helicopter and ball balancer systems
$e_{ss}$	Steady-state error for the characteristics of helicopter and ball balancer systems
$t_p$	Peak time for the characteristics of helicopter and ball balancer systems
$M_p$	Peak overshoot for the characteristics of helicopter and ball balancer systems

# Chapter 1. INTRODUCTION

## 1.1 OVERVIEW

The approximation of underactuated, holonomic, and non-holonomic systems through automatic decision development and nonlinear control methods [1] is an issue that appears in many problems [2] and can be tackled through various approaches. The diversity and complexity of these systems have led researchers in the field to analyze the action of various controllers which are mainly focussed at achieving self-balancing control and steady-state operation. Conventionally, a series of scientific, industrial, and military applications motivated rigorous analysis and control design for these systems. On the other hand, theoretically challenging nature of analysis of the behavior of nonlinear dynamical systems attracted many mathematicians to study control systems. As a result, the efforts of engineers and scientists together led to creation of linear control, optimal control, adaptive control, and nonlinear control theories.

For the past few decades, aerospace and robotics applications remained as some of the most influential sources of motivation for rigorous analysis and control of nonlinear systems. All along the way, advances achieved by researchers in these areas has mutually affected and enhanced each other. In light of these advancements, this chapter identifies the various aspects of nonlinear control theory associated with the nonlinear systems.

## 1.2 BACKGROUND AND EXISTING CHALLENGES

In this section, an overview of the past research on classes of nonlinear control systems and mechanical systems that are relevant to this work are presented. In this thesis, our main focus in is on control design for nonlinear systems that arise from control of an important and broad class of mechanical benchmark systems. It is identified that, most of these mechanical control systems have fewer actuators (i.e., controls) than configuration variables making the system underactuated. One of the main contributions of this thesis is explicit transformation of high-order underactuated systems into cascade nonlinear systems by keeping the structural properties of the system intact. This transformation is performed using a global/semiglobal change of coordinates obtained from the Lagrangian of the system in closed-form. After applying this transformation, the control of the original high-order system is reduced to the control of lower-order nonlinear systems. This motivated the development of new control

design methods for both cascade nonlinear systems and different classes of nonlinear systems.

As a result, the main body of this thesis involves the following topics:

- Nonlinear control of cascade systems
- Dynamics, Reduction, and Control of underactuated systems

In the following sections, the state-of-the-art of research in each of the above topics is discussed.

### 1.2.1 *Nonlinear systems*

To describe the highly nonlinear systems, an evolution of control systems from linear systems to nonlinear systems is presented. Initially, the linear time-invariant control systems are expressed in the form

$$\dot{x} = Ax + Bu, \quad y = Cx \tag{1.1}$$

where  $x \in \mathbb{R}^n, y \in \mathbb{R}^m, u \in \mathbb{R}^p$ . Questions regarding controllability, observability, stabilization, and tracking for this system using state or output feedback has been quite well-understood for a long time. However, adding constraints or further specifications to the description of the system might make the control design for the system rather complicated. For example, if the system is controllable with bounded control inputs, is there any static or dynamic state feedback that asymptotically stabilizes the system? or what if  $A, B, C$  are known up to an uncertainty (i.e.  $A = A_0 + \Delta$  where  $A_0$  is known and  $\Delta$  is unknown but norm-bounded). None of these problems can be addressed as simple as the control problems for the original linear system (1.1). One can observe that minor deviations from the standard problem of stabilization of a linear time invariant (LTI) control system and additional constraints make the system rather complex. As a further step, consider the following linear system in feedback connection with a memoryless nonlinearity

$$\dot{x} = Ax + Bu, \quad y = \phi(Cx) \tag{1.2}$$

where  $\phi(0) = 0$  and  $\exists c_2 > c_1 \geq 0: c_1 z^2 \leq z \cdot \phi(z) \leq c_2 z^2, \forall z \in \mathbb{R}$ . These systems are considered slightly nonlinear in nature. Further, the control design and analysis for systems in the form (1.2) with different types of nonlinearities led to absolute stability theory. An input-output stability approach or a frequency domain analysis are the dominant tools in dealing with slightly nonlinear systems. In the literature, this notion of feedback interconnection of a linear system and a nonlinearity was generalized to feedback of an LTI system and an uncertainty (or operator) that has bounded gain. This led to the development of robust stability theory [3] and integral quadratic constraints [4]. Though, these methods are successful in dealing with linear

systems, uncertain linear systems, and slightly nonlinear systems, they are not applicable to truly nonlinear systems (i.e., systems with nonlinear time evolution of the state) that do not include any basic linear parts. To be more precise, a modification of (1.2) as

$$x = \sigma(Ax + Bu), \quad y = Cx \quad (1.3)$$

called a recurrent neural network with a saturation-type (i.e., sigmoidal) nonlinearity  $\sigma$ , has no fundamental similarities in terms of controllability and observability to an LTI control system or a slightly nonlinear system. In addition, the linearity of the output does not simplify analysis of the system due to the fact that time-evolution of the system follows a nonlinear law. Dynamic neural networks are examples of highly nonlinear systems and the conditions for controllability (only the discrete-time case) and observability of these systems were introduced in [5], [6]. The analysis method employed in the preceding work was a rather involved time-domain analysis. Obviously, due to the fact that a frequency domain analysis can only deal with systems that have linear time-evolution of the state.

A rather standard form for nonlinear systems affine in control in analogy to (1.1) is

$$\dot{x} = f(x) + g(x)u, \quad y = h(x) \quad (1.4)$$

where  $f, g, h$  are nonlinear smooth functions. By a highly nonlinear system, a system is defined as

$$\dot{x} = f(x, u), \quad y = h(x) \quad (1.5)$$

where  $f$  is a nonlinear function of  $(x, u)$  (regardless of linearity or nonlinearity of  $h(x)$ ) such that there exist' no invertible function  $(z, v) = (T_1(x), T_2(x, u))$  and matrices  $A, B$  satisfying  $\dot{z} = Az + Bv$ . A comprehensive local theory regarding controllability, observability, stabilization, tracking, and disturbance decoupling for nonlinear systems in (1.4) can be found in [7]. The main tools to address these control problems were differential-geometry and Lie theory that became very common in the literature. Though these methods were rather successful in local analysis of nonlinear systems affine in control they usually fail to work for a global analysis and nonlinear systems that are nonaffine in control. Moreover, lie algebraic conditions are not robust to uncertainties in  $f, g, h$ .

Further, the input-to-state stability theory [8], combines both absolute stability and robust stability theories in one for highly nonlinear systems in the general form (1.5). The main tools in this theory for robustness analysis to disturbances are control Lyapunov functions (CLF's). The problem is that in general it is not easy to construct CLF's for highly nonlinear systems. In many control applications, a global/semiglobal control design and analysis is required. In addition, after applying certain nonlinear coordinate transformation to the dynamics of

nonlinear systems affine in control, the transformed system or its subsystems could be nonlinear systems that are non-affine in control. This motivated us to consider global/semiglobal stabilization and analysis of highly nonlinear systems that arise from the study of underactuated mechanical systems and nonholonomic systems.

### 1.2.2 Cascade nonlinear systems

Cascade nonlinear systems arise in many control applications either naturally after some change of coordinates, or due to design of an output feedback or a dynamic state feedback. In general, they are in the following form

$$\begin{aligned}\dot{z} &= f(z, \xi) \\ \dot{\xi} &= g(\xi, u)\end{aligned}\tag{1.6}$$

The most well-known results for cascade systems are related to the nonlinear systems in strict feedback forms and feedforward forms. The backstepping procedure has proven to be successful for systems in strict feedback form [7], [9], [10]. Control of feedforward cascade nonlinear systems is also well-studied. The simplest example of a feedforward nonlinear system is a perturbed chain of integrators that can be stabilized using small nested saturations [11]. Then, more general classes of feedforward systems were either controlled using a nonlinear small-gain theorem [12] or construction of Lyapunov functions [13]. However, control of cascade nonlinear systems with nontriangular structures has proven to be rather problematic. This is due to a counterexample in [14] and existence of the peaking phenomenon [15]. Further, it turns out that if some coupling terms in the dynamics of cascade nonlinear systems satisfy appropriate growth conditions then it is possible to stabilize them using low-gain or high-gain feedback laws [16], [17].

### 1.2.3 Underactuated systems

Underactuated mechanical systems are systems that have fewer control inputs than configuration variables. Underactuated systems appear in a broad range of applications including Robotics, Aerospace Systems, Marine Systems, Flexible Systems, Mobile Systems, and Locomotive Systems. The "under actuation" property of underactuated systems is due to the following four reasons: i) dynamics of the system (e.g. aircraft, spacecraft, helicopters, underwater vehicles, locomotive systems without wheels), ii) by design for reduction of the cost or some practical purposes (e.g. satellites with two thrusters and flexible-link robots), iii) actuator failure (e.g. in a surface vessel or aircraft), iv) imposed artificially to create complex

low-order nonlinear systems for the purpose of gaining insight in control of high-order underactuated systems (e.g. the Acrobot, the Pendubot, the Beam-and-Ball system, the Cart-Pole system, the Rotating Pendulum, the TORA system).

The main control methods applied to examples of inverted-pendulum type underactuated systems is based on swing-up of the pendulum from its downward position and then switching to a balancing controller that is designed using a linearization technique or gain scheduling to balance the pendulum [18]. This includes swing up control using energy-based methods [19] for the Acrobot (i.e. a two-link planar robot with an actuator at the elbow) [20], the Pendubot (i.e. a two-link planar robot with an actuator at the shoulder) [21], the cart-pole system [22], a triple-link inverted pendulum [19], and the rotating pendulum [23]. The balancing controller for the Acrobot using spline functions can be found in [24]. Due to its complexity, the beam-and-ball has been the focus of study among researchers with diverse interests. These methods include approximate feedback linearization methods by Hauser et al. [25], small nested saturations for stabilization of feedforward cascade nonlinear systems by Teel [11], and stabilization by output feedback as discussed in [26].

Moreover, stabilization of the beam-and-ball by construction of Lyapunov functions is addressed in [16]. Besides, global stabilization of the beam-and-ball system is achieved in [17]. Passivity-based method is mostly used for swing-up control design of underactuated systems with inverted-pendulums [19]. Moreover, a passivity-based approach was employed for a special example of a pendulum on a cart that was transformed to a cascade form [27]. The main drawback of these passivity-based methods is their narrow range of applications. In fact, it is identified that, no applications of energy-based methods in control of real-life underactuated systems in robotics, aerospace, and marine applications is known.

The vertical take-off and landing (VTOL) aircraft is another example of an underactuated system that has been extensively used as a test-bed for different methods of trajectory tracking and configuration stabilization. This includes tracking for slightly non-minimum phase systems [28] and hybrid/switching based control methods [29]. Exponential stabilization of examples of underwater vehicles and surface vessels that are underactuated was achieved in [30] and [31] using appropriate coordinate transformations and analysis of a time-varying linear system. A similar type result for attitude control of an underactuated spacecraft is given in [32]. The role of second-order nonholonomic constraints in the necessity of the use of discontinuous stabilizing feedback laws for stabilization of underactuated systems is discussed in [33]. This is mainly based on the famous condition on stabilizability of nonlinear systems using time-invariant continuously differentiable state feedbacks [34]. In addition to this issue, accessibility



of classes of underactuated mechanical systems has been recently addressed in [35]. This is based on a framework applied to analysis of controllability of nonholonomic systems [36] and a controllability theorem [37]. An example of a discontinuous stabilizing feedback for a system with an internal unactuated degree of freedom is given in [38].

Adaptive control [39] and sliding mode control techniques [40] have been also applied to underactuated mechanical systems for rather limited applications. Flexible-link robots are an important class of underactuated systems that are appropriate for space applications due to their lightweight and fast execution of commands. The Euler-Bernoulli model for a flexible arm is an infinite dimensional system [41]. A truncated modal analysis can be used to obtain a finite dimensional state-space model for flexible robots [41], [42]. Trajectory tracking for flexible robots is rather complicated and common measurements like the angle of rotation or the position of the tip, respectively, lead to a poor performance and non-minimum phase zero dynamics. In [43], a non-collocated minimum-phase output is proposed based on an analysis of the initial infinite dimensional model and then a finite-order compensator is designed for trajectory tracking. Here, a nonlinear non-collocated minimum-phase output for a flexible one-link robot arm is obtained from a finite-order Euler-Lagrangian equations of the system. The method of controlled Lagrangians (i.e. applying a control input that preserves the Lagrangian structure of a mechanical system) has been applied to local stabilization of the cart-pole system and the rotating pendulum to an equilibrium manifold [44]. However, so far this method has been unable to stabilize the rotating pendulum or more general underactuated systems to an equilibrium point. In addition to more traditional methods, recently, hybrid and switching-based control methods are finding their way in control of underactuated mechanical systems [44] and bipedal locomotion of walking robots [19]. In conclusion, apart from linearization-based techniques, control of underactuated mechanical systems has been mainly along the line of stabilization. These include some special examples of cascade nonlinear systems and using energy-based methods combined with a supervisory-based switching control. The state-of-the-art of research in control of underactuated systems is currently very far from our goal to find control design methods for broad classes of high-order underactuated systems that are effective for robotics and aerospace applications.

### 1.3 MOTIVATION

The description of highly nonlinear systems can be presented as an evolution of control systems from linear systems to nonlinear systems. Here, the questions regarding controllability,

observability, stabilization, and tracking for this system using state or output feedback has been quite well-understood for a long time. However, adding constraints or further specifications to the description of the system might make the control design for the system rather complicated. These complications mainly deal with the controllability with bounded control inputs, asymptotical stabilization of the system with any static or dynamic state feedback, and uncertainties in system parameters. None of these problems can be addressed as simple as the control problems for the original linear system. One can observe that minor deviations from the standard problem of stabilization of a linear time invariant (LTI) control system and additional constraints make the system rather complex.

Further, the control design and analysis for systems with different types of nonlinearities led to absolute stability theory. An input- output stability approach or a frequency domain analysis are the dominant tools in dealing with slightly nonlinear systems. In the literature, this notion of feedback interconnection of a linear system and a nonlinearity was generalized to feedback of an LTI system and an uncertainty that has bounded gain. This led to the development of robust stability theory and integral quadratic constraints. Though, these methods are successful in dealing with linear systems, uncertain linear systems, and slightly nonlinear systems, they are not applicable to truly nonlinear systems that do not include any basic linear parts. In addition, the linearity of the output does not simplify analysis of the system due to the fact that time-evolution of the system follows a nonlinear law. Apart from the truly nonlinear systems, the cascade nonlinear systems arise in many control applications either naturally after some change of coordinates, or due to design of an output feedback or a dynamic state feedback. The most well-known results for cascade systems are related to nonlinear systems in triangular forms. Namely, nonlinear systems in strict feedback forms and feedforward forms. The backstepping procedure has proven to be successful for systems in strict feedback form. Besides, the control of feedforward cascade nonlinear systems is also well-studied. The simplest example of a feedforward nonlinear system is a perturbed chain of integrators that can be stabilized using small nested saturations. Then, more general classes of feedforward systems were either controlled using a nonlinear small-gain theorem or construction of Lyapunov functions. However, control of cascade nonlinear systems with nontriangular structures has proven to be rather problematic. This is due to the existence of the peaking phenomenon. Further, it turns out that if some coupling terms in the dynamics of cascade nonlinear systems satisfy appropriate growth conditions then it is possible to stabilize them using low-gain or high-gain feedback laws.

From the above observations it is identified that, apart from linearization-based techniques, control of underactuated mechanical systems has been mainly along the line of stabilization. These include some special examples of cascade nonlinear systems and using energy-based methods combined with a supervisory-based switching control. The state-of-the-art of research in control of nonlinear systems is currently very far from our goal to find control design methods for broad classes of that are effective for different robotic applications. These gaps can be highlighted as follows:

- In many control applications, the response of the controller to sudden and transient variations in the behaviour of the system are merely addressed.
- The collective effect of parametric, and random uncertainties on the sensitivity of the system operation is a major drawback in different linear and nonlinear controllers.
- The effect of component and system failures on the operation of the conventional control approaches is not addressed deeply in the previous works.
- The failure conditions in the nonlinear systems are mainly identified based on the threshold of different characteristics at the output of the system, and the effect of external disturbance, and classification of failures at various levels of the system is a major drawback.
- The conventional control approaches lack in learning the system behavior for various operations it carries. This resulted in frequent tuning of the control actions for the system whenever there is a change in system operation.

These drawbacks in the conventional control approaches especially while dealing with the nonlinear and underactuated systems motivated the development of different control strategies in this thesis. To discuss in brief, the wavelet fuzzy control and the evolving type 2 fuzzy control are developed to overcome the drawbacks of identifying sudden and transient variations in the system behaviour. These controllers learn the operating state of the system in an online mode and provides the corresponding control action from the precomputed and hypothetical control rules. Further, the probabilistic control algorithms are developed for handling the parametric and random uncertainties in the system operation. These algorithms are motivated at handling the uncertainties, improving the sensitivity of the system for various disturbances, and achieving efficient control action. Furthermore, the fault classification based reconfigurable control mechanisms are developed with a motivation to classify different failure conditions in a system and achieve suitable control action to compensate the failure mode. This

approach is developed in an offline mode and has advantages with accurate condition monitoring and fast identification of the faults. Finally, the reinforcement learning approaches are developed to learn the system behaviour while performing multiple tasks and adapt the necessary control action for any change in the system operation. All these control techniques are highly motivated at overcoming the drawbacks in the conventional control approaches mentioned earlier.

## 1.4 OBJECTIVES

The main focus in this research is on the control of two degree of freedom (2DoF) benchmark mechanical systems. This is motivated by broad applications of underactuated systems and theoretically challenging problems that they have to offer. The objectives of this research in achieving the control and stability of the benchmark systems is based on the defined motives for the research as follows:

- Mathematical modelling of two degree of freedom (2DoF) ball balancer and 2DoF helicopter system.
- Design, control and implementation of classical controlling techniques like proportional derivative, proportional integral derivative, and linear quadratic control, for 2DoF ball balancer and 2DoF helicopter system using Simulink.
- Design, control and implementation of intelligent controlling techniques like fuzzy inference system, and adaptive neuro fuzzy inference system for 2DoF ball balancer and 2DoF helicopter system using Simulink. Further, develop wavelet fuzzy control, and evolving type 2 quantum fuzzy neural network for ball balancer and helicopter system.
- Real-Time control of 2DoF ball balancer and helicopter system for defined intelligent and classical controller.
- Validation and performance analysis of all types of control such as classical and intelligent controller for 2DoF ball balancer and helicopter system. Model the machine learning and reinforcement learning technique to validate the performance of ball balancer and helicopter system in the presence of various faults and operating conditions.
- Formation of hybrid controllers like fuzzy-proportional integral derivative controller, linear quadratic regulator with fuzzy controller, neuro fuzzy controller, and implementation of optimization technique-based control for nonlinear systems. Design

of randomization and stochastic perturbation-based algorithms for uncertainties in nonlinear systems.

## 1.5 METHODOLOGY OF THE RESEARCH WORK

The main contribution in this research is to provide analytical tools that allows the translation and solves the tracking and balancing control problems of the benchmark robotic systems. To achieve this, initially, the mathematical modelling of both the systems are developed. Besides, the conventional control approaches associated with the modelling, operation, and control of these systems are analysed to formulate the problems and develop new control methods. Further, the following solutions are proposed to meet the control problem of a nonlinear, underactuated, and uncertain dynamics of the system:

### **Solution I: Optimized intelligent control for 2DoF systems**

This solution develops an intelligent approach for ball balancer position control, and unmanned helicopter position tracking using wavelet transform based fuzzy controller and evolving type2 quantum fuzzy neural network.

- *Wavelet Fuzzy Logic Controller*

The wavelet transform based fuzzy controller overcomes the drawbacks of transparent interpretation of choosing fuzzy rules with techniques available in the literature. Besides, the effect of parametric uncertainties on the operation of the controller and system are overcome by the signal processing abilities of the discrete wavelet transform. Further, the advantage of wavelet transforms while analysing the non-stationary signals provides an upper hand while developing a hybrid controller.

- *Evolving fuzzy based control system*

The evolving type-2 quantum fuzzy neural network is targeted at overcoming the drawbacks of conventional fuzzy systems and achieving path tracking with helicopter systems and position control for ball balancer system. The proposed approach overcomes the effects of parametric uncertainties in the benchmark nonlinear system with the help of a quantum rule growing scenario. Besides, the drawbacks of conventional control techniques with harmonic noises, and initializing weights for achieving trajectory tracking and position control are overcome by the automatic generation of the rules in a single pass learning mode.

### **Solution II: Probabilistic algorithms for control of 2DoF systems**

This solution provides a set of probabilistic control approaches with randomized algorithms and stochastic approximations for achieving trajectory tracking and position control respectively under parametric uncertainty conditions.

- *Randomized Algorithms*

The randomized algorithms are defined as an optimization technique where the actual system uncertainties initiate the randomization process while the deterministic decision parameters remain unconsidered. Usually, this randomness is generated in a system due to the involvement of stochastic uncertainties which produce various results for same input at different runs. This indicates a probabilistic property which defines the system as probably approximately correct (PAC). This approach is adapted to design a control algorithm with reference to randomization for gain matrix calculation and for closed-loop system analysis involving the unmanned helicopter for path tracking operation.

- *Simultaneous perturbation and stochastic approximation*

The simultaneous perturbation stochastic approximation (SPSA) algorithm is used for efficient estimation of unknown vectors based on small measurements for application with signal identification and adaptive control. Generally, the SPSA works on selected coefficients with two observations which generate estimations recursively in random directions at each iteration. Initially, the SPSA algorithms perturbs the current design parameter in random directions and measures the objective function of each observation. The measured observation is used to estimate the unknown vector which updates a new design parameter until the termination criterion is achieved. This provides an opportunity for achieving better convergence towards optimal solution for updating the design parameter. The motivation for using SPSA for adaptive control of a system with unknown but bounded disturbances is due to its easy to implement searching algorithm especially for the real-time control applications. Furthermore, the SPSA has advantages due to its iterative process which implies the idea of online learning with adaptability of new data and memory saving. The algorithm remains operational while accommodating the growing dimension of the estimated parameters and is resistant to arbitrary external noise at the point of input data. Besides, it exhibits less computation time due to small number of measurements and is capable of solving high dimensional optimization problems. The proposed approach is aimed at achieving a finite bound of residual between estimates and time-varying unknown parameters when observations are made under an unknown but bounded noise. This provides an intuitive tuning method for the PID controller, and achieves balancing

control for the closed loop ball balancer system by updating the adaptive parameters in real-time.

### **Solution III: Fault classification-based reconfigurable control for 2DoF Systems**

In this solution, a fault classification based reconfigurable control approach motivated at the operation of the benchmark systems by overcoming the failure modes in their components is developed. Initially, a fault classification-based controller capable of identifying the faults in the helicopter system is developed. This approach is based upon the previously trained data to achieve optimal performance by combining the classified output with an intelligent controller. The proposed approach adopts wavelet transforms and support vector data descriptor for efficient classification of the operating state of the helicopter. Further, the classified fault data is combined with the proposed neural integrated fuzzy controller to achieve the efficient operation of the helicopter in attitude and trajectory tracking.

Similarly, the variation in operating characteristics of the ball balancer system for different faults and operating conditions in time and frequency domain are identified with support vector machines. Initially, these characteristics are processed with the help of discrete wavelet transform to extract various features in the signal and train them with a supervised machine learning technique to develop a fault classifier. The developed fault classifier continuously monitors the operating state of the vehicle and provides necessary information to the controller to achieve steady state operation. The controller operation here is achieved with the help of a wavelet fuzzy controller.

### **Solution IV: Reinforcement learning with 2DoF systems**

This chapter establishes a useful baseline for trajectory tracking and balancing control with most fundamental problems in optimal control theory. The model free controllers are developed in linear and nonlinear environment using generalized learning algorithms, policy iteration (PI) and value iteration (VI). The PI and VI are implemented using various iterative and adaptive algorithms for solving the optimal control problem without a system model. To assess the performance of the developed model free controllers, real-time analysis is performed using benchmark control problems.

- *Learning algorithm-based linear model free control*

The learning algorithm learns the optimal function and control policy, based on the temporal difference error of value and policy iterations. The policy iterations are used while stabilizing the control policy and the value iterations are used without stabilizing the control policy. The

main advantage with policy and value iteration is that, they work forward in time and are mostly suitable for real-time control.

- *Learning algorithm-based nonlinear model free control*

To approximate the optimal value of nonlinear affine discrete time system, the neural network approach is adapted as a generalized actor critic architecture for learning the optimal policy and optimal learning function. Further, to achieve model free operation of the neural network, an additional network is added which provides the information of the systems inner dynamics to the neural network. This process is known as interleaved learning process.

## 1.6 CONCLUSION

This chapter summarizes some of the correlated issues regarding the control of a nonlinear systems. Though, the control of nonlinear system is very well studied, there is very little known about control of underactuated systems with higher order constraints and uncertainties. In fact, based on the background, existing challenges, and motivation, the control of general nonlinear is currently considered as a major open problem. In this chapter, problems associated with control of nonlinear systems in the presence of constraints, uncertainties, and failure modes are defined. Further, a brief overview of the solutions provided to overcome these drawbacks are identified. This thesis comprises the contributions of methodology on intelligent control, randomised approaches, fault classification based reconfigurable control, and reinforcement learning methods for the nonlinear systems.

## 1.7 OUTLINE

This thesis is outlined as follows:

**Chapter 1** (this chapter) introduces the control of nonlinear and underactuated systems, along with the basics of nonlinear control theory.

**Chapter 2** provides a detailed idea of problem formulation in benchmark nonlinear systems with 2DoF helicopter and 2DoF ball balancer system. Initially, the mathematical modelling of both the systems are developed. Further, a detailed literature on linear quadratic regulator (LQR), proportional integral derivative (PID), sliding mode controller (SMC), model predictive controller (MPC), H-infinity controller, neural network controller, fuzzy systems, randomized algorithms, fault tolerant controllers, and other hybrid algorithms are identified for



both the helicopter and ball balancer systems. Furthermore, the neural integrated fuzzy inference system (NiF) approach is developed to analyse the operation, and control of the benchmark robotic systems for formulating the problems and developing new control methods.

**Chapter 3** develops two optimal intelligent approaches for ball balancer position control, and unmanned helicopter position tracking using wavelet transform based fuzzy controller and evolving type2 quantum fuzzy neural network.

**Chapter 4** analyses the parametric uncertainties in the helicopter and ball balancer systems by developing a set of probabilistic control approaches with randomized algorithms and stochastic approximations.

**Chapter 5** develops a fault classification based reconfigurable control approach motivated at the operation of the benchmark systems by overcoming the failure modes in their components. The wavelet transform approach is used to achieve data preparation for developing the fault classifier using machine learning algorithms. Further, intelligent control techniques are developed to achieve the reconfigurable control.

**Chapter 6** establishes a useful baseline for trajectory tracking of helicopter system by reinforcement learning based linear quadratic control, and balancing control of ball balancer system with reinforcement learning based neural network control. Both the approaches are dependent on the reinforcement learning to train with different trajectories for tracking and position control.

**Chapter 7** provides the concluding remarks for all the approaches and identifies the state possible directions of future research.

## Chapter 2. LITERATURE REVIEW

The control of autonomous robots to perform complex tasks in dynamic environments has been a crucial area of control. Most of these control aspects are directly related to position control, path planning, trajectory tracking and balancing control of vehicles, etc. This chapter provides a summary of the most fundamental and recent advances on the theories and methodologies for achieving the control aspects of two benchmark robotic systems. Initially, the trajectory tracking and attitude control are identified on a two degree of freedom (2DoF) helicopter model (twin rotor multi-input multi-output system), and later the position tracking and balancing control is observed with a 2DoF ball balancer system. Conventionally, a detailed overview of different position tracking and balancing control strategies was provided in [45], [46]. These state-of-the-art solutions followed two main approaches. The first one is referred to as cross-coupling control [47]–[49], where the control loops of all machine axes are considered simultaneously, and the control objective is expressed as a contour-error regulation problem [50], [51]. In the second approach, called individual axis control, where each axis is handled separately. Here, the effects of the dynamics of other axes are treated as disturbances, and the control design reduces to that of individual axes [52]. Both the approaches relied on adjusting either the operating parameters (pitch and yaw movement, plate angle, etc.) [53], [54] or achieving position tracking [55], [56], such that the difference between measured and desired paths are sufficiently small and certain constraints are satisfied. Further, it is identified that both the approaches are dependent on each other while developing the control action. Moreover, while dealing with the individual axis control, apart from few exceptions (varying-gain, intelligent, and stochastic approach-based control that accounts for unknown noises, and fault-tolerant controllers), the majority of nonlinear control techniques for the 2DoF systems do not utilize parameter estimation in the system dynamics. This could be overcome by adapting the use of nonlinear and adaptive control algorithms in individual axis control architectures because the problem of nonlinear position control for flexible axis is well explored. Indeed, several relevant studies that employ techniques from nonlinear control theory, have been reported in the literature regarding the position and balancing control. A detailed overview of these techniques for achieving different control aspects is discussed in further sections.

## 2.1 MODELLING OF 2 DOF HELICOPTER

The helicopter test rig designed by Quanser [57] is used for replicating the behaviour of an unmanned helicopter in 2DoF motion. The main components of the test rig are actuator sensors, Q2-USB data acquisition (DAQ) card [58], and the power amplifier (VoltPAQ-X2) [59]. Two DC motors are used as yaw motor for actuating the back propeller and pitch motor for rotating the front propeller. The yaw motor is a Faulhaber series 2842 – 006C motor with 1.6 *ohms* terminal resistance, and 0.0109 *N.m/A* torque constant [58]. Similarly, the pitch motor is a Pittman model 9234 with 0.83 *ohms* electrical resistance and 0.0182 *N.m/A* torque constant [60]. These motors are operated at a rated voltage of 12V, 3A for pitch normal operation and 12V, 1A for yaw normal operation, and can be extended up to a peak voltage of  $\pm 24 V$ , 5 A for pitch and  $\pm 15 V$ , 3A for yaw without damaging the motor. Further, the pitch and yaw propellers are mounted to the aluminium propelled shield and the motor shaft. The pitch propeller is a Graupner 20 *cm* with a thrust force constant of 0.104 *N/V*, and the yaw propeller is a Graupner 15 *cm* with a thrust force constant of 0.43 *N/V* [57]. To measure the angular position of pitch and yaw angle, two encoders are used. The pitch encoder has a resolution of 4096 counts per revolution, whereas the yaw encoder has a resolution of 8192 counts per revolution when operated in quadrature mode. This gives the effective position resolution of pitch and yaw axis which is about 0.0879 *degrees* and 0.0439 *degrees* respectively. These pitch and yaw encoders are directly connected with the DAQ board to provide necessary position feedback to control the helicopter. Further, the controller developed is interconnected with the test rig by performing data exchange between the simulated models and the helicopter setup. This is achieved by building the C code through the MATLAB and Quanser software. The developed C code is dumped into the hardware in the loop-application programming interface available with the VoltPAQ-X2 through immediate I/O which helps in controlling the test rig. The experimental setup of the Quanser 2DoF helicopter system used in this research is shown in Figure 2.1.

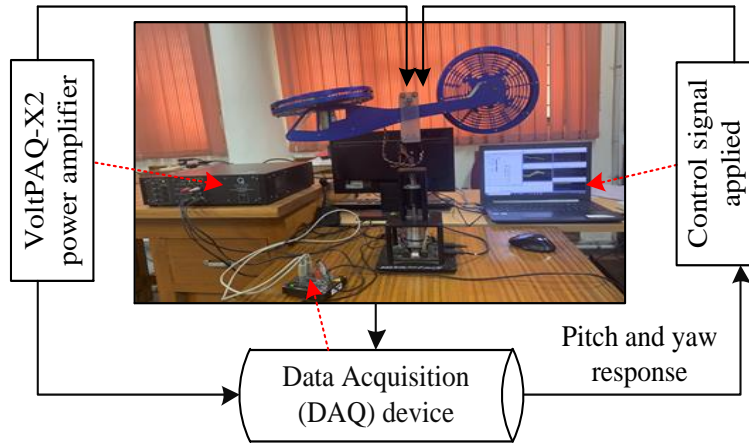


Figure 2.1. Experimental setup of a helicopter system [57]

### 2.1.1 Mathematical Modelling

To observe the operation and controllability, the main and tail rotors of the helicopter are modeled as shown in Figure 2.2.

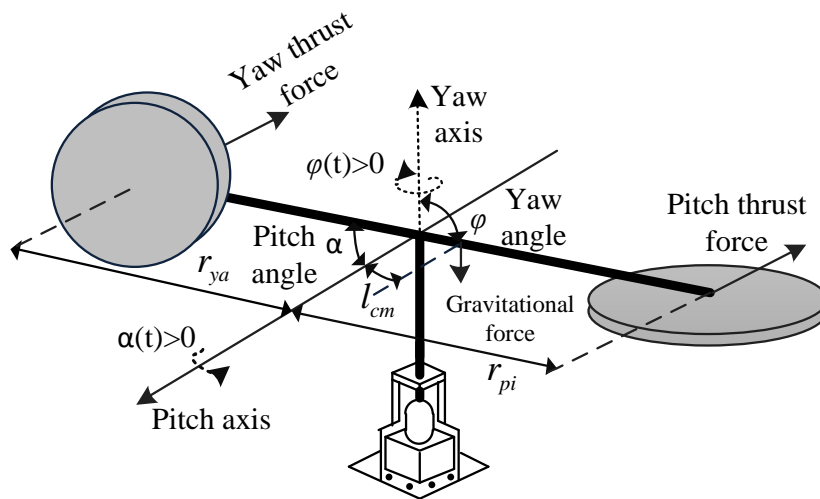


Figure 2.2. Free body diagram of 2DoF helicopter model with main and tail rotor [61]

The main rotor generates the vertical thrust, enabling the model to pitch around the horizontal  $X$  and  $Z$  axes, and the tail rotor generates horizontal thrust, enabling the model to yaw around the vertical  $Y$  axis. These twin-rotor DC motors are operated by voltage input which returns torque  $\tau_1$  for main rotor and torque  $\tau_2$  for the tail rotor. These rotors are operated by voltage inputs which return torque  $\tau_1$  for main rotor and torque  $\tau_2$  for the tail rotor. To demonstrate the couplings between various subsystems of the helicopter system, a block diagram is demonstrated in Figure 2.3. The dotted lines indicate the cross-coupling between the two planes and the outputs of the mechanics block depict the pitch  $\psi$  and yaw  $\phi$  angles for both the vertical

and horizontal planes, respectively. Further, the mathematical modeling of the setup can be explained based on DC motor modeling considering the main and tail rotors, and mechanics modeling considering the horizontal and vertical planes. Conventionally, many researchers have well established the mathematical modeling for pitch and yaw rotor of the helicopter system. Hence, this research mainly focused on understanding the operation of pitch and yaw motors and evaluating them to ensure the desired control objective.

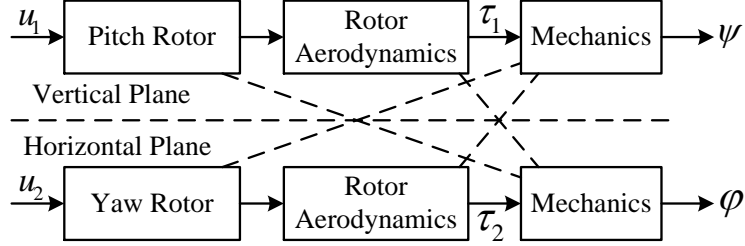


Figure 2.3. Block diagram of couplings in the twin-rotor helicopter system [62]

Since the twin-rotor, DC motors are operated by voltage inputs  $u_1$  and  $u_2$  which returns torque  $\tau_1$  and  $\tau_2$  for main and tail rotors respectively, the mechanical and electrical characteristics of the motors can be approximated [63]–[66] with the differential equations as follows:

A. For the main rotor and vertical movement:

$$\frac{d}{dt} \psi = \dot{\psi} \quad (2.1)$$

$$\begin{aligned} \frac{d}{dt} \dot{\psi} = & \frac{a_1}{I_1} \tau_1^2 + \frac{b_1}{I_1} \tau_1 - \frac{M_g}{I_1} \sin \psi + \frac{0.0326}{2I_1} \sin(2\psi) (\dot{\phi})^2 - \frac{B_{1\psi}}{I_1} \dot{\psi} - \frac{k_{gy}}{I_1} a_1 \cos(\psi) \dot{\phi} \tau_1^2 - \\ & \frac{k_{gy}}{I_1} b_1 \cos(\psi) \dot{\phi} \tau_1 \end{aligned} \quad (2.2)$$

$$\frac{d}{dt} \tau_1 = -\frac{T_{m1}}{T_{m2}} \tau_1 + \frac{k_1}{T_{m2}} u_1 \quad (2.3)$$

B. For tail rotor and horizontal movement:

$$\frac{d}{dt} \phi = \dot{\phi} \quad (2.4)$$

$$\frac{d}{dt} \dot{\phi} = \frac{a_2}{I_2} \tau_2^2 + \frac{b_2}{I_2} \tau_2 - \frac{B_{1\phi}}{I_2} \dot{\phi} - \frac{1.75}{I_2} k_c a_1 \tau_1^2 - \frac{1.75}{I_2} k_c b_1 \tau_1 \quad (2.5)$$

$$\frac{d}{dt} \tau_2 = -\frac{T_{t1}}{T_{t2}} \tau_2 + \frac{k_2}{T_{t2}} u_2 \quad (2.6)$$

Where,  $a_1, a_2, b_1, b_2$  are the static characteristics parameters,  $I_1$  and  $I_2$  are the moment of inertia of main and tail rotor,  $M_g$  is the gravity momentum,  $B_{1\psi}$  and  $B_{1\varphi}$  are friction momentum function parameters for main and tail motors,  $T_{m1}, T_{m2}, T_{t1}, T_{t2}$  are denominator parameters for main and tail motors,  $k_1$  and  $k_2$  are the motor gains for main and tail rotors respectively, and  $k_c$  is the cross-reaction momentum gain.

Now, the modeling of the helicopter can be expressed as a continuous-time linear system, which is given by:

$$\begin{aligned} x' &= Ax + Bu \\ y &= Cx + Du \end{aligned} \quad (2.7)$$

where  $x \in R^6 = [\psi \ \dot{\psi} \ \varphi \ \dot{\varphi} \ \tau_1 \ \tau_2]^T$  corresponds to the states,  $u \in R^2 = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$  is the control input, and  $y \in R^2 = \begin{bmatrix} \psi \\ \varphi \end{bmatrix}$  is the measured output.

Further, the matrices  $A, B, C$ , and  $D$  can be estimated by linearizing equations (2.1-2.6) as:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -\frac{M_g}{I_1} & -\frac{B_{1\psi}}{I_1} & 0 & 0 & \frac{b_1}{I_1} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\frac{B_{1\psi}}{I_2} & -1.75k_c \frac{b_1}{I_2} & \frac{b_2}{I_2} \\ 0 & 0 & 0 & 0 & -\frac{T_{m1}}{T_{m2}} & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{T_{t1}}{T_{t2}} \end{bmatrix}, B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{k_1}{T_{m2}} & 0 \\ 0 & \frac{k_2}{T_{t2}} \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}, D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

To determine the relationship between input and output variables, a relative gain analysis is performed [67] which reveals a low interaction between the inputs and outputs of the vertical and horizontal systems. Hence, the differential equations are approximated to the states involved to achieve the corresponding outputs.

### C. For Pitch Model:

Let  $u_2 = 0$ , then the pitch variables  $\psi, \dot{\psi}$ , and  $\tau_1$  can be represented into the state space as:

$$\frac{d}{dt}\psi = \dot{\psi} \quad (2.8)$$

$$\frac{d}{dt}\psi = \frac{a_1}{I_1}\tau_1^2 + \frac{b_1}{I_1}\tau_1 - \frac{Mg}{I_1}\sin\psi - \frac{B_1\psi}{I_1}\dot{\psi} \quad (2.9)$$

$$\frac{d}{dt}\tau_1 = -\frac{T_{m1}}{T_{m2}}\tau_1 + \frac{k_1}{T_{m2}}u_1 \quad (2.10)$$

$$x = \begin{bmatrix} \psi \\ \dot{\psi} \\ \tau_1 \end{bmatrix}, u = \begin{bmatrix} u_1 \\ 0 \end{bmatrix}, y = [\psi]$$

*D. For Yaw Model:*

Let  $u_1 = 0$ , then the yaw variables  $\varphi$ ,  $\dot{\varphi}$ , and  $\tau_2$  can be represented into the state space as:

$$\frac{d}{dt}\varphi = \dot{\varphi} \quad (2.11)$$

$$\frac{d}{dt}\dot{\varphi} = \frac{a_2}{I_2}\tau_2^2 + \frac{b_2}{I_2}\tau_2 - \frac{B_1\varphi}{I_2}\dot{\varphi} \quad (2.12)$$

$$\frac{d}{dt}\tau_2 = -\frac{T_{t1}}{T_{t2}}\tau_2 + \frac{k_2}{T_{t2}}u_2 \quad (2.13)$$

$$x = \begin{bmatrix} \varphi \\ \dot{\varphi} \\ \tau_2 \end{bmatrix}, u = \begin{bmatrix} 0 \\ u_2 \end{bmatrix}, y = [\varphi]$$

This representation of approximated state-space results in a linearized model and helps in formulating the control objective.

### 2.1.2 Problem Formulation

The trajectory tracking and attitude control are the main objectives while dealing with the closed-loop control of the 2DoF helicopter system. The closed-loop control of the pitch and yaw motor is developed as follows:

*A. Control of Pitch Subsystem:*

Considering the linearized pitch model in (2.8) -(2.10), the parameterized balance point concerning its input  $u_1 = U_1$  is indicated as:

$$\psi = \Psi(U_1) = \arcsin \left[ \frac{k_1}{MgT_{m1}} \left( \frac{a_1k_1}{T_{m1}} U_1^2 + b_1 U_1 \right) \right] \quad (2.14)$$

$$\dot{\psi} = 0 \quad (2.15)$$

$$\tau_1 = T_1(U_1) = \frac{k_1}{T_{m1}} \quad (2.16)$$

Hence, the parameters of the linearized pitch model are given by:

$$A_P = \begin{bmatrix} 0 & 1 & 0 \\ -\frac{M_g}{I_1} \cos\Psi(U_1) & -\frac{B_{1\psi}}{I_1} & \frac{2a_1}{I_1} T_1(U_1) + \frac{b_1}{I_1} \\ 0 & 0 & -\frac{T_{m1}}{T_{m2}} \end{bmatrix}, B_P = \begin{bmatrix} 0 \\ 0 \\ \frac{k_1}{T_{m2}} \end{bmatrix}, C_P = [1 \quad 0 \quad 0], D_P = 0 \quad (2.17)$$

$$\text{Let } -\frac{M_g}{I_1} \cos\Psi(U_1) = \alpha, -\frac{B_{1\psi}}{I_1} = \beta, \frac{2a_1}{I_1} T_1(U_1) + \frac{b_1}{I_1} = \gamma, -\frac{T_{m1}}{T_{m2}} = \theta, \text{ and } \frac{k_1}{T_{m2}} = \delta$$

Then, the model in (2.17) can be further expressed as:

$$G_{U_1}(s) = \frac{\delta\gamma}{s^3 - s^2(\beta + \theta) + s(\theta\beta - \alpha) + \theta\alpha} \quad (2.18)$$

### B. Control of Yaw Subsystem

Considering the linearized yaw model in (2.11) -(2.13), the parameterized balance point concerning its input  $u_2 = U_2$  is indicated as:

$$\varphi = \Phi(U_2) = 0 \quad (2.19)$$

$$\dot{\varphi} = 0 \quad (2.20)$$

$$\tau_2 = T_2(U_2) = \frac{k_2}{T_{20}} U_2 \quad (2.21)$$

Hence, the parameters of the linearized yaw model are given by:

$$A_Y = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{B_{1\varphi}}{I_2} & \left(\frac{2a_2}{I_2}\right) \left(\frac{k_2}{T_{t1}} U_2\right) + \frac{b_2}{I_2} \\ 0 & 0 & -\frac{T_{t1}}{T_{t2}} \end{bmatrix}, B_Y = \begin{bmatrix} 0 \\ 0 \\ \frac{k_2}{T_{t2}} \end{bmatrix}, C_Y = [1 \quad 0 \quad 0], D_Y = 0 \quad (2.22)$$

$$\text{Let } -\frac{B_{1\varphi}}{I_2} = \zeta, \left(\frac{2a_2}{I_2}\right) \left(\frac{k_2}{T_{t1}} U_2\right) + \frac{b_2}{I_2} = \eta, -\frac{T_{t1}}{T_{t2}} = \lambda, \text{ and } \frac{k_2}{T_{t2}} = \rho$$

Then, the model in (22) can be further expressed as:

$$G_{U_2}(s) = \frac{\rho\eta}{s^3 - s^2(\zeta + \lambda) + s(\zeta\lambda)} \quad (2.23)$$



The generalized closed-loop representation for pitch and yaw motor control in a 2DoF helicopter is shown in Figure 2.4.

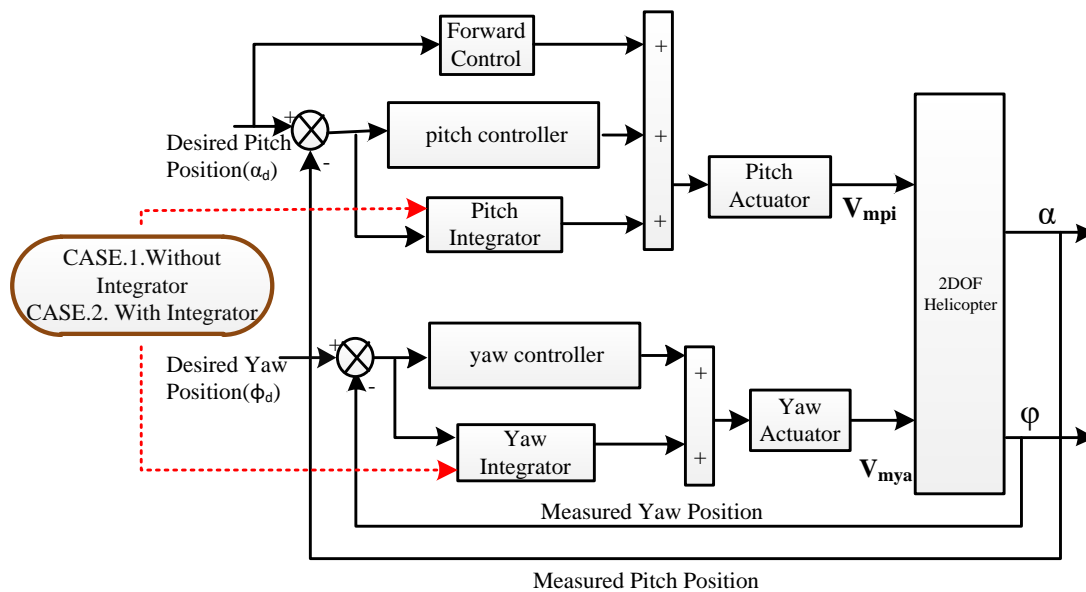


Figure 2.4. Generalized closed-loop control of pitch and yaw motors of a 2DoF helicopter System

## 2.2 REVIEW OF CONTROL TECHNIQUES FOR 2DOF HELICOPTER SYSTEM

Considering the above-discussed control problem, the first step towards achieving a stable and reliable operation of the helicopter is to develop a controller for following a predefined trajectory [68]. Various controllers developed in the literature for achieving these operations are discussed as follows:

*Linear Quadratic Regulator:* Linear quadratic regulator (LQR) is based on the receding horizon concept such that future outputs are predicted at every time step to minimize a global cost function [69]. For a helicopter system, the LQR control is developed linearization process [70] which provides a complete interpolation between a pair of operating points. But the limitations of the linearization model, when subjected to system stability, as it only guarantees the local asymptotic stability makes it incompetent for the system.

Further, LQR with an integrator has been considered to get optimum tracking of the helicopter [71]. This performed effectively to parametric uncertainties but limited robustness against unstructured uncertainty. Besides, in [72], and [73], the authors developed a linear quadratic gaussian (LQG) controller which results in a constant gain for a model with uncertainty,

making it robust. But the problem while allocating the covariance matrices, which are intuitive and directly depend on the weighting functions, before optimizing the gains makes it difficult to rely on the controller.

*Back-Stepping Controller:* The back-stepping approach presents a systematic method for designing a control system to stabilize a reference signal, by selecting an appropriate Lyapunov function. The design of the backstepping controller for helicopter systems [74] is progressed in the literature by developing a dual boundary conditional integral backstepping controller [75], and an integral backstepping controller [76]. This is achieved by deriving reference attitude angles to control the position, and by defining final inputs to the connect position and the attitude dynamics. Although backstepping control results in good dynamic performance, but its recursive design and repeated differentiation, limits its merits by increasing the system complexity and making it difficult to apply for multiple states controlling [77].

*Proportional–Integral–Derivative controller:* Proportional–Integral–Derivative (PID) controllers are implemented primarily to hold measured process value at a setpoint, or desired value. This is achieved by tuning the gains of PID controllers for adjusting the reactions to the setpoint changes and unmeasured disturbances such that the variability of control error is minimized. These tuning objectives are often conflicting, and engineers must balance the importance of the objectives based on cost, i.e. solve an optimization problem. In the control of helicopter systems, rotor controlling is achieved by auto-tuning of PID based on fractional-order reference model [78]. Further, a cross-coupled PID is designed for tracking purposes [79]. It is observed that the PID control of high order systems however often leads to oscillations resulting in slower settling times [80]–[82]. Apart from the above, the attitude stabilization of quadrotor UAVs has been done by the PID controller [83]. The PID is enough to stabilize the aero vehicle but the limitation lies in the need of converting the multi-rotor system into to linearized model and the hypothesis regarding linearizing point becomes necessary in this situation. The PID is restricted to control around a fixed set point and unable to work efficiently when the system is highly nonlinear.

*Sliding Mode Controllers (SMCs):* Sliding-mode methods were first introduced in [84] and were elaborated through specific controller and observer architectures in [85]. The basic principle in designs with sliding modes is the use of high-theoretically infinite-frequency switching terms in the control signals. The resulting control laws are discontinuous and apart from robust disturbance rejection, they can provide finite-time stabilization and tracking of a

reference signal, under certain disturbance boundedness conditions. Conventionally, the sliding mode control has been widely used with exceptional advantages such as invariance, precise path following, negligible vibrations, good precision, and excellent robustness even in the presence of variable load conditions. In [86], the authors developed a sliding mode control for trajectory tracking of UAVs. Besides, various SMC controlling techniques have been proposed for TRMS such as disturbance observer-based controller [87], [88], adaptive second-order sliding mode controller (ASMC) [89], [90], multivariable integral sliding mode [64] and terminal sliding mode controller [91], [92]. It is identified that terminal SMC provides excellence in reducing the turbulence on the other side of ASMC [87], [89] robust to disturbances but with satisfactory tracking capabilities and indulgence of the integral term in SMC improves trajectory tracking but stayed with residual error [64]. In another approach, parametric uncertainties and model error are dealt with sliding mode controller (SMC) for position control of 3 DoF helicopter [93] and gives asymptotic stability only. But when the SMC is combined with a backstepping controller, the fast convergence and global stability can be achieved simultaneously in the presence of uncertainties [94]. Apart from the above, basic sliding mode control (SMC) techniques were adapted for fault diagnosis, isolation, and control [95]. Further, the SMC has been advanced with a composite nonlinear feedback method where external disturbances cause the actuator saturation of helicopter [96].

From the above literature, the chattering problem of SMCs is a considered as a harmful phenomenon as it leads to low control accuracy, high wear of moving mechanical parts, and high heat losses in power circuits [97]. This is the most common phenomenon in SMCs, and it is difficult to attenuate and becomes unavoidable when subjected to discontinuous control.

*Model Predictive Control:* Model Predictive Controller (MPC) makes use of a process model to obtain the control action by minimizing an objective function. This is based on the explicit use of a system model to predict the process output at a future time instant (horizon). Further, it calculates the control sequence minimizing an objective function by adapting a receding strategy. Here, at each instant, the horizon is displaced towards the future, which involves the application of the first control signal of the sequence at each step. The development of MPC for helicopter systems is discussed in [98], [99]. In [28], constrained output feedback based MPC approach is developed for TRMS by considering the state variables observed using an unscented Kalman filter. The approach adopted a nonlinear dynamic model of the system, which is developed by considering all possible effective elements of the system. Further, the

elements were adaptively linearized along the prediction horizon using a state-dependent state-space representation. In other work, a model predictive control (MPC) uses online optimization within a predicted framework while taking advantage of the receding horizon with soft constraint [100] to provide an exact path tracking problem. In [29], a stable model predictive control approach is developed for constrained highly nonlinear TRMS. This research also adapted the same modeling technique mentioned in [28]. Besides, the MPC is developed for control of various helicopter applications in [101]–[103]. From the literature, it is identified that the disadvantage of MPC lies in its complex algorithm that needs a longer time than the other controller. Another major disadvantage of MPC is the model updating scheme which is dependent on the computational cost if the model is updated too frequently. To alleviate this disadvantage, one may try to decrease the model updating frequency, but this will deteriorate the system's stability rapidly [104].

*H-infinity Controller:* The H-infinity ( $H_\infty$ ) control method is adapted in attitude control systems to reduce oscillations and regulate the rotational moments in the presence of unmodeled dynamics and parametric uncertainties [105]. This controller requires all the bounded functions of the system that are analytic in the right-half complex plane to define the weighting function. Further, the weighting functions are varied concerning plant real-time performance during attitude control [106] to achieve the desired and accurate model/state information of the system. In [107], the  $H_2/H_\infty$  control is developed for the purpose of balancing and trajectory tracking of UAVs in the presence of disturbances and with very less response time. In [108], the H-infinity observer is developed to protect the TRMS from partial and complete failure that arises due to actuator and sensor faults. But the inability of the  $H_2/H_\infty$  control technique to define weighting functions and its dependency on the trial-and-error method makes it an unpredictable control procedure.

*Neural Network Controllers:* The use of neural networks for solving highly nonlinear control problems is widely seen in the literature [109]–[111]. The operation of neural networks involves many simple and similar processing elements at its inputs and outputs of its structure. The processing elements have internal parameters called weights which help in altering the behaviour of the whole network to achieve optimal output for the controller and its plant. Considering this advantage, various neural network-based control and estimation approaches for TRMS are discussed in [112], [113]. In [114], nonlinear flight control of the 2DoF helicopter using NN is developed using a backpropagation, feed-forward NN model. In [115],

an indirect adaptive neural network framework is chosen for developing an adaptive neural network controller for a simulated helicopter system. The controller is developed based on three interconnected neural networks called the observer, actor, and critic. The actor and critic networks rely on the observer network responsible for state estimation. In [116], a robust adaptive neural network (NN) control for helicopter systems using the implicit function theorem (IFT) and the mean value theorem (MVT) is developed for handling nonlinear nonaffine systems. This paper focused on single-input single-output (SISO) helicopter systems, which are exemplified by certain single-channel modes of operation, such as vertical flight and pitch regulation, and also by special conditions under which the multiple channels become decoupled.

*Fuzzy Systems:* Fuzzy logic has evolved as one of the emerging information processing technologies, especially from the last few years. Rapid growth has been witnessed in the number and variety of applications of fuzzy logic. In [117], the input uncertainties of quadrotor UAVs are dealt using a fuzzy controller which minimizes the system error based on the inputs and the antecedent fuzzy sets of non-singleton fuzzy logic controllers (FLCs). This identified that the fuzzy logic controllers have robust performance due to the knowledge-based design. Besides, the mathematical model of the plant is not required and the controller can directly deal with all possible uncertainties [118]–[120]. In [121], and [122] the intelligent adaptive fuzzy controllers are developed to track the output of the multi-input multi-output system by approximating the error between actuator and sensor measurements during external disturbances. Further, the type 1 FLCs (T1FLC) are adapted in various fields, but their inability to perform while dealing with higher-order uncertainties is considered as a major drawback [123], [124]. In another kind, the type 2 FLCs (T2FLC) are more powerful and can cope up with higher-order uncertainties. Also, the increasing fuzzy sets have improved the degree of freedom for T2FLC to handle uncertainties conveniently [125], [126]. But the type reduction scenario made the general T2FLCs computationally intensive [127]–[129].

Moreover, the advancements in artificial neural networks have widely improved the abilities of T2FLC due to their inherent learning ability and uniform approximation of nonlinear systems [130]–[132]. This resulted in the development of interval type 2 fuzzy neural networks (IT2FNN). The evolution of IT2FNN has been widely adopted in the field of control especially for path tracking problems in UAVs [133], [134]. But their drawbacks while calculating the correlation matrix burdens the computation process [135]. Besides, the drawbacks of

conventional membership functions and inadequacy to adapt to change in inputs raised the demand for innovation in the field of computation intelligence [136].

*Hybrid Controllers:* Besides the different linear, nonlinear, and intelligent control strategies for different systems, the hybridization of controllers is widely discussed in the literature [137]–[139]. These techniques combined the advantages of two or more controllers to overcome the individual drawbacks of each controller. The use of optimization techniques like a genetic algorithm (GA) and particle swarm optimization (PSO) algorithms along with the non-linear controllers has been depicted in [140] and [141]. GA based PID has been designed to modify control parameters with performance index as a fitness function [80] [142] [143] and an augmented control scheme comprising robust PID-based deadbeat control [81]. Here the control parameters of PID are modified to avoid the tuning problem and give a precise value of control gain [143]. In [144], an assessment of various conventional control methods and intelligent control methods dependent on fuzzy logic and GA were implemented on the TRMS. Further in [145] SMC, hybridization through fuzzy has been carried out and compared with fuzzy sliding and fuzzy integral sliding controller. In [146], the fuzzy LQR controller is distributed parallelly for individual axis control of the system. Similarly, the gradient descent algorithm has been used in designing of the robust adaptive fuzzy controller (RAFC) [147]. The Lyapunov theory-based stability analysis for both the cases in [146] and [147] has identified that the LQR integration of fuzzy gives an asymptotical response to the system output while RAFC guaranteed complete close loop stability for TRMS. Further, Huaman et.al [148] explained the elimination of the chattering from the SMC when augmented by a fuzzy controller while achieving trajectory tracking for a quadrotor. Besides, the fuzzy-sliding hybrid controller[149], adaptive sliding [150], LMI based observer[151], the fuzzy controller using non-monotonic Lyapunov function [152] have been designed in the existence of disturbances and uncertainty in the helicopter to improve the performance.

*Randomized Algorithms:* Advanced control techniques which depended upon randomization and probabilistic control are implemented in the literature to deal with parametric uncertainties and achieve the stability and performance requirements of nonlinear systems. For that matter, literature in [153]–[156] has been considered. Initially, these methods assumed that the parametric uncertainty affects the probabilistic nature of the system, and then a performance level has been provided to check its ability. Further, the mathematical model required to design these algorithm-based controls is obtained by numerical linearization of the full order nonlinear

system [157]. Since the explicit relationship between the state-space matrices and the uncertain parameters should be available all the time, the linearization process subjected to repetition at the time of variation in the uncertain parameters. These randomization-based control algorithms take less computational time and easy to implement [153] and at the same time robust boundary conditions are less conservative but at the cost of probability risk failure. This approach is not limited to control engineering but also explores the general engineering design, robust optimization where the environment directly affected by the uncertainty. Other than the analysis of the system, the probabilistic methodology reveals its maximum capacity concerning control frameworks.

Besides, the controllers discussed above, many different controllers are adapted in the literature for modeling, estimation, and achieving tracking and balancing control for nonlinear systems. A quasi-linear parametric varying for modeling, identification, and control of the TRMS system is developed in [158]. In [159]–[161], the disturbance rejection controllers were developed for achieving turbulence compensation in small scale helicopters. In [162], various data-driven control techniques based on model-free adaptive control, model-free control, and virtual reference feedback tuning are developed for twin rotor aerodynamic systems. In [163], an adaptive controller is developed for asymptotical trajectory tracking with the help of gain scheduling techniques [164]. Lyapunov’s direct method [165], and Lyapunov’s function-based control [166] are developed for achieving trajectory tracking for helicopter systems. These techniques are capable of tracking path using image-based visual servo and integral barrier. This method ensures that the closed-loop stability is achieved without solving an ordinary differential equation. In [167] Tugrul et al. proposed a variance constrained method for defining the active and passive rotor morphing for helicopter blade motion control. Here, the optimization problem minimizes the control energy of the system. As the blade modeling is considered with the flight dynamics, it is quite complex and time-consuming to define the exact bound variance. The major problem with the output variance constrained control is the variance inequality.

*Fault-based control:* Apart from tracking and balancing control, and dealing with parametric uncertainties in the system, the fault-tolerant and fault detection-based control techniques are also available in the literature. These techniques are generally classified depending on their passive or active responses by fixed or reorganizable control methods [168]. The passive techniques are grounded exclusively on the use of robust and steady control, where possible

faults are treated as unknown signals (external disturbances) acting on the system dynamics. This can be identified with the idea of reliable control [169]. Generally, passive techniques do not require either on-line fault statistics from the fault-diagnosis technique or control reconfiguration [170], [171]. Several passive fault recognition and control techniques have been established based on robust control theories, e.g., multi-objective optimization, quantitative feedback theory method, H-infinity optimization, absolute stability theory, nonlinear regulation theory, etc. More details regarding these techniques can be obtained from [172], [173]. Since a passive technique system uses a controller designed off-line, based on the certain knowledge of the faults, it can handle very limited fault scenarios.

Unlike passive systems, active techniques provide a system with fault-tolerant capability by equipping it with a mechanism to diagnose the faults. This helps the controller to choose the essential corrective action to uphold acceptable post-fault closed-loop performance. In the absence of faults, a baseline controller is used to ensure good stability and tracking performances. They also make usage of the supervision level data and reconfigure the controller to achieve the mandatory corrective actions. Compared with passive techniques, active techniques are applicable for a broader range of areas and thus has been a major concern [173][174]. Recently, several interesting fault detection and control approaches have been developed and verified using the sliding mode method for on-line control allocation to achieve robust performance [175], [176]. The virtual actuator approach is studied extensively for linear time-invariant systems [177]. It has been extended to a range of the nonlinear system, e.g., linear parameter varying systems [178], T-S fuzzy systems [179], piecewise affine systems [180], and Lipschitz nonlinear systems [181]. Many works have been published on fault estimation-based fault-tolerant control systems through state observers [90], unknown input observers [182], moving horizon estimation [183], and usage of quantized measurements for sliding mode observer synthesis of Markovian jump systems [184]. Fault-tolerant control is applied using linear parameter-varying models to adjust the plant failure to the minimal control rather than self-repair according to the fault that occurred [185].

The capability of a system to tolerate faults through control design, using either passive or active approaches, is essentially a structural property of the system itself. A system with insufficient redundancy cannot be made successfully tolerant to fault irrespective of the control approach used. Therefore, a tool is needed to check the fault-tolerance capability before designing the controller. The concept of control reconfigurability was first developed by [186]



for linear time-invariant systems. Later, the concept of coverage of fault-tolerant control was developed in [187] for analyzing the reliability of the controller. These two concepts have been further developed and used by many researchers as analysis and design tools for fault-tolerant control systems. The concept of control reconfigurability has been extended to bilinear systems by [188] and switched systems by [189]. Huang *et al.* [190] proposed a fault-tolerant placement strategy for phasor measurement units based on control reconfigurability. Yang *et al.* [191] studied the fault recoverability and fault-tolerant control for interconnected nonlinear systems. A comprehensive review of fault detection methods can be found in [192], [193]. Smith *et al.* examined the vibration data of the helicopter components and classified the faults accordingly using principal component analysis (PCA) [194]. The drawbacks with PCA is that it is difficult to evaluate the covariance matrix accurately and it also fails to capture the simplest invariance unless the information is explicitly provided to the training data [195]. Besides, the dimensionality reduction of PCA fails to reproduce the original behavior of the signal during the testing process due to the loss of information. Camerini *et al.* developed a vibration-based automated procedure using support vector data description to detect bearing and drive train faults of a helicopter [196]. The research adopted a one-class classification, which is solely developed for monitoring a particular fault in the system. The adapted algorithm generally requires a large number of samples and is resistant to training data. It also fails to represent the complete density distribution of the training and testing data. Further work for helicopter gearbox monitoring has been done using analog neural networks [197] and signal processing techniques including self-adaptive noise cancellation (SANC), discrete-random separation (DRS), cepstrum editing, kurtogram, envelope analysis and iterative envelope cancellation [198]. A detailed survey of fault tolerant control and fault detection, isolation and recovery for helicopter system during faults and various methods to increase the safety and reliability of the system are discussed in [199]. The above discussed fault classification, identification and detection techniques has major disadvantages with learning and modelling nonlinear complex systems which is very important for developing a fault classification algorithm. In addition, their demerits due to restrictions on input variables, heteroscedasticity, and low efficiency during testing process, identified the need to develop a new classification approach for the fault classification of helicopter systems.

Besides, the performance of the helicopter is also affected when a fault occurs during real-time operation and makes the system unstable. So, to provide consistency, and achieve operational safety for the helicopter operation, localization of fault and development of fault-tolerant

controllers became an essential and structural property [200]. Further, to control the failures in higher-order flight control actuators, a decentralized fault-tolerant control system has been designed in [201]. The partial and complete failure that arises from the actuator and sensor fault is assisted by the H-infinity observer in [202]. Besides, with the advancement of sensor technology, the measurement the fault detection and isolation became more feasible [203]. But the action of external disturbances created huge errors from the measurements of sensor and actuator output and remains as a major concern. Some of these concerns were approximated by intelligent adaptive controllers [204]. A comprehensive review of fault detection methods can be found in [193].

Apart from the widely available classifiers in the literature, the true structure of the fault data is observed to be hard for developing a complete model [205]. This resulted in ignoring the delicate details of the data. To overcome this and improve classification performance, a combination of classifiers is suggested by many researchers [206]. But these methods proved to be costly and lacked redundancy while dealing with delicate data and unknown faults.

### 2.3 MODELLING OF 2DOF BALL BALANCER SYSTEMS

The 2DoF ball balancer system comprises of a quadrangular metallic plate, fixed at the center through a gimble joint. The gimble joint has a two degree of rotational freedom to tilt in 2-dimensional direction, x, and y-axis. A Faulhaber DC micromotor series 2338 motor [207] along with a potentiometer and tachometer is used for balancing the system in both the directions. The main objective of the system is to balance the ball without falling off the quadrangular metallic plate. The experimental setup of the 2DoF ball balancer as per the specifications developed by Quanser [208] is depicted by Figure 2.5. To explain the working of the setup, a workflow diagram is depicted in Figure 2.6. To begin with, a control signal related to the disposition of the plate is extracted and sustained to the data acquisition (DAQ) – Q2USB [208] board to observe the movement of the ball concerning voltage. As soon as the signal is attained, it is sent to the controller, and its coordinates are captured through the camera and combined, and the output is forwarded to amplify the power using amplifier through the DAQ for further alteration. Finally, the output is processed to the hardware.

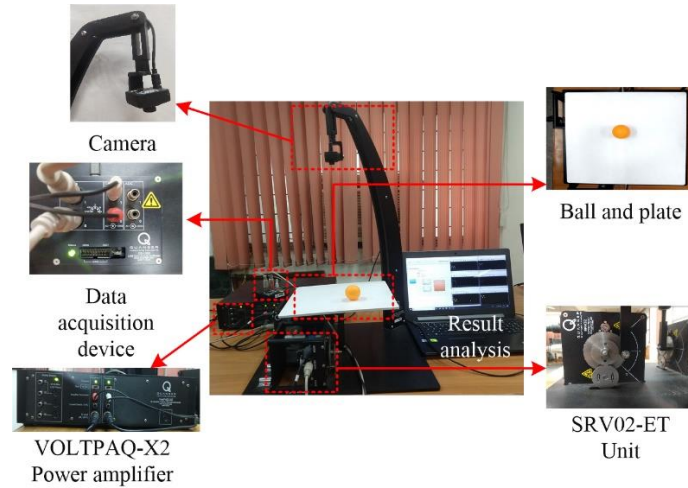


Figure 2.5. Laboratory setup of ball balancer system [208]

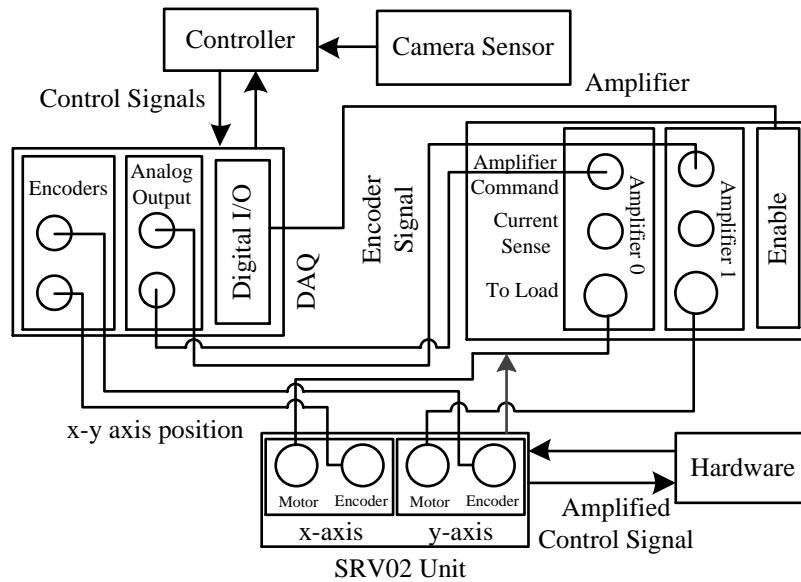
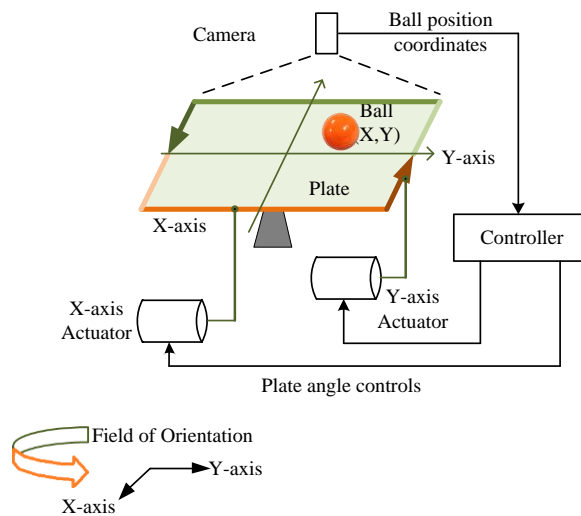


Figure 2.6. Typical wiring diagram between different components of Quanser 2DoF ball balancer system [208]

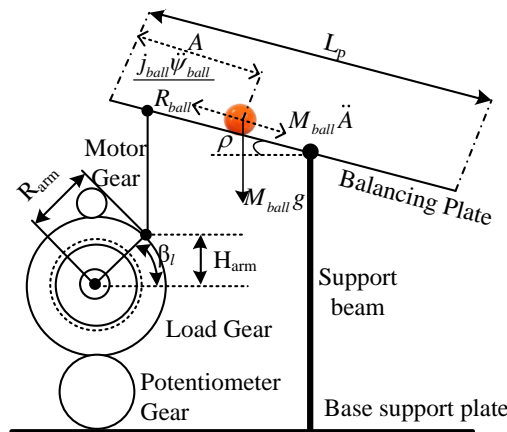
### 2.3.1 Mathematical Modelling of Ball Balancer Setup

The typical representation of a ball and plate system is given in Figure 2.7(a). This is an underactuated system with access to four degrees of freedom (DoF) controlled by two actuators. Hence its operation is referred to as a 2DoF ball balancer. The other parts of the system consist of a digital camera that captures the images of X and Y coordinates for the ball movement on the plate. Further, a vision algorithm computes and reads the ball coordinates from the image and provides information to the controller [209] for adjusting the angle of the plate through the X and Y servo motors to make the ball track the time-varying reference. As the operation of the 2DoF ball balancer is dependent on the control of two rotary servo base

units and the arrangement of the plate is symmetrical to these servo devices, it is anticipated that both the device dynamics are same. Hence the 2DoF ball balancer can be modeled as two decoupled ball and beam systems [210]. The dynamics among the subsequent angle of the motor load gear  $\beta_l(s)$  and motor input voltage  $V_m(s)$  of SRV02-ET base unit are depicted by the transfer function  $W_s(s)$ . The ratio of dynamics of the ball position  $A(s)$  to the load gear angle  $\beta_l(s)$  is depicted with  $W_{ss}(s)$ . Since the ball balancer coordinates have similar servo dynamics, the modelling of control through X direction is presented in this research. The X-axis control of ball and plate system is presented in Figure 2.7(b).



(a) A typical representation of ball and plate system



(b) X-axis control of ball and plate system

Figure 2.7. Schematic of 2DoF ball balancer system [211]

The nonlinear motion of the 2DoF ball balancer is obtained from Qunaser 2DoF ball balancer model [66] as follows:

$$\ddot{A}(t) = \frac{2M_{ball}g\beta_l R_{arm}R_{ball}^2}{L_p(M_{ball}R_{ball}^2 + j_{ball})} \quad (2.24)$$

$$\ddot{\beta}_l(t) = \frac{(K_m u(t) - \dot{\beta}(t))}{\tau} \quad (2.25)$$

where  $M_{ball}$  is the mass of the ball,  $j_{ball}$  is the moment of inertia of the ball,  $R_{ball}$  is the radius of the ball,  $R_{arm}$  is the distance between the servo motor and output gear shaft,  $g$  is gravitational constant,  $L_p$  is the length of the plate,  $K_m$  is the motor speed constant and  $\tau$  is the time constant of the motor. Detailed modeling of the 2DoF ball balancer system is discussed by the authors in their previous work [212].

For an input  $\beta_l$  and output  $A$ , the transfer function for ball position control can be formulated as:

$$W_{ss}(s) = \frac{A(s)}{\beta_l(s)} = \frac{K_b}{s^2} \quad (2.26)$$

$$\text{where, } K_b = \frac{2M_{ball}g\beta_l R_{arm}R_{ball}^2}{L_p(M_{ball}R_{ball}^2 + j_{ball})}$$

Further, the transfer function of the servo motor controlling the plate angle is given by:

$$W_s(s) = \frac{\beta_l(s)}{V_m(s)} = \frac{K_g}{s(\tau s + 1)} \quad (2.27)$$

where,  $K_g$  is the static gain.

The overall transfer function of the cascaded connection between the servo motor and ball balancer module is given by:

$$W(s) = W_{ss}(s)W_s(s) = \frac{\beta_l(s)}{V_m(s)} = \frac{K_g}{s^3(\tau s + 1)} \quad (2.28)$$

The model parameters,  $K_g$  and  $\tau$ , of the servo unit under no load and in the high-gear configuration are  $K = 1.53 \text{ rad}/(s)$  and  $\tau = 0.0248 \text{ s}$ .

Thus, representing the system in a state-space variable is given by:

$$\begin{bmatrix} \dot{A}(t) \\ \ddot{A}(t) \\ \dot{\beta}_l(t) \\ \ddot{\beta}_l(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & K_b & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{1}{\tau} \end{bmatrix} \begin{bmatrix} A(t) \\ \dot{A}(t) \\ \beta_l(t) \\ \dot{\beta}_l(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{K_m}{\tau} \end{bmatrix} u(t) \quad (2.29)$$

### 2.3.2 Problem Formulation

The open-loop block diagram of the 2DoF ball balancer is represented as a decoupled model where x-axis servo doesn't affect the response of the y-axis as depicted in Figure 2.8.

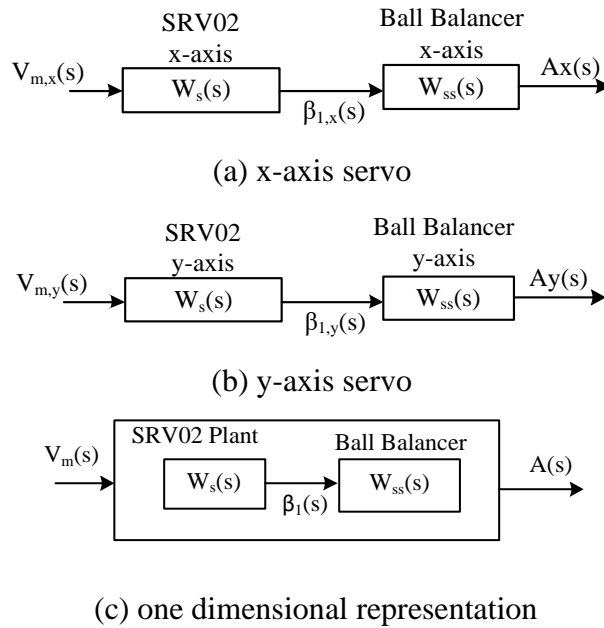


Figure 2.8. Open-loop block diagram of 2DoF ball balancer system [208]

Considering this, the controlling model for the x-axis of the SRV02 integrated with the ball balancer system is illustrated in Figure 2.9. The ball balancer block diagram explains the controlling in two loops. The first one for the SRV02 motor model and the second one is the 1D ball balancer. The inner loop describes the SRV02 controller ( $Z_b(s)$ ) for position control of D.C. series motor and estimates the needed voltage to track the desired angle of the ball.

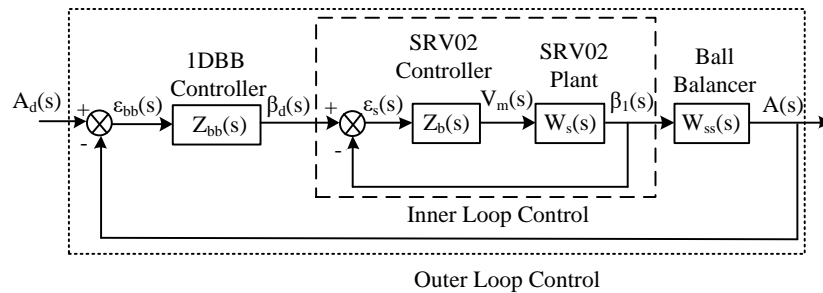


Figure 2.9. General closed-loop block diagram for ball balancer system [208]

For any given controller, the difference between the desired position and real-time position of the ball in the ball balancer model defines the input for the controller. As per the system specification, the plate dimensions are 27.5cm length X 27.5cm width. These dimensions restrict the freedom of movement for the ball within a length of 13.75cm considering the ball is to be positioned at the center of the plate. But this length makes the ball move to the corners of the plate, and the target of the research is to stabilize the ball in the center. So, a range of 0-6.875 cm has been taken to restrict the ball movement in the middle of the plate, and a reference of 3 cm has been chosen to keep the ball closer to the plate center. Hence, a 0.08 Hz square wave of amplitude 3 is considered as input for the system [66], [213]. The monitoring for the inner loop is done by controlling the SRV02 position by providing a proportional motor gain ( $K_{PG-14}$ ), and for outer loop control, the external controllers can be designed.

## 2.4 REVIEW OF CONTROL TECHNIQUES FOR 2DOF BALL BALANCER SYSTEMS

The above-discussed problem formulation for the position and balancing control identifies the approximation of underactuated nonlinear systems through automatic decision development and intelligent control methods [1]. This is an issue that appears in many problems [2] and can be tackled through various approaches. Further, the diversity and complexity of the ball balancer system have led researchers in the field to analyze the action of various linear, nonlinear, model-free, passivity, and intelligent controllers. These controllers are focussed at achieving self-balancing control and steady-state operation for various systems as follows:

*Linear Quadratic Regulator:* The LQR problem is developed based on a dynamical system where the equation of state evolution is described as a linear function. This linear function depends on the current state and input of the system which is subjected to quadratic cost. The objective of the controller is to minimize the cost function of system variables. For ball and plate system control, an alternative synthesis methodology has been designed using repetitive and resonant controllers where energy minimization depends on linear quadratic regulator (LQR) theory [214]. The LQR control delivers simplicity in the programming and guarantees that the system converges to the origin asymptotically while following the desired trajectory [215]. Further, the motion planning methods using LQR and the discrete model predictive controller has been suggested for a ball-plate Mecanum robot system [216]. The infinite horizon discounted LQR [217] defines non-asymptotic bounds but this increases the sample complexity when compared to other model-free and model-based methods. The major drawback of LQR is that it requests a good knowledge about the state of the system which is

not feasible every time. Such as, when the noise is considered in the system, the transition of states becomes difficult due to deterministic assumption made for the transition of other states.

*Back-Stepping Controller:* The back-stepping controller conducts a recursive procedure with the help of a Lyapunov function and follows a systematic design method for various nonlinear systems. The work by Kazim et. al represents a trajectory tracking control for ball and plate systems using robust backstepping control. The controller rejects bounded uncertainties with unknown periodicity [218]. It can guarantee global stability and improvement of tracking and transient performances. The limitation of the conventional backstepping controller is that it can apply only when the state model is in strict feedback form [219]. To avoid this problem, the backstepping controller is designed by combining it with  $H^\infty$  tracking controller while dealing with external disturbances for a ball and plate system and shows the asymptotic closed-loop stability [220]. Some other control methods like adaptive fuzzy [221] or neural networks [222] have been combined with backstepping to solve this model inaccuracy problem. However, these methods require knowing full-state output information.

*Proportional–Integral–Derivative controller:* There are numerous control strategies available in the literature for achieving self-balancing control with balancer systems but the PID controller is widely adapted in practical engineering applications. The proportional integral derivative (PID) controller which is nonlinear has been considered for one-to-one control of the ball on the plate system in classical control [223]. The reaction of PID is upgraded, when structured dependent on generalized Kalman-Yanukovych-Popov lemma (GKYPL) strategy, and contrasted a typical PID concerning the relentless state reaction [11]. The major advantages with the PID controller are its simple structure, high reliability, and good stability. But the key drawback while dealing with traditional PID controllers is their problem with parameter tuning. There are many techniques available in the literature to achieve the tuning of PID parameters. These techniques involve various intelligent techniques [12], self-tuning algorithms [13], genetic[227], and evolutionary algorithms [228].

The limitation of PID comes into the scenario when the system is subjected to parametric uncertainty and external disturbances. In practical application [16], [17], PID handles the uncertainty in a wide manner but limited to handle only constant parametric uncertainty and required an accurate system model for implementation purposes.



*Sliding Mode Controllers:* The sliding mode control is one of the most effective control methodologies in dealing with a large class of uncertain systems. The controller consists of a high-frequency switching term that completely compensates matched perturbation and provides both finite-time convergence and insensitivity concerning matched uncertainties/disturbances [231]. The sliding mode controller has been designed for achieving self-balancing control of the ball and plate system using the Stewart platform with rotary actuators [232]. An extension of sliding mode termed as an interpolating sliding mode observer is designed for balancing the system [233]. This observer incorporates multiple linear observers through interpolation of multiple estimates, which is treated as a type of adaptation. Besides, a fractional-order sliding mode controller (FOSMC) is applied to the problem of trajectory control of the ball while maintaining the non-linearity of the system to keep the minimum deviation between the simulated model and the actual system [234]. To achieve robustness while controlling, the sliding mode approach is considered as a reasonable solution [235]. But the switching surface behavior and chattering problem of the sliding mode controller makes it insensitive to external disturbances and parametric variations. The literature to overcome the chattering phenomenon of conventional SMC with enhanced efficiency has been widely discussed [236]-[237]. In [236] a memory-based sliding surface is proposed which consists of not only the current state but also the delayed state. It is different from the conventional memoryless sliding surface, as here both robust and adaptive fuzzy sliding mode controllers solve the problem of switching surfaces and make the closed-loop control system asymptotically stable. With the improvement in sliding mode controller, it provides finite-time stability of the closed-loop system, but it is limited to the continuous-time system and may not apply for the discrete-time system.

*Model Predictive Control:* The model predictive control algorithms are based on numerically solving of an optimization problem at each step, constrained optimization, and receding horizon control. In the MPC approach, the current control action is computed on-line rather than using a pre-computed, off-line, control law. The future state of the system is predicted with the help of MPC and generated control vector. The cost function is minimized for the vector over a predicted horizon during the constraint existence. System input is applied at any instant of time after generating the computed control vector and the rest of the value is discarded due to its inability to find the solution of violating input and output constraints. The model predictive controllers were also widely employed with ball balancer systems because of their advantages with time-varying reference [25]. The dynamic trajectory planning and control

for ball motion on the plate have been done using MPC which tracks periodic references and secure the system stability [239]. This method ensures that the closed-loop system converges asymptotically to the optimal admissible periodic trajectory while promising constraint satisfaction. In [238], model predictive control has been implemented on a limited-performance microcontroller and then applied on tracking control problem using a laboratory-built ball-and-plate system. In this method, MPC solves the quadratic programming problem as it creates difficulties in implementing MPC. To solve QP, the author converts quadratic structure into an equivalent nonnegative least-squares problem and develop the solver in a C-program. The separation of the offline calculation from the online process diminishes the calculation time. Further, to handle the uncertainty of accurate bounds the  $H_\infty$  combined with model predictive controller (MPC) and ensured the closed-loop performance under the explicit control law of MPC. The generated control law focuses on enhancing the utilization of given constraints with the help of the optimization of the receding horizon. This will improve the closed-loop disturbance attenuation level and is optimized towards the lower bound given by unconstrained  $H_\infty$  control. The problem with this technique is that here MPC optimizes the  $H_\infty$  performances for predefined trajectory but ignores the previous or history states to achieve closed-loop stability. This problem is explained by Hong Chen [240], as past states to define  $H_\infty$  performance is finite. However, the drawback with these predictive controllers is the need for an accurate dynamic model, which implies high computational cost and considered uncertainties make the problem difficult to solve.

*H-infinity Controller:* The H-infinity controller is a robust control method which is a part of control theory that deals with improbability in its approach to controller design. Robust control methods are designed to function in such a manner that the system parameter remains within some standard limits. These methods attempt to achieve robust performance or stability in the occurrence of bounded modeling errors. The controller is designed to minimize the worst-case effect of the disturbance on the output/error signal as measured by the L2 norm of the signals.

In the modern control approach,  $H_\infty/H_2$  controllers have been designed to deal with uncertainty. These controllers are robust under the assumption that the uncertainties are norm-bounded, and designed for the worst-case scenario [241]. The  $H_\infty/H_2$  controllers are responsible to provide guaranteed stability but within the bounds due to the modeling error [242] or filtering error [243]. Testing the uncertainties within these bounds is unacceptable for various real-world applications [155], even if the problems are solvable. Further to find

accurate bounds for the uncertainties to be allowed in the system, a constraint handling based controller has been designed [244] for linear systems.

*Neural Network Controllers:* The use of neural networks for solving highly nonlinear control problems is widely seen in the literature [245]-[246]. The operation of neural networks involves many simple and similar processing elements at its inputs and outputs of its structure. The processing elements have internal parameters called weights which help in altering the behavior of the whole network to achieve optimal output for the controller and its plant. For balancing purposes, a recurrent neural network is designed in such manner that input for the controller is differences of the ball's consecutive positions[247]. The weights of the network are updated by node decoupled extended Kalman filter (NDEKF) algorithm. In another work, for ball and beam system feedforward neural network is designed with feedback linearization in [248]. The method ensures the exponentially global uniform ultimate bounded stability and rejects the disturbance without using any learning algorithms. Further hybridization of neural with the fuzzy controller has been done by the various researcher for balancing problems. The neural network with fuzzy has been proposed for nonlinear dynamic ball balancing on beam in [249]. An indirect adaptive control strategy is presented using hierarchical fuzzy CMAC neuronal networks [250]. The CMAC is an auto-associative feed-forward memory artificial neural network, which guarantees the stability by theoretical analysis and assures that all signals are bounded. The serious drawback with most neuro-fuzzy methods is that they do often furnish rules without a transparent interpretation. Because of this difficulty, they have not yet reached widespread acceptance in industrial applications, despite the good performance they offer with a reduced design effort.

*Fuzzy Logic Systems:* Fuzzy control delivers a proper approach for demonstrating, controlling, and employing heuristic knowledge of humans to control any system. The main elements of the fuzzy controller are rule-base, inference mechanism, fuzzification interface, and defuzzification interface. Fundamentally, the fuzzy controller act as an artificial decision-maker that works in a closed-loop system of real-time problem. It gathers plant output data  $y(t)$ , compares it to the reference input  $r(t)$ , and then decides what the plant input  $u(t)$  should be to ensure that the performance objectives will be met. A fuzzy controller with online learning control has been demonstrated to an open-loop unstable system [251] of the ball and plate system. The FLC learns from a conventional PD controller with a known tuning parameter. The extension of the controller has been done for online tuning of the fuzzy rule base. The rule

base of the proposed online fuzzy controller is empty at the start and as time goes by, the rules are generated and the rule -base is compiled.

Besides, the self-balancing control, trajectory tracking and ball position control of the ball and plate system were achieved by various intelligent and hybrid controllers such as fuzzy [252]-[253], fuzzy cerebellar model articulation controller [250], and particle swarm optimization based fuzzy-neural controller [254]. In another approach, the supervisory fuzzy controller has been proposed based on a single input rule module and visual servo control for trajectory tracking and motion control of the ball on the plate [255]. Another hybrid controller has been designed with a sliding mode controller tuned by the supervisory fuzzy for position control [236].

For the study on the multivariable and complex B&P system, a touchscreen, and a rotary pneumatic cylinder are used as a replacement for of a camera and step motor, correspondingly and controlling has been carried on fuzzy & state observer [253]. Fuzzy visual control of the Mamdani-type model has been applied to control the movement of a ball on the plate [256]. The problem with fuzzy control is to find proper fuzzification, de-fuzzification method, and fuzzy rules. So, neuro-fuzzy identifiers are proposed to solve this problem [257]. Complexed and metacognitive valued neuro-fuzzy systems are designed and studied over different control problems [258].

Neuro-fuzzy system application has been reviewed in the student modeling system, unknown nonlinear system, electrical and electronics system, traffic control, image processing, and feature extraction, NFS enhancements and social sciences, technical diagnostics, and measurement [259]–[262]. A neural network with fuzzy has been proposed for nonlinear dynamic ball balancing on beam [247], [263]. Further, the cost control problem of the T-S fuzzy system is resolved by the type-2 fuzzy neural controller for the uncertain stochastic system. In this, a learning model has been designed based on type-2 fuzzy neural networks theory for the ball-beam system [264]. The robust stability of the closed-loop system is definite by the Lyapunov theorem, and all error signals are uniformly ultimately bounded. The major challenge to these intelligent techniques is the translation of the information contained implicitly in a collection of data points into linguistically interpretable rules. Conventionally, the neuro-fuzzy methods have been developed to overcome this issue.

*Hybrid Controllers:* The hybrid control techniques combined the advantages of two or more controllers to overcome the individual drawbacks of each controller. There are many techniques available in the literature to achieve the tuning of PID parameters and able to generate hybrid control action. The various optimization techniques like a genetic algorithm (GA), particle swarm optimization (PSO), and shuffled frog leaping algorithm (SFLA) algorithms along with the linear, non-linear controllers have been represented in [265]–[267]. In [268], a GPSO ( gaussian particle swarm optimization) is employed to dynamically perform nonlinear constraint optimization problem of ball and plate system. The GPSO is combined with NMPC (nonlinear model predictive control) through Taylor expansion and solve a quadratic programming problem with unknown parameters. In another work, the on-line training of the PIDNN controller has been done using improved differential evolution particle swarm optimization (DEPSO) algorithm for trajectory tracking of the ball and plate system [269]. The PSO algorithm is improvised using a DEPSO to train the weighting factors of the multi-layered feedforward neural network. In another approach, a hybrid GA based PIDNN controller has been designed where the weighting factor of multi-layered feedforward neural network is trained by the genetic algorithm [270]. For optimization of trajectory tracking of the ball on the plate, the genetic algorithm is hybridized with fuzzy [271]. The output membership functions of the fuzzy planning controller are optimized using a genetic algorithm. However, these techniques are focused on saving the better generations resulting in local optimum rather than the global optima. Further, the problems due to weight adjustment of the intelligent controllers, low memory, premature convergence, weak local search, and high computational efforts for genetic and other evolutionary algorithms has resulted in the need for an optimal multi-objective approach for solving combinatorial optimization problems [272], [273].

*Randomized Algorithms:* The randomized algorithm uses a source of pseudo-random numbers with the input data to check the behavior of output data with different probability [274], [275]. Due to randomness in the behavior of the input side, the output will vary every time for the same input. As the random variable is involved at the initial stage so, even for a fixed input the algorithm will give different outputs every time. The desired output for the system will depend on the random choices neither input distribution nor on running time. To assess the system robustness the main concern is to deal with the maximum amount of uncertainty that a system can handle or tolerate, and a randomized algorithm is the best solution to this problem due to the involvement of uncertain random variables.

Further to deal with probabilistic constraints with random uncertainty, it is typically not possible to prevent failure in all possible cases. So, a simultaneous perturbation stochastic approximation (SPSA) algorithm which recursively generates estimates along random directions has been designed to deal with random uncertainty [276]–[279]. Conventionally, the stochastic approximation methods were used for statistical computations and later emerged as a separate field of control theory [280]. In the initial stage, these methods were proven for the minimization of stationary functionals [281]. Later, the drawbacks of the gradient and newton methods while dealing with time-varying functionals due to the known bounds of the Hessian matrix two times differentiable functionals resulted in the development of stochastic approximation algorithms [282]. But the issue of constant step size in stochastic approximation limited their applications to time-varying systems and tracking problems [280].

In general, linear controllers offer a simple way of designing closed-loop control for balancing systems. Various linear controllers were available in the literature to solve the difficulties in balancing systems [283]–[285]. But the complicated nonlinear dynamics of underactuated systems affect the capability of providing a plausible solution and limits the generalized applications of control laws. These controllers face various drawbacks during the early stage of their development such as heavy effect on the speed of system response. To solve this problem various nonlinear techniques have been explored by researchers [286], [287]. For ball and beam system control passivity-based nonlinear controller has been designed [235] where the aim is to passivate the system with a storage function, which has a minimum at the desired balance point. This has a disadvantage with differential feedback as it cannot amplify the measurement noises. The ball can be positioned around a fixed plate area by controlling the motors in the electrical system. This control operation is performed by developing a feedback controller. Conventionally, exchanging controllers is developed depending on the non-straight examination of the ball and plate framework [288]. Simultaneously, depending upon the Lyapunov dependability hypothesis, the back-venturing controllers were developed to accomplish an explicit control law and achieved the control execution rapidly [289]. An inventive disturbance observer utilizing ADRC (Adaptive Disturbance Rejection Control) strategy is intended to maintain a strategic distance from destructive grating consequences for the plate for adjusting the ball [290]. Further, disturbance rejection controllers [291] and metaheuristic optimization algorithms [292] were illustrated on the ball and plate. Besides, the position of the ball can likewise be controlled through a machine vision framework. The picture handling algorithms for the same framework was linked and implemented on an FPGA (field-

programmable gate array) appliance to encounter continuous requirements of the real-world [293]. B&P demonstrated by the computer-generated laboratory utilizing 3D java re-enactment pursues the predefined direction for the ball and accomplished point to point controllability [294]. The full state feedback input linearizing control law was built up and could control divergent rappings and additionally link among the co-ordinate axis [295] systems to achieve desired tracking performance.

*Fault-based control:* Various fault-tolerant control techniques such as adaptive linear parameter varying [296], network-based filter [297], and parametric linear quadratic regulator [298], etc. have been implemented for achieving self-balancing control. In [297], an observer-based fault detection filter is developed considering network models, along with a controller coordinated design. These network-based models are developed by subjecting the system to actuator faults and wave-induced disturbances. Further, Wang et al. [299] developed an event-triggered fault detection filter considering the network environment. A similar framework of faults, as discussed in [297] were introduced to develop a residual model, and an integral-based event is introduced to save communication resources. Apart from the network-based models, intelligent collision avoidance maneuvers were studied by Campbell et al. [300]. The research identified the challenges posed by unmanned marine vehicles for achieving conformance with the navigation rules during marine traffic and automatic obstacle avoidance. A comprehensive review of fault detection using machine learning classifiers for unmanned vehicles can be found in [193]. From the widely available classifiers and developed fault classification mechanisms, it is observed that very delicate details of the data are neglected while classifying the fault. To overcome this and improve classification performance, the combination of classifiers is suggested by many researchers [206]. But these methods proved to be costly and lacked while dealing with delicate data and unknown faults.

## 2.5 INTELLIGENT CONTROL OF 2DOF SYSTEMS

In this section, the neural integrated fuzzy controller is considered as a baseline for analyzing the behavior of the system and to develop various optimal and hybrid control strategies.

### 2.5.1 *Neural-integrated-fuzzy controller*

A fuzzy inference system is a fuzzy model that maps inputs to outputs based on the concepts of fuzzy set theory, fuzzy if-then rules and fuzzy reasoning. While FIS systems have been

widely popular due to their use in numerous real-world control applications, they rely on two important factors, i.e. knowledge acquisition and human expertise which often pose a limitation. On the other hand, artificial neural networks provide a learning ability to systems by making them more adaptive. A neural integrated fuzzy (NiF) control combines the advantages of fuzzy inference system (FIS) and artificial neural networks into one single hybrid system. Conventionally, the fuzzy logic for the single axis control of robotic systems was achieved by designing the rules for the uncoupled motion between different actuator systems. Generally, the fuzzy inference schemes are divided into two models, Mamdani [301] and TSK (Takagi-Sugeno-Kang) [302]. In a Mamdani model, the fuzzy rules are synthesized by developing a set of linguistic control rules that are obtained from the multiple test conditions and experiences of human operators. Moreover, in the TSK-type fuzzy system, the consequent of each rule are the functions of input linguistic variables. Hence, the general adopted function of TSK-fuzzy can be interpreted as a linear combination of input variables and a constant term. This phenomenon of fuzzy is adapted in this research to develop the NiF controller. An initial fuzzy model along with its inputs are derived with the help of rules extracted from the input output data of the system that is being modelled are given as

$$\text{Rule } q: \text{ if } x_1(t) \text{ is } A_{q1}, \dots, \text{ and } x_n(t) \text{ is } A_{qn} \text{ then } \hat{y}(t+1) = b_q + \sum_r a_{qr} x_r(t) \quad (2.30)$$

where  $x = [X_1, X_2, \dots, X_n]^T \in X$  is the vector of input variables &  $A_{q1}, A_{q2}, \dots, A_{qn}$  are fuzzy sets,  $q = 1, 2, \dots, K$  represent the number of rules, and  $\hat{y}$  is the rule output. The final network output is a weight average of each rule's output which is given as

$$\hat{y} = \frac{\sum_{q=1}^K \eta_q(x) \hat{y}_q}{\sum_{q=1}^K \eta_q(x)} \quad (2.31)$$

where  $\eta_q(x)$  is the degree of initiation of the  $q^{th}$  rule:

$$\eta_q(x) = \prod_{r=1}^n \beta_{A_{qr}}(x_j) \quad (2.32)$$

where  $\beta_{A_{qr}}(x_j)$  is the membership function of the fuzzy set  $A_{qr}$  at the antecedent (input) of  $S_q$ . Further, the use of neural network fine tunes the rules of the initial fuzzy model to produce the final NiF model for controlling the plant.



### 2.5.2 Architecture of Neural integrated Fuzzy controller

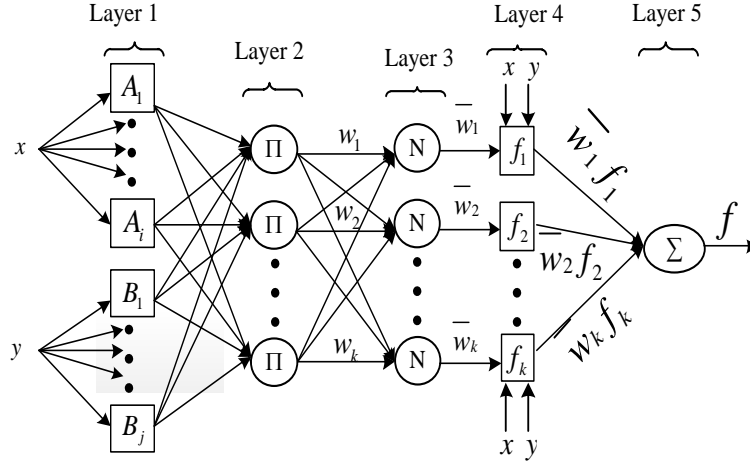


Figure 2.10. Schematic diagram of neural integrated fuzzy control

Figure 2.10 shows a simple NiF architecture based on the fuzzy antecedent and a functional consequent of a Sugeno model. For simplicity of illustration, a two-input system is considered and the rules are stated as

Rule 1: if  $x$  is  $A_1$  and  $y$  is  $B_1$ , then  $f_1 = p_1x + q_1y + r_1$

Rule 2: if  $x$  is  $A_2$  and  $y$  is  $B_2$ , then  $f_2 = p_2x + q_2y + r_2$

Rule  $k$ : if  $x$  is  $A_k$  and  $y$  is  $B_k$ , then  $f_k = p_kx + q_ky + r_k$

Here  $x$  and  $y$  are the crisp inputs to the nodes;  $A_1, A_2, A_3, \dots$  and  $B_1, B_2, B_3, \dots$  are the linguistic variables (small, medium, etc) associated with the inputs  $x$  and  $y$  respectively;  $f$  corresponds to the weighted average which is calculated as:

$$f = \frac{\sum w_i f_i}{\sum w_i} \quad (2.33)$$

Let  $O_i^k$  be the  $i^{th}$  node in the  $k^{th}$  layer. An adaptive node has been indicated by a square and the fixed node is given by a circle. Further, the five layers of the NiF architecture are detailed as

**Layer 1:** The first layer is the input layer with adaptive nodes. Also called as the input linguistic layer, it is responsible for the fuzzification of the crisp inputs. The output of each node  $i$  is given by:

$$O_i^1 = \mu_{A_i}(x) \quad (2.34)$$

Usually,  $\mu_{A_i}(x)$  is chosen to be a bell-shaped membership function with maximum equal to 1 and minimum equal to 0.

$$\mu_{A_i}(x) = \frac{1}{1 + \left[ \left( \frac{x - c_i}{a_i} \right)^2 \right]^{b_i}} \quad (2.35)$$

**Layer 2:** This layer is also called as the condition layer. The nodes in this layer are fixed and labelled as  $\Pi$ . Their function is to multiply the incoming signals and output a product which acts as the firing strength of each rule.

$$O_i^2 = w_i = \mu_{A_i}(x) \times \mu_{B_i}(y) \text{ where } i = 1, 2, \dots \quad (2.36)$$

**Layer 3:** This layer is also called as the rule layer. Each node in this layer is fixed and labelled as  $N$ . The function of each node in this layer is to normalize the values of the firing strength of the rules produced by the previous layer.

$$O_i^3 = \bar{w}_i = \frac{w_i}{\sum w_i} \quad (2.37)$$

**Layer 4:** Also called as the consequent layer, the nodes are adaptive in this layer. The output of each node is given by:

$$O_i^4 = \bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_i) \quad (2.38)$$

**Layer 5:** The output linguistic layer, this layer produces a summation of all outputs from the previous layers to give a final output. It is responsible for the defuzzification of the outputs.

$$O_i^5 = \sum \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \quad (2.39)$$

The final output is calculated from  $f = \sum_k \bar{w}_k f_k$ , and elaborated as

$$f = \frac{w_1}{w_1+w_2+\dots+w_k} f_1 + \frac{w_2}{w_1+w_2+\dots+w_k} f_2 + \dots + \frac{w_k}{w_1+w_2+\dots+w_k} f_k \quad (2.40)$$

For the NiF implementation of the helicopter and ball balancer model, the design procedure completely depends on a data training method, calculating the input-output membership function, and rule base, to achieve optimum output.

### 2.5.3 NiF controller for trajectory tracking of helicopter

The development of the NiF model for helicopter system is achieved by measuring the error and change in error between the measured and reference pitch angles, and measured and reference yaw angles of the system. The data input selection was done by trial-and-error method, heuristically. Basically, NiF takes the initial fuzzy model generated by MATLAB functions and tunes it by means of a hybrid learning algorithm. At each iteration, an attempt is made to reduce the error measure, usually defined as the sum of the squared difference between pitch or yaw error and the rate of change of pitch error or yaw error. Training stops when either the predefined epoch number or error rate is obtained. When the values of the premise parameters are learned, the overall output can be expressed as a linear combination of the consequent parameters.

The input variables of the NiF are selected based on pre-processing of the original data and guidelines for input selection for NiF learning [303]. Real-world modelling problems usually involves tens (or even hundreds) of potential inputs and use them accordingly. Therefore, a heuristic way is needed to quickly determine the priorities of these potential inputs and use them accordingly. The rule bases of NiF are generated based on linear Sugeno fuzzy model using the hybrid algorithm. The MATLAB software is used to hypothesize a parameterized model (FIS). This is done using a subset of the predefined trajectory, normally referred to as training data. Then the other input/output data-subsets, normally referred to as checking data is used to train the model and mimic the training data by modifying the membership function parameters according to a chosen error criterion [304]. Further, to validate the model estimated by NiF, the input/output dataset is divided into checking and training data.

This ensures that the corresponding output datasets from the FIS model are similar and are fully representative of the network. Both the training and checking datasets are verified to check for the presence of noisy measurements in the dataset. Moreover, the training dataset checks the generalization capability of the resulting FIS, whereas the checking dataset is used to validate this FIS to observe any over-fitting concerns. Generally, over-fitting is accounted by comparing the training and checking errors. Ideally, both the errors must decrease at the same time throughout the training period. If they don't, then this indicates over-fitting. But if the checking error begins increasing even at the first iteration, while the training error decreases, then the trained FIS has to be retrained because clearly, this membership function is not the best choice for modeling the entire dataset. In such cases, other membership function choices can be made or the size of the dataset can be increased. These errors computed are actually root mean squared error (RMSE) which is given by:

$$RMSE = \sqrt{\frac{\sum_{t=1}^n (y_t - \hat{y}_t)^2}{n}} \quad (2.41)$$

where  $\hat{y}_t$  are the estimated values for times  $t$  of the specific target  $y_t$  for  $n$  samples.

The generalised representation for NiF controller implementation with the helicopter system is shown in Figure 2.11. The NiF controller is developed, first starting with the rules governing the uncoupled motion of pitch and yaw motors respectively.

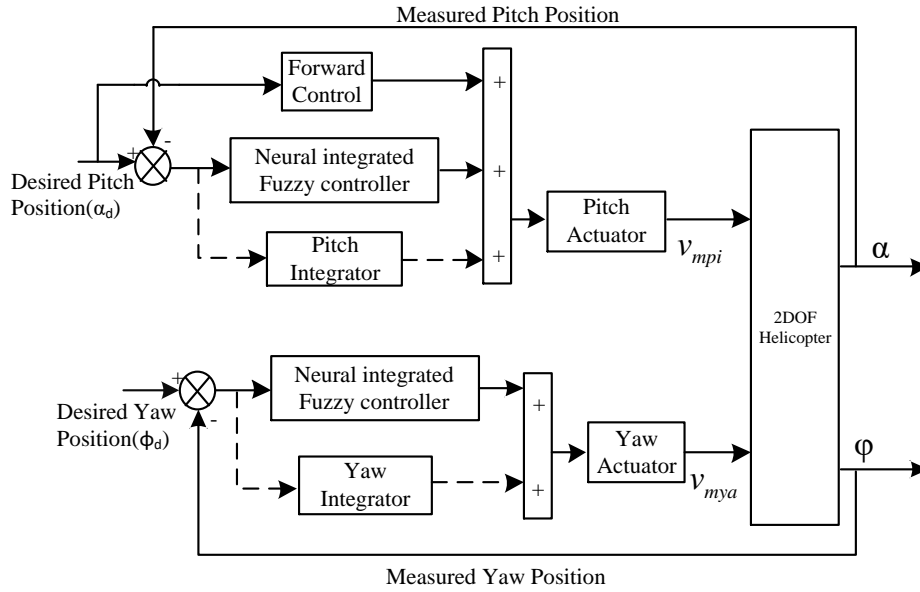
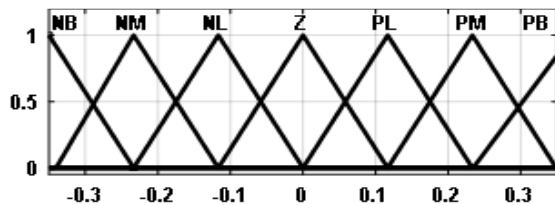
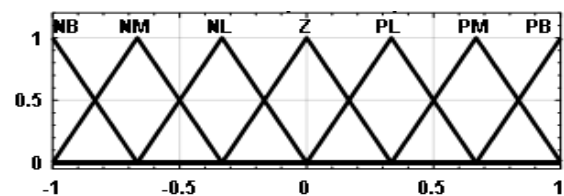


Figure 2.11. Block diagram of NiF controller for helicopter system

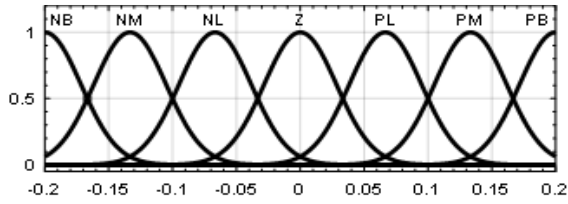
The pitch and derivative of pitch are the two inputs whose all possible combination within range is fed to train NiF. Similarly, for yaw controller design, the yaw angle and its derivative are considered with the NiF controller. The universe of discourse is chosen to be  $[-0.35, 0.35]$  for pitch error and  $[-1, 1]$  for pitch derivative error and the universe of discourse is chosen to be  $[-0.2, 0.2]$  for yaw error & yaw derivative error. NiF is trained for 50 epochs for pitch and yaw, using a combination of least square and back-propagation gradient descent method. Seven linguistic variables (NB, NM, NL, Z, PL, PM & PB) are associated with each input, so the input space is partitioned into 49 fuzzy subspaces, each of which is governed by fuzzy if-then rules. Here these variables correspond to: NB- negatively big, NM- negatively medium, NL- negatively low, Z-zero, PL- positively low, PM- positively medium & PB- positively big. In association with fuzzy, neural network will have the following number of neurons in each layer: 2 neurons are available in layer 1, 14 neurons are there in layer 2, the layers 3 and 4 have 49 neurons each, and 1 neuron is available in the output layer. The arrangement of membership function for the inputs of NiF controller are given in Figure 2.12.



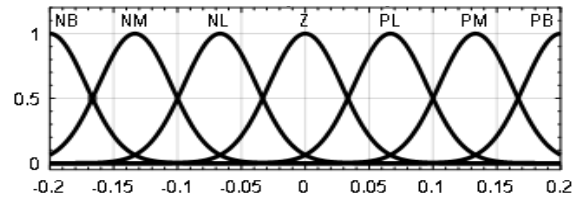
(a) Pitch angle error



(b) Rate of change of pitch angle error



(c) Yaw angle error

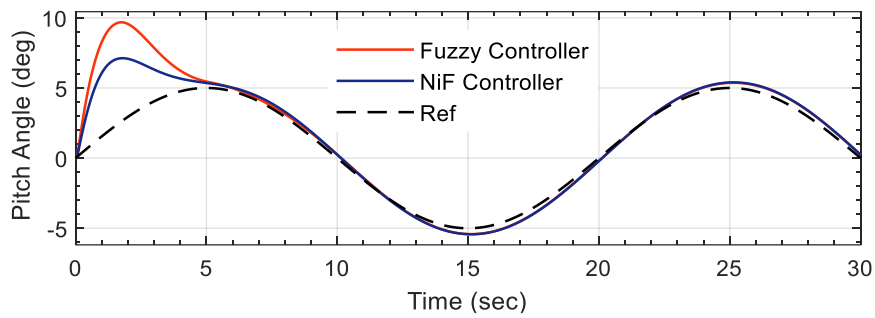


(d) Rate of change of yaw angle error

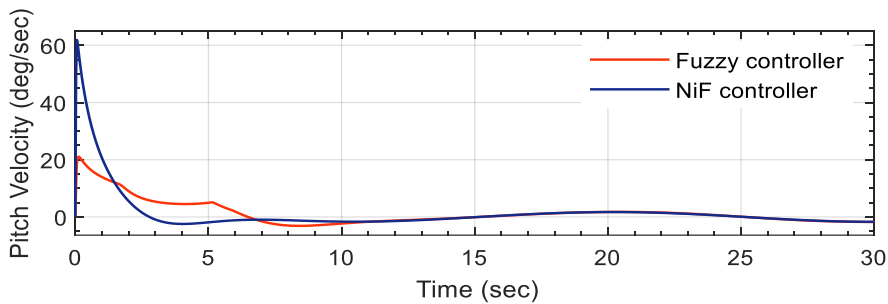
Figure 2.12. NiF membership function of the Pitch and Yaw errors and their derivatives after training

### 2.5.3.1 Simulation Analysis

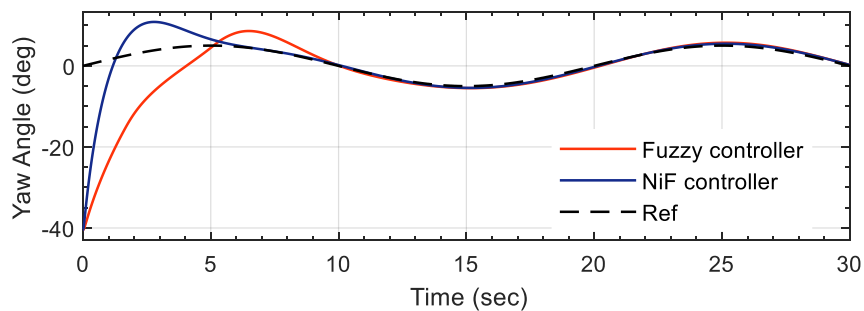
For a sine trajectory of amplitude 5, and frequency 0.8, the simulation analysis provides the pitch and yaw characteristics of the helicopter as shown in Figure 2.13. To assess the operation of the developed NiF controller, the simulation aspects are compared with the conventional fuzzy controller.



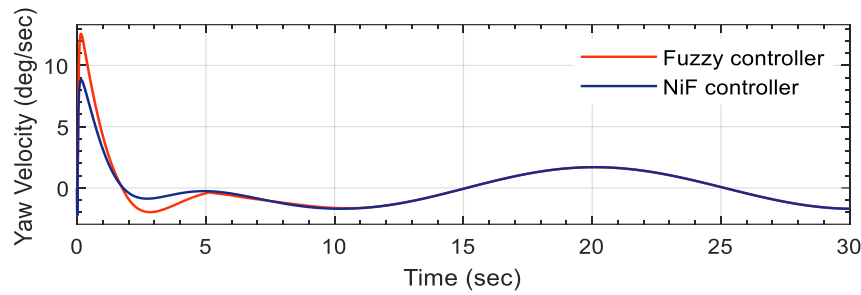
(a) Pitch angle



(b) Pitch velocity



(c) Yaw angle



(d) Yaw velocity

Figure 2.13. Pitch and yaw output with neural integrated fuzzy control of helicopter in Simulink

From the results in Figure 2.13 it is identified that, with the action of the conventional fuzzy controller, the pitch angle suddenly increases to 9.7 degrees generating a huge movement in the vertical position of the helicopter. Further, it is varied between 7 degrees to 5 degrees through the action of NiF controller. For horizontal motion, the angle variation by yaw is not sufficiently high during the conventional fuzzy control, which may cause a huge coupling misalignment between yaw and pitch motions. The action of the NiF controller provides high initial gain which is necessary to uplift the thrust force for rotation of blades of the helicopter. To achieve proper coupling between the rotors, the pitch velocity is increased to 62 degrees/sec from 20 degrees/sec, and yaw velocity is reduced to 8.8 degrees/sec from 12.5 degrees/sec by the action of NiF controller. Further, the time response analysis and root mean square error are analysed for the implemented control actions for better understanding of the superiority of the developed approach. The corresponding results are shown in Tables 2.1 and 2.2 respectively.

Table 2.1. Time response analysis for neural integrated fuzzy control of helicopter in Simulink

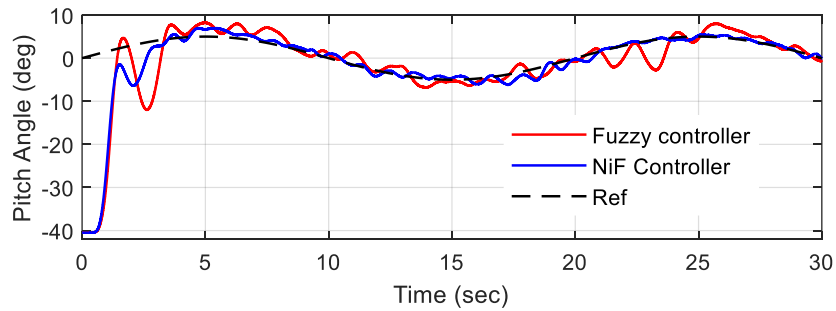
Controller	Pitch Response		Yaw Response	
	Settling time ( $t_s$ ) (sec)	steady-state error ( $e_{ss}$ ) (cm)	settling time ( $t_s$ ) (sec)	steady-state error ( $e_{ss}$ ) (cm)
Fuzzy Controller	6.7 sec	0.99 cm	9.2 sec	9.82 cm
NiF Controller	5.1 sec	0.18 cm	6.2 sec	0.51 cm

Table 2.2. Root mean square error for neural integrated fuzzy control of helicopter in Simulink

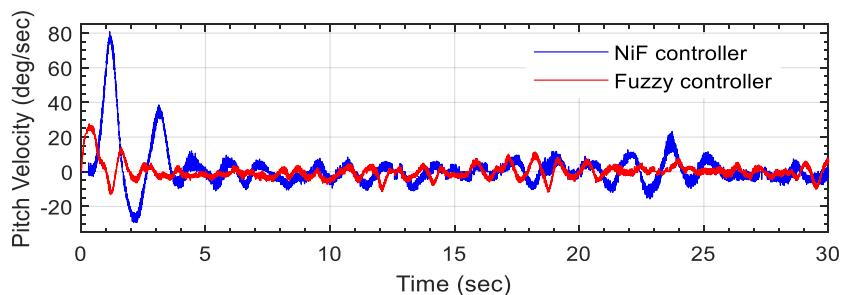
Controller	Root mean square error	
	Pitch angle (deg)	yaw angle (deg)
Fuzzy Controller	3.6816 deg	8.6938 deg
NiF Controller	2.4160 deg	0.2991 deg

### 2.5.3.2 Real-time analysis

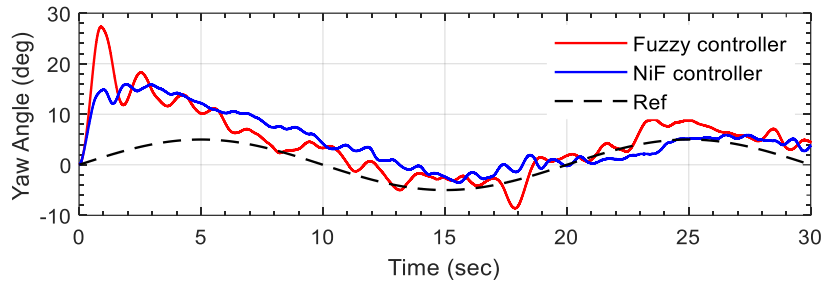
To analyse the action of the controller on the real-time operation of the helicopter system, the 2DoF helicopter system discussed in the above section is considered. The implementation of the controller is achieved through the hardware in loop-application programming interface (HIL-API) with the Quanser helicopter setup. Initially, the simulated models are calibrated for code generation with MATLAB/Simulink and Quanser software. Further, a C code generated in using the MEX function and interpreted to the hardware using the RS232 interface with the DAQ board. To achieve the efficient operation of the developed controller and simultaneous action of the helicopter system for every rule fired by the fuzzy, Immediate input/output is adapted. This performs the hardware-in-the-loop-application programming interface (HIL-API) operation of the developed controller with the real-time setup. The corresponding pitch and yaw characteristics for the real-time operation of the helicopter and their comparisons with fuzzy controller action are shown in Figure 2.14.



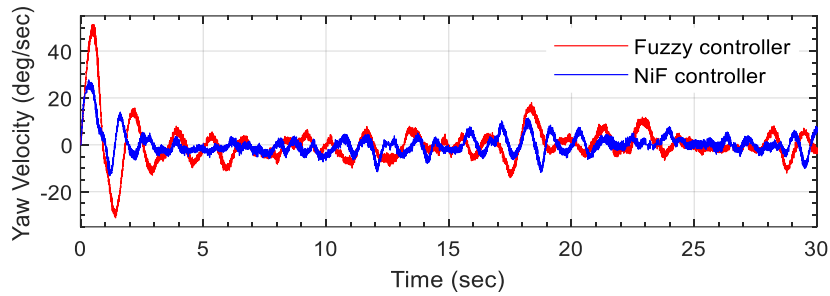
(a) Pitch angle



(b) Pitch velocity



(c) Yaw angle



(d) Yaw velocity

Figure 2.14. Pitch and yaw output with neural integrated fuzzy control of helicopter in real-time

From the results in Figure 2.14 it is identified that, the variation in oscillation with conventional fuzzy controller goes from 8 degrees to -12 degree, but the action of the NiF controller, reduces the oscillations to vary between 5 degree to -5 degree for pitch angle. For yaw angle, the oscillations with conventional fuzzy controller is 28 degree to -9.8 degrees, but with the action of the NiF controller, the oscillations are reduced from 15 degrees to -3 degrees. Further, the pitch and yaw motor provide the desired velocity from the controlled output of NiF controller. This clearly demonstrates the good performance and robustness property of the proposed controller in trajectory tracking operations in real-time as well. Further the time response analysis and root mean square error are calculated for the real-time operation of the helicopter with NiF and fuzzy controllers and the results are shown in Tables 2.3 and 2.4 respectively.

Table 2.3. Time response analysis for neural integrated fuzzy control of helicopter in real-time

Controller	Pitch Response		Yaw Response	
	Settling time ( $t_s$ ) (sec)	Steady-state error ( $e_{ss}$ ) (cm)	Settling time ( $t_s$ ) (sec)	Steady state error ( $e_{ss}$ ) (cm)
Fuzzy Controller	8.47 sec	3.76 cm	19.93 sec	12.31 cm
NiF Controller	6.77sec	1.18 cm	18.72 sec	7.61 cm



Table 2.4. Root mean square error for neural integrated fuzzy control of helicopter in real-time

Controller	Root mean square error	
	Pitch angle (deg)	yaw angle (deg)
Fuzzy Controller	6.3189 deg	13.2147 deg
NiF Controller	3.1716 deg	7.1346 deg

As per the responses available in the graph it has been observing that NiF controller has an appropriate response to the lowest error with the adequate time response and shows the excellent control on the axis without any oscillations.

#### 2.5.4 NiF controller for ball balancer system

For the design of the NiF controller for the ball balancer system, the first step was to collect the relevant data that may be used for training. Training of the data to design NiF controller for its desired behavior, the idea was to use a fuzzy controller substituting the NiF. The second step was to choose the input-output data set for training by hybrid optimization method. The ball position error  $e(t)$  and change in position error  $de(t)/dt$  are the two inputs whose all possible combination within each range is fed to train NiF. The control signal  $u(t)$  obtained from fuzzy control is fed as a target output. The universe of discourse is chosen to be  $[-0.2, 0.2]$  for both ball position error and change in position error and universe of discourse for the expected plate angle is  $[-30, 30]$ . Block diagram for NiF controller is shown in Figure 2.15.

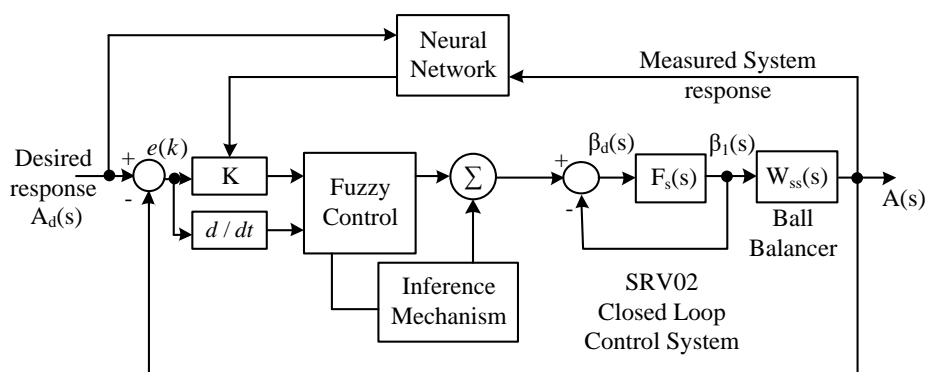


Figure 2.15. Block diagram for NiF controller for ball balancer operation

The NiF controller is trained for 200 epochs, using a hybrid training algorithm which is a combination of least square and back-propagation gradient descent method. Consequently, membership function parameters of single-output, Sugeno type fuzzy inference system are

obtained. Seven linguistic variables (MF1, MF2, MF3, MF4, MF5, MF6 & MF7) are associated with each input, so the input space is partitioned into 49 fuzzy subspaces, each of which is governed by fuzzy if-then rules. The membership functions for both the inputs of the NiF controller for ball balancer control are shown Figure 2.16.

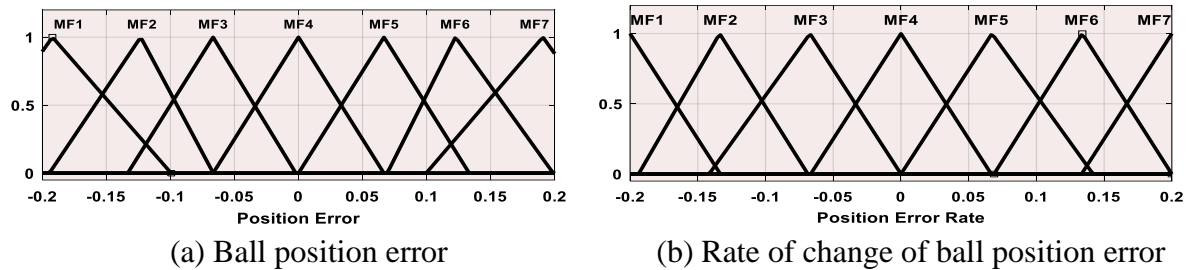
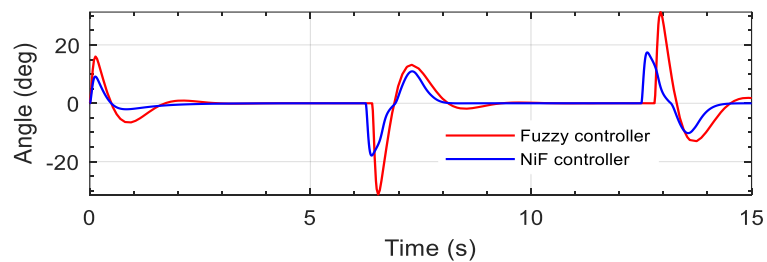


Figure 2.16. NiF membership function of the position error and position rate after training

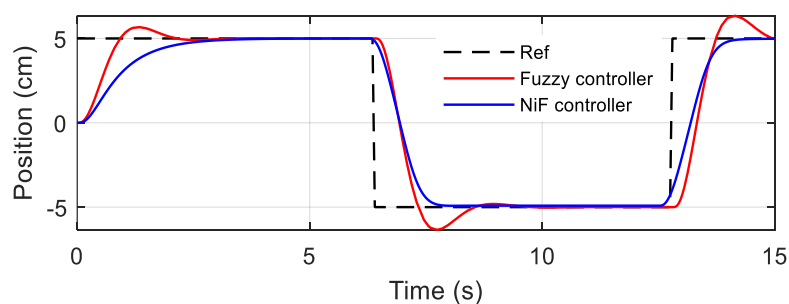
In association with fuzzy, the neural network will have the following number of neurons in each layer: 2 neurons in layer 1, 14 neurons in layer 2, 49 neurons in layers 3 and 4, and 1 neuron in layer 5.

#### 2.5.4.1 Simulation analysis

The simulation analysis of the developed NiF controller on ball balancer system is carried out by tracking the ball position around a square trajectory. The square trajectory is of 5 amplitude and 0.8 frequency. The results corresponding to the control action in simulation are shown in Figure 2.17. Further, to assess the superiority of the proposed approach on the ball balancer setup, the results are compared with a conventional fuzzy logic controller.



(a) Plate angle



(b) Ball position

Figure 2.17. Ball balancer output with neural integrated fuzzy control in Simulink

The focus is on the movement of the ball on the plate with less vibration in ball movement. In Figure 2.17 (a) the Plate angle on the x-axis is 9.1 degree, which is the lowest plate angle as compared to fuzzy controller and hence the plate will not have oscillations and ball will move very slowly and steady within a short period of time. Hence the position of the ball from Figure 2.17 (b) shows that variation in ball position while moving on the plate surface having the lowest settling time with NiF control. The time response analysis and root mean square error measurement for the action of the developed controller on the ball balancer system are discussed in tables 2.5 and 2.6 respectively.

Table 2.5. Time response analysis for neural integrated fuzzy control of ball balancer in Simulink

<b>Controllers</b>	<b>Peak Time (<math>t_p</math>) (sec)</b>	<b>Settling Time (<math>t_s</math>) (sec)</b>	<b>Peak Overshoot (<math>M_p</math>) (%)</b>	<b>Steady-state error (<math>e_{ss}</math>) (cm)</b>
Fuzzy controller	0.3 sec	3.8 sec	13.5 %	0.000472cm
NiF Controller	0.2 sec	2.51 sec	9.56e – 06 %	8.35e – 05 cm

Table 2.6. Root mean square error for neural integrated fuzzy control of ball balancer in Simulink

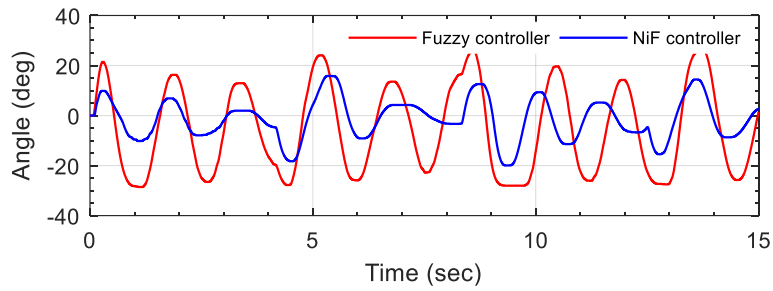
<b>Controllers</b>	<b>Root mean square error</b>	
	<b>Position (cm)</b>	<b>Angle (deg)</b>
Fuzzy controller	3.16189 cm	3.2178 deg
NiF Controller	1.9358 cm	0.9163 deg

Further, the training of data with NiF improves peak overshoot & steady-state-error and hence the position of the ball while moving on the plate surface having the better settling time.

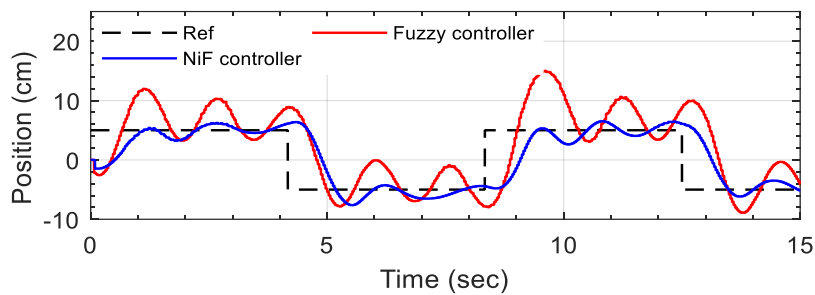
#### 2.5.4.2 Real-time analysis

The real-time implementation of the NiF controller action on the ball balancer setup is achieved by considering the 2DoF ball balancer system discussed in above sections. The controller is implemented with the real-time setup by performing the data exchange between the numerical simulation and the laboratory setup. Initially, the simulated models are calibrated for code generation with MATLAB/Simulink and Quanser software. Further, the generated code is interpreted to the hardware using the RS232 interface with the DAQ board. Outcomes from the implementation of these controllers to the hardware and simulation are verified based on the

momentary analysis. The action of NiF controller on real-time system and its comparison with the conventional fuzzy controller are shown in Figure 2.18.



(a) Plate angle



(b) Ball position

Figure 2.18. Ball balancer output with neural integrated fuzzy control in real-time

From the results in Figure 2.18 (a) it is identified that the plate angle varies between +15 and -20 degrees. This made the ball movement smooth and stable for the NiF controller. Further, the scaling factor used for stabilizing the ball position around the predefined trajectory with less oscillations as shown in Figure 2.18 (b). The values of scaling factors for error, change in error and control output in case of NiF controller were 0.5, 110, and 4 respectively. Further, the time response analysis and root mean square error measurements for the applied control actions on the ball balancer system are discussed in tables 2.7 and 2.8 respectively.

Table 2.7. Time response analysis for neural integrated fuzzy control of ball balancer in real-time

<b>Controllers</b>	<b>Peak Time (<math>t_p</math>) (sec)</b>	<b>Peak Overshoot (<math>M_p</math>) (%)</b>	<b>Steady-state error (<math>e_{ss}</math>) (cm)</b>
Fuzzy controller	0.47 sec	36.8 %	3.68 cm
NiF Controller	0.35 sec	4.16 %	1.08 cm

Table 2.8. Root mean square error for neural integrated fuzzy control of ball balancer in real-time

Controllers	Root mean square error	
	Position (cm)	Angle (deg)
Fuzzy controller	5.9617 cm	4.1209 deg
NiF Controller	3.6241 cm	2.7918 deg

From the results it is identified that the response of NiF is best based on settling time, peak overshoot and steady-state-error. The hybrid algorithm-based controller implemented on ball balancer system proved to be more efficient in terms of various performance parameters when compared with conventional controllers.

## 2.6 CONCLUSION

In this chapter, the modelling and control of helicopter and ball balancer system along with the details of the literature and action of baseline intelligent control approaches are developed. From the literature, it is observed that the control aspects like position control, path planning, trajectory tracking and balancing control [305], [306] are necessary to perform complex tasks in dynamic environments. Moreover, it is identified that many efforts have been made to achieve them especially in the field of trajectory tracking, path planning [83], [93], [94], [105], and balancing control [307]–[309]. Although the afore-mentioned techniques facilitate robust position control with respect to model and parameter uncertainties, they mainly focus on the stability of the closed-loop dynamics without emphasizing the effects of system complexities, faults, or external disturbances for high-accuracy control. Hence, the integration of such designs into conventional architectures could provide solutions that ensure high performance in both normal conditions and in the presence of failure modes. Besides, from the development of NiF controller for both the helicopter and ball balancer systems, the problems with control action in the presence of parametric uncertainties are identified. This recognized the need for designing hybrid control approaches, randomized algorithms, fault classification-based control techniques, and reinforcement learning algorithms.

## Chapter 3. OPTIMIZED INTELLIGENT CONTROL FOR 2Dof SYSTEMS

### 3.1 WAVELET FUZZY BASED CONTROL SYSTEM

The use of wavelets is widely accepted in the field of control theory especially to deal with uncertain systems [310], [311], due to their signal processing abilities and their advantage with hybridization of controllers [312]. Hence, the proposal of an intelligent wavelet controller for the control of nonlinear systems can be taken up as a giant opportunity for innovation purposes. The wavelet transforms (WT) investigate the input signals which extricate and distinguish the segments of the frequency signal in a different period and represents them to another frame. Further, this idea of signal sampling and reconstruction is relatively new for path tracking and stabilizing applications with zero oscillations. This concept is remarkably important in the field of robotics to understand the balancing mechanism of various applications.

#### 3.1.1 Wavelet Transform

Any signal can easily be represented by WT using the execution of typical math series calculations. These signals are investigated in both the time and frequency domain[313], [314]. Furthermore, the WT can focus the energy of the processed signal into a finite number of coefficients and delivers the frequency-time localization of the signal [315]. The transformed signal is represented in mathematical form as

$$wt(\beta, s) = \frac{1}{s} \int x(t) \delta^* \left[ \frac{t-\beta}{s} \right] dt \quad (3.1)$$

where,  $x(t)$  is the given signal,  $\beta$  is the translational parameter,  $s$  is the scaling parameter, and  $s > 0$  illustrates the size of the window, which describes the graded wavelet resolution  $\delta^* \left[ \frac{t-\beta}{s} \right]$  in frequency-time domains. Further, these parameters decrease as the frequency increases and vice-versa. Moreover, the signal  $x(t)$  after discretization using WT method are represented as

$$WT_{a,b}x(t) = \int_{-\infty}^{\infty} x(t) \delta_{a,b}^*(t) d(t) \quad (3.2)$$

where,  $\delta^*(t)$  depicts the function of the wavelet,  $a$  and  $b$  represents the dilation and translational parameters, respectively.

##### 3.1.1.1 Signal Pre-processing

Generally, the electrical and physical noises in a system restrict the clean signals and cause the operating system to draw various frequencies while processing the signal. The controller based

on WT can perform tremendously well when it comes to the discretization of such noisy signal into various bands of frequency. The wavelet decomposes the signal into different bands which is a major drawback in conventional filters, especially with Fourier transformations. Hence, the error signals are reconstructed in a better way with less loss of information. Here, the discrete wavelet transform (DWT) is used to select application dependent mother wavelet with scaling function and suitable wavelet function for removing noise. The signal is parameterized and expanded reliant on suitable wavelet function. This wavelet function decomposes and reconstructs the signal using dilated and shifted form of the wavelet function. Besides, wavelet function makes signal compact, orthogonal, linear in-phase, and with low approximation error, etc.[316]. This represents that the denoising should obtain modified wavelet coefficients which can depict the informative part of the signal. Hence, the minimal description length (MDL) data is used for the selection of the finest wavelet function [316]. While performing MDL, the wavelet denoising is considered as a model selection task, where the data model of the shortest description among all models is given by [316]:

$$MDL(k, n) = \min \left\{ \frac{3}{2} k \log N + \frac{N}{2} \log \left\| \hat{\epsilon}_n - \epsilon_n^{(k)} \right\|^2 \right\} \quad 0 \leq k < N; 1 \leq n \leq M \quad (3.3)$$

Here  $k, n$  denote the indices, integers  $N$  and  $M$  correspond to signal length and wavelet filters respectively, and  $\hat{\epsilon}_n$  is the wavelet vector. This wavelet vector and integer attained from the transformation of signal coefficients with the wavelet filter. Number of nonzero elements ( $k$ ) in wavelet vector defined by  $\epsilon_n^{(k)} = \theta^k, \hat{\epsilon}_n$ . The value of  $k$  sets to largest one in  $\hat{\epsilon}_n$ , while keeping other elements to zero and selects the optimized level using the MDL criterion [317].

### 3.1.1.2 Decomposition and Reconstruction

Wavelet basic function makes the DWT suitable for time response analysis, which causes minimum delay time as compared to other transform techniques as identified in [318]. The computational complexities of various WT techniques are depicted in Table 3.1 [318].

Table 3.1. The computational complexity of transform analysis

<b>Techniques</b>	<b>Computational Time</b>
Stationary wavelet transform [319]	0.0154
Hilbert–Huang transform [320]	0.2410
Continuous wavelet transform [321]	0.2415
Wigner Ville distribution [322]	0.0807
Discrete wavelet transforms	0.0049

The computation time of different WT techniques defines the superiority of using the DWT. Hence, the DWT is adapted as a signal processing technique for denoising the signal. Low-pass filter and high-pass filter are used in the cascaded form, one after another to realize the transformed signal, and further frequency dilation is performed by the down sampling method. The block diagram of a three-level discrete wavelet decomposition is shown in Figure 3.1.

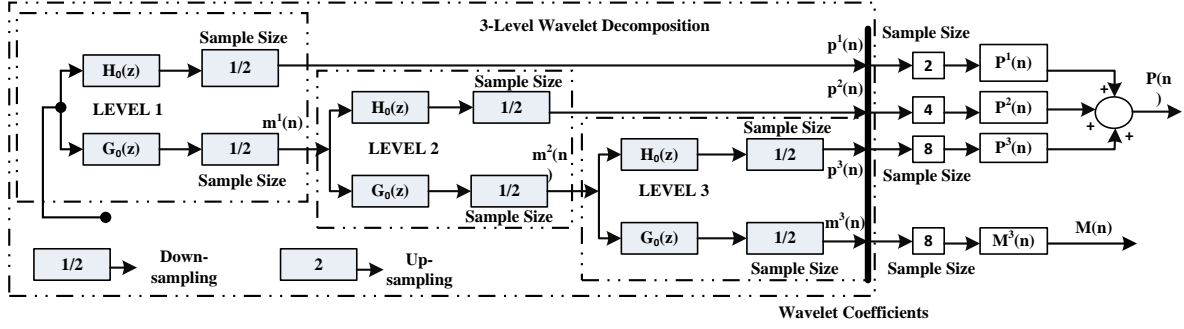


Figure 3.1. Three-level discrete wavelet decomposition

The high-pass filter output is the detailed signal coefficients at the first level of decomposition denoted by  $p^1$ . The output from the low pass filter is the approximation signal coefficients at the first level of decomposition is represented by  $m^1$ . The coefficients  $m^1$  and  $p^1$  establish the decomposition at first level and mathematically denoted as [313], [315]

$$m^1[n] = \sum_{k=0}^{N-1} g[k]x[n - k] \quad (3.4)$$

$$p^1[n] = \sum_{k=0}^{N-1} h[k]x[n - k] \quad (3.5)$$

Where  $h[k]$  and  $g[k]$  denote a finite set of coefficients that define the dilation and scaling functions respectively. At the first level of decomposition, this approximated coefficient  $m^1$  is delivered as input to both kinds of filters after the down-sampling of signals by two. Further, a 2-level high pass and low pass filter produce an approximation and detailed coefficients of length  $N/2$ . The second and third level in decomposition can be obtained by the following equations:

$$m^2[n] = \sum_{k=0}^{N/2-1} g[k]m^1[2n - k] \quad (3.6)$$

$$p^2[n] = \sum_{k=0}^{N/2-1} h[k]p^1[2n - k] \quad (3.7)$$

$$m^3[n] = \sum_{k=0}^{N/3-1} g[k]m^2[3n - k] \quad (3.8)$$

$$p^3[n] = \sum_{k=0}^{N/3-1} h[k]p^2[3n - k] \quad (3.9)$$

This process of filtration and down-sampling continues until the third level reached. Now, the denoising of signals can be stopped at the optimum level of decomposition. Once the noise is eliminated, and the modified coefficients are obtained, the signal quantization is performed to obtain a sampled signal truly digital and ready for control application. The difference between



the unquantized sample and the quantized output is called the quantization error. Once the signal is quantized, each input value of the quantization region is compared with the input value of a quantizer boundary, defined by the user [323]. The output depicts the zero-based index of the associated region. The zero-based input index values are transformed into quantized output values by decoding the encoded signal. The decoded signal helps in setting the output data type parameter to perform decomposition and reconstruction. Further, the signal attains its original form by reconstruction process which converts the discrete signal into the continuous signal. The signal is reconstructed using the low pass and high pass finite impulse response filters from high-frequency sub-band and low-frequency sub-band to remove noise present in the signal.

### 3.1.2 Wavelet Fuzzy Logic Controller

The design method of wavelet fuzzy controller is motivated at overcoming the drawbacks of transparent interpretation of choosing rules with the conventional fuzzy controllers. To begin with the development of the wavelet fuzzy controller, consider a sample window whose length is chosen based on the rising and falling edge of the reference. Further, the DWT decomposes the observed signals into detailed and approximate frequency components. The expected control actions are formed to design a wavelet-based controller due to this property of WT. Further, the decomposed coefficient support to create the desired control signal for the wavelet-based controller [324] which is expressed as:

$$u_w = k_{d^1}e_{d^1} + k_{d^2}e_{d^2} + \dots + k_{d^N}e_{d^N} + k_{a^N}e_{a^N} \quad (3.10)$$

Where  $e_{d^1}, e_{d^2}, \dots, e_{d^N}$  are the detailed components and  $e_{a^N}$  corresponds to the approximate components of the error signal, and  $k_{d^1}, k_{d^2}, \dots, k_{d^N}$ , and  $k_{a^N}$  tunes to the medium and high-frequency gains of the error signal [325].

Generally, the sensor noises associated with the servo units of robotic systems correspond to the high-frequency signals and the other external and internal disturbances associated with the signal are considered as the low-frequency signals. Therefore, the low-frequency gains associated with the error signal eliminate the low-frequency components and the high-frequency gains compensate for the high-frequency components [326]. Further, the quantised error signal is provided as an input to both the fuzzy logic controller to achieve the ball position control as shown in Figure 3.2.

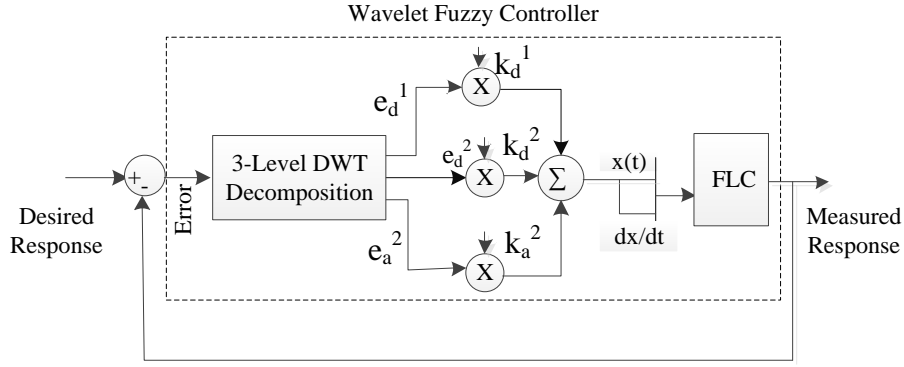


Figure 3.2. Three-level discrete wavelet transform based fuzzy logic controller

Internally, the design of the fuzzy logic controller is achieved by identifying the range of the input and output control variables. The fuzzy logic controller has two inputs and each input is subjected to carry linguistic variables expressed as *NB*-negative big, *NM*-negative medium, *NS*-negative small, *ZO*-zero, *PS*-positive small, *PM*-positive medium, and *PB* – positive big. These input variables are normalized with a universe of discourse using different membership functions. Similarly, the output variables which correspond to the scaling gains are formulated using the linguistic variables discussed above. Further, the rules are formulated between the inputs and outputs. Moreover, the weighted average method [327] can be adapted to compute the aggregate of each rule for fast and optimized output.

### 3.1.3 Stability Analysis

The scaling gains of the fuzzy controller are determined and updated by the action of the wavelet transform. The stability of the proposed controller depends on the performance of the optimized fuzzy logic control algorithm. Hence, a passivity approach can be used to determine the absolute stability of the fuzzy controller. This approach is based on some general characteristics of the input-output mapping of the controller and the input-output dynamics of the controlled system [328]. The mapping between the input and output of the fuzzy logic can be in general described by:

$$u_k = \frac{\sum_{i,j} \left[ \left( \mu_{E_i}(e_1) \cap \mu_{E_j}(e_2) \right) \cdot u_{n(i,j)} \right]}{\sum_{i,j} \left( \mu_{E_i}(e_1) \cap \mu_{E_j}(e_2) \right)} \quad (3.11)$$

where  $e_1$  corresponds to error  $e(k)$  at time  $k$ ,  $e_2$  corresponds to change in error  $de(k)$  at time  $k$ ,  $u_k$  represents the output of the fuzzy logic control,  $E_i$ ,  $E_j$  and  $u_{n(i,j)}$  are the linguistic variables of  $e_1$ ,  $e_2$  and  $u_k$  respectively, and  $\cap$  represents a fuzzy set intersection operator. Hence, the distinguishing characteristics of the fuzzy logic control in this work are:

- 1) In the fuzzy rule matrix, the control rules are antisymmetric about its off-diagonal (odd symmetry).
- 2) The fuzzy table numerical values increase/decrease gradually from top to bottom in a column and left to right in a row (monotony).
- 3) The central element of the rule table which corresponds to a control decision is usually zero and the elements surrounding the central area have small values.

These properties of the fuzzy controller reflect the consistency of operator control action and reflect the general properties of the system. Further, the numerical implications to analyse the stability of the controller are given by:

Consider  $e_1$  and  $e_2$  be two scalar inputs normalized to a range of  $[-L, L]$ . Hence,  $2N + 1$  input fuzzy sets are uniformly defined within this range with linguistic names as  $E_i, i = -N, \dots, -1, 0, 1, \dots, +N$ . The properties of these input membership functions (MFs) are:

- 1) The summation of the value of MFs is 1.

$$\sum_{i=-N}^N \mu_{E_i}(e) = 1 \quad (3.12)$$

- 2) For the value of input value exterior to the range  $[-L, L]$ ,

$$\begin{aligned} e > L &\Rightarrow \mu_{E_N}(e) = 1 \text{ and} \\ e < -L &\Rightarrow \mu_{E_{-N}}(e) = 1 \end{aligned} \quad (3.13)$$

- 3) The cover intervals  $E_i$  and  $E_{-i}$  which are symmetric about 0:

$$0 \leq \mu_{E_i}(-e) \leq 1 \quad (3.14)$$

- 4) Not more than two adjacent fuzzy sets are fired at a time for each input value with complimentary membership grades:

$$\text{if } |i - j| > 1 \Rightarrow \mu_{E_i}(e)\mu_{E_j}(e) = 0 \quad (3.15)$$

- 5) The inputs MFs fuzzy sets must be convex:

$$\mu_{E_i}[\lambda e + (1 - \lambda)e'] \geq \min[\mu_{E_i}(e), \mu_{E_i}(e')], \quad \forall e \neq e', \quad \forall \lambda \in [0, 1] \quad (3.16)$$

Further, the fuzzy rule base for the scalar inputs and output  $u$  is formed as:

$$\text{if } e_1 \text{ is } E_i \text{ and } e_2 \text{ is } E_j, \text{ then } u \text{ is } out_{f(i,j)}$$

where  $f(i, j)$  corresponds to a function whose value at  $i$  and  $j$  is an integer. This function relates the indices  $i$  and  $j$  of input fuzzy set to the index of output fuzzy set  $out_{f(i,j)}$  with center  $U_{f(i,j)}$ .

Hence with  $(2N + 1)^2$  fuzzy rules, the properties of the function  $f(i, j)$  are given by:

- 1) If any input condition is zero then the output must be zero:

$$f(0,0) = 0 \quad (3.17)$$

- 2) The control rules are odd symmetric about 0:

$$f(i, j) = -f(-i, -j), \quad \forall i, j \quad (3.18)$$

3) For a constant input, the output is a convex function:

$$j. [f(i, j) - f(i, 0)] \geq 0, \quad \forall i, j \geq 0 \quad (3.19)$$

$$i. [f(i, j) - f(0, j)] \geq 0, \quad \forall i, j \geq 0 \quad (3.20)$$

The central value of output MF is related to a property given as:

$$U_0 = 0, \quad U_i = -U_{-i} \text{ and } i \geq j \Rightarrow U_i \geq U_j \quad (3.21)$$

Moreover, the output  $u$  of the scalar controller is estimated by centre average defuzzification using min operator as follows:

$$u = \Phi(e_1, e_2) = \frac{\sum_{i,j} U_{f(i,j)} \cdot \min[\mu_{E_i}(e_1), \mu_{E_j}(e_2)]}{\sum_{i,j} \min[\mu_{E_i}(e_1), \mu_{E_j}(e_2)]} \quad (3.22)$$

Besides, the properties of input-output nonlinear mapping described based on the continuous-time Lipschitz function  $\Phi(.,.)$  are given as:

- Property a:  $|\Phi(e_1, e_2)| \leq u_M$  and  $M = \max_{i,j} U_{f(i,j)}$
- Property b:  $\Phi(0,0) = 0$  defines the steady-state condition.
- Property c:  $\Phi(e_1, e_2) = -\Phi(-e_1, -e_2)$  defines the odd symmetry
- Property d:  $\Phi(e_1, 0) \Rightarrow e_1 = 0$
- Property e:  $\Phi(.,.)$  corresponds to a sectorial function where for every scalar input there exists a solution  $\lambda', \gamma' > 0$  given by:

$$0 \leq e_1 \cdot [\Phi(e_1, e_2) - \Phi(0, e_2)] \leq \lambda' e_1^2 \quad (3.23)$$

$$0 \leq e_2 \cdot [\Phi(e_1, e_2) - \Phi(e_1, 0)] \leq \gamma' e_2^2 \quad (3.24)$$

where  $\lambda$  and  $\gamma$  are constants.

The proof for the above properties is detailed in [329].

Hence, a continuous-time system with state vector  $x \in R^n$ , input  $y(.): R \rightarrow R$ , and output  $u(.): R \rightarrow R$  is passive if there exists a real-valued storage function  $V(x)$  which is continuous nonnegative with  $V(0) = 0$  and supply rate  $W(y(\tau), u(\tau)) = y(\tau)u(\tau)$ , to the extent that the subsequent dissipation inequality holds  $\forall t > 0, u \in U, x(0) \in X$ :

$$V(x(t)) - V(x(0)) \leq \int_0^t W(y(\tau), u(\tau)) d\tau \quad (3.25)$$

where  $u$  is the control decision.

If the supply rate is:

$$W(y(\tau), u(\tau)) = y(\tau)u(\tau) - \varepsilon y(\tau)^2, \quad \varepsilon > 0 \quad (3.26)$$

Here, the input is a strictly passive system.

$$W(y(\tau), u(\tau)) = y(\tau)u(\tau) - \varepsilon u(\tau)^2, \quad \varepsilon > 0 \quad (3.27)$$

Here, the output is a strictly passive system.

$$W(y(\tau), u(\tau)) = y(\tau)u(\tau) - \varepsilon_1 y(\tau)^2 - \varepsilon_2 u(\tau)^2; \varepsilon_1, \varepsilon_2 > 0 \quad (3.28)$$

Here, the input, and output are strictly passive systems.

**Proof:** Consider the generalized representation of the continuous-time system driven by a fuzzy controller with state access:

$$\dot{x} = f(x) + G(x)u \quad (3.29)$$

$$\zeta = h(x) \quad (3.30)$$

$$\dot{y} = \zeta \quad (3.31)$$

$$u = \Phi(e_1, e_2) \quad (3.32)$$

$$e(t) = y_d - y(t) \quad (3.33)$$

where  $x \in X \subset R^n$ ,  $\zeta \in R$ ,  $u \in R$ , and  $f(x): X \rightarrow R^n$ ,  $f(0) = 0$ ,  $G(x): X \rightarrow R^n$ ,  $h(x): X \rightarrow R$ ,  $h(0) = 0$  are smooth functions,  $e_1$  corresponds to the error between measured and desired outputs  $y$  and  $y_d$  respectively,  $e_2$  is the rate of change of error  $e_1$ , and  $\Phi: R \times R \rightarrow R \in C^1$  corresponds to the control function. A basic assumption is made regarding the complete reachability and zero-state detectability of the system by following the above properties as follows:

$$u(t) = 0, \text{ and } \zeta(t) = 0 \Rightarrow \lim_{t \rightarrow \infty} x(t) = 0 \quad (3.34)$$

This assumption is obvious considering that the origin is an equilibrium point. Further, properties b and e lead to:

$$0 \leq e_1 \cdot \Phi(e_1, 0) \leq \lambda' e_1^2 \quad (3.35)$$

$$0 \leq e_2 \cdot \Phi(0, e_2) \leq \gamma' e_2^2 \quad (3.36)$$

Let

$$\Delta_{e_2}(e_1, e_2) = \Phi(e_1, e_2) - \Phi(e_1, 0) \quad (3.37)$$

$$\Delta_{e_1}(e_1, e_2) = \Phi(e_1, e_2) - \Phi(0, e_2) \quad (3.38)$$

This infers that

$$0 \leq e_2 \cdot \Delta_{e_2}(e_1, e_2) \leq \gamma' e_2^2 \quad (3.39)$$

$$0 \leq e_1 \cdot \Delta_{e_1}(e_1, e_2) \leq \lambda' e_1^2 \quad (3.40)$$

From the definition of passivity, the scalar control inputs result in

$$\int_0^t e_2(\tau) \cdot \Phi(e_1(\tau), e_2(\tau)) d\tau = \int_0^t \dot{e}_1(\tau) \cdot \Phi(e_1(\tau), 0) d\tau + \int_0^t e_2(\tau) \cdot \Delta_{e_2}(e_1(\tau), e_2(\tau)) d\tau \geq V[e_1(t)] - V[e_1(0)] \quad (3.41)$$

where

$$V[e_1(t)] - V[e_1(0)] = \int_0^t \dot{e}_1(\tau) \cdot \Phi(e_1(\tau), 0) d\tau = \int_{e_1(0)}^{e_1(t)} \Phi(e_1, 0) de_1 \quad (3.42)$$

This identifies that in a continuous domain the fuzzy controller is input-output passive stable.

#### 3.1.4 *Wavelet Fuzzy Control of Helicopter system*

The performance of the developed wavelet fuzzy controller on the helicopter system is assessed by comparing its operation with the action of conventional fuzzy logic controller in MATLAB/Simulink and Quarc real-time control integration. The details of modelling for the Real-time setup considered in this research are provided in Chapter 2. The complete setup is operated with an intel i3 quad core CPU with 3.6 GHz clock speed. In this setup, the main rotor lift is responsible for change in pitch, and pulling forces of tail rotor are responsible for yaw angle movements. The objective is to decouple the two typical output channels such that the output responses follows a desired trajectory. From the dynamic model of the 2DoF helicopter system it has been identified that the pitch and yaw axes can be controlled by appropriately selecting the voltages of the pitch and yaw motors, respectively. Due to the linear dependency of each rule on the input variables, the developed wavelet approach is considered to be ideal for acting as an interpolating supervisor for the fuzzy controller.

##### 3.1.4.1 Simulation Analysis

The wavelet fuzzy control of the helicopter model can be established by designing the controller for the uncoupled motion between the rotors of the horizontal and vertical axis. Further, a feedforward controller can be used to provide the pitch angle with the required gain. This forward control process allows a foremost signal to pass from the supply in the external control loop. This approach is directly related to the gain-based control. The compensation of output signal is achieved by feedforward control which includes the gravitational forces as well. These gravitational forces try to drag the pitch angle down due to gravity effect. Further, the huge variation in yaw and pitch error needed more control action when integrated with wavelet fuzzy controller. This variation continuously varies due to pitch and yaw setpoint difference, which further induced a large voltage that can saturate the power amplifier. The desired output can be disrupted due to this error mechanism and oscillations in system can reach to its extent. This condition can be avoided by adding pitch and yaw integrator with wavelet fuzzy. This integral anti-windup makes the saturation error negative when large voltage developed. The reset time divides this error and then sums at the integrator input. The complete block diagram is shown in Figures 3.3.

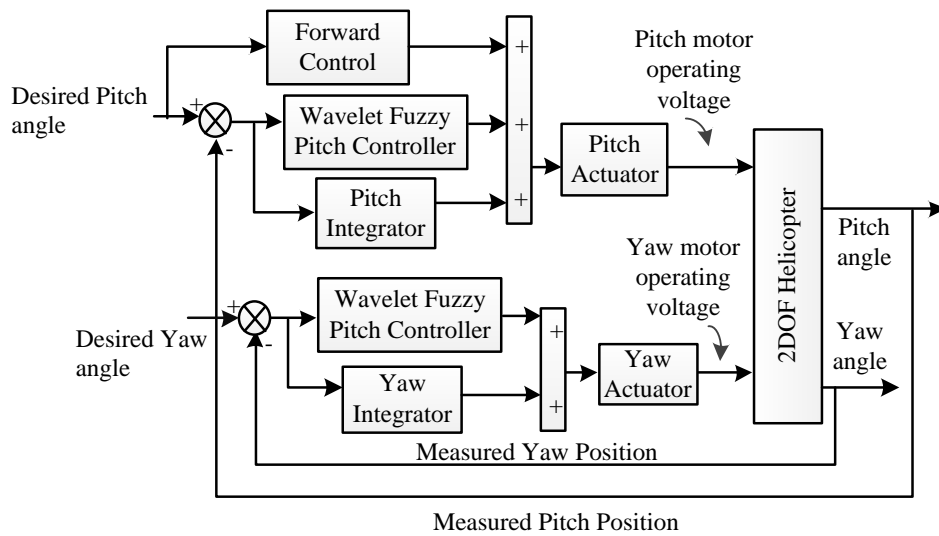
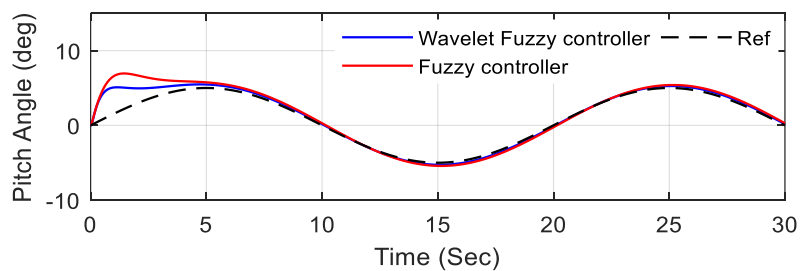
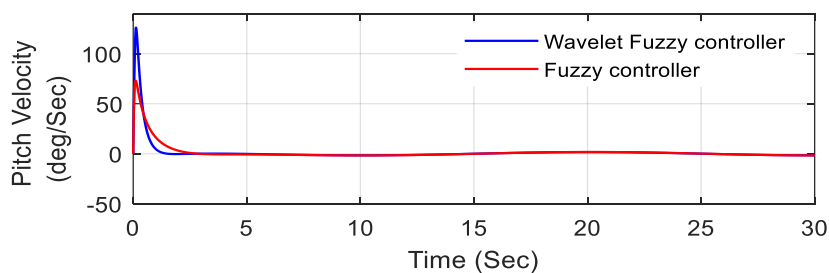


Figure 3.3. Wavelet fuzzy logic controller for helicopter control

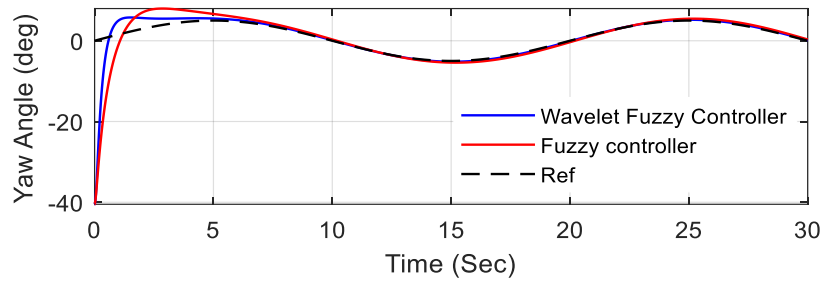
The Takagi-Sugeno type fuzzy inference system has been used for the pitch and yaw angle control purpose. Two different wavelet fuzzy controllers are used here for pitch and yaw angle control. First input is pitch error ( $e$ ), and the other one is the pitch derivative error ( $\dot{e}$ ) for pitch control and vice versa inputs are chosen for yaw control. Further, triangle membership function has been chosen for pitch error and derivative of pitch error while gaussian membership function has been selected for yaw error and derivative of yaw error. Seven linguistic variables have been chosen for every input variable named as  $NB$ ,  $NM$ ,  $NS$ ,  $ZO$ ,  $PS$ ,  $PM$ , and  $PB$ . Starting with the rules governing the uncoupled motion and then providing appropriate gains to the system, the Simulink results of the helicopter system are shown in Figures 3.4. To assess the performance of the developed approach, the results are compared with the action of conventional Fuzzy logic controller developed for the same helicopter setup.



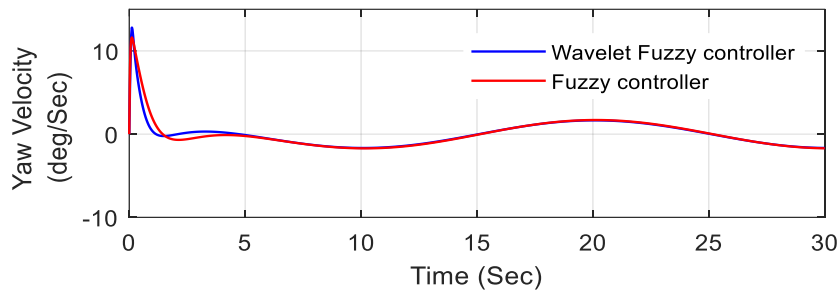
(a) pitch angle



(b) pitch velocity



(c) yaw angle



(d) yaw velocity

Figure 3.4. Pitch and yaw output with wavelet fuzzy control of helicopter in Simulink

In this simulation, the control objective is to maintain the system tracking the desired trajectories. From the Figure 3.4, the wavelet fuzzy control on the helicopter is efficiently able to track the desired reference trajectory. With the action of the conventional fuzzy controller, the pitch angle suddenly increases to 7.2 degrees. This generates a huge movement in the vertical position of the helicopter. Further, the pitch angle is diminished to 3.5 degree by controlling through wavelet fuzzy. Similarly, for horizontal motion, the angle variation by yaw is not sufficiently high during the conventional fuzzy control, which may cause a huge coupling misalignment between yaw and pitch motions. With the action of the wavelet fuzzy controller, necessary thrust force for rotation of blades of the helicopter is achieved by providing high initial gain. Further, to achieve proper coupling between the rotors, the pitch velocity is increased to 120 degrees/sec from 60 degrees/sec, and yaw velocity is varied from 12.6 degrees/sec to 13.3 degrees/sec by the action of wavelet fuzzy controller. The pitch and yaw motor deliver necessary torque and cross torque from the controlled gain generated by the controller. It clearly demonstrates the good performance and robustness property of the proposed controller in path tracking operations.

Further, the root mean square error (RMSE) has been calculated for the measured pitch angle with the reference sine wave trajectory as shown in Table 3.2 using Simulink. Besides, Table 3.3 points out a comparative assessment of controller's performance in terms of time response characteristics for pitch and yaw angle of helicopter system.



Table 3.2. Root mean square error for wavelet fuzzy control of helicopter in Simulink

Controller	Root mean square error	
	Pitch angle (deg)	Yaw angle (deg)
Fuzzy Controller	3.7129 deg	7.1271 deg
Wavelet Fuzzy	2.3117 deg	0.4735 deg

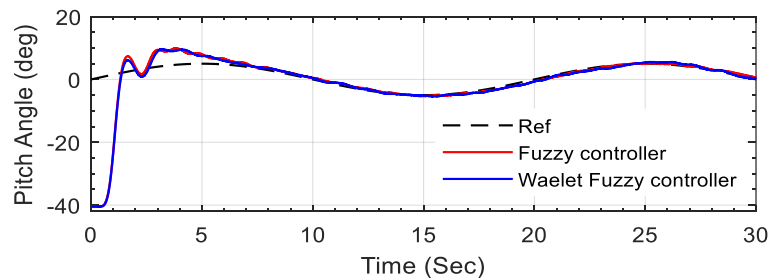
Table 3.3. Time response analysis for wavelet fuzzy control of helicopter in Simulink

Controller	Pitch Response		Yaw Response	
	Settling time ( $t_s$ ) (sec)	Steady-state error ( $e_{ss}$ ) (cm)	Settling time ( $t_s$ ) (sec)	Steady-state error ( $e_{ss}$ ) (cm)
Fuzzy Controller	5.3 sec	0.987 cm	7.17 sec	1.19 cm
Wavelet Fuzzy	4.13 sec	0.176 cm	5.31 sec	0.47 cm

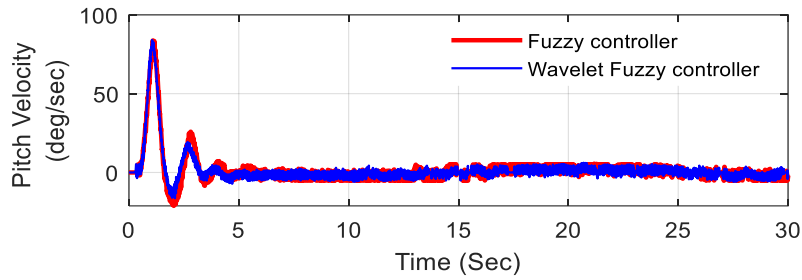
From the observation of the time response analysis, settling time  $t_s$  and steady-state error  $e_{ss}$  are minimum in case of wavelet fuzzy which are 4.13sec and 0.176cm for pitch control & 5.31sec and 0.47cm for yaw control.

#### 3.1.4.2 Real-time Analysis

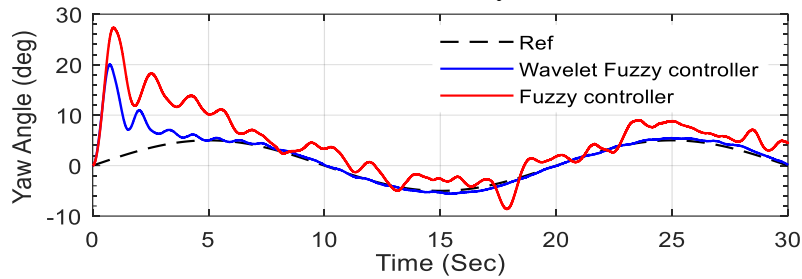
To analyse the action of the controller on the real-time operation of the helicopter system, the 2DoF helicopter system discussed in Chapter 2 is considered. As explained in simulation, the main objective is to track the helicopter to desired trajectories. Due to strong cross-coupling effect, the deviations of pitch characteristics affect both pitch and yaw parameters and so pitch angle, pitch velocity, yaw angle, and yaw velocity affected. The corresponding pitch and yaw characteristics for the real-time operation of the helicopter are shown in Figure3.5. Similar to the simulation process, the performance of the developed approach is assessed and the results are compared with the action of conventional Fuzzy logic controller developed for the same helicopter setup.



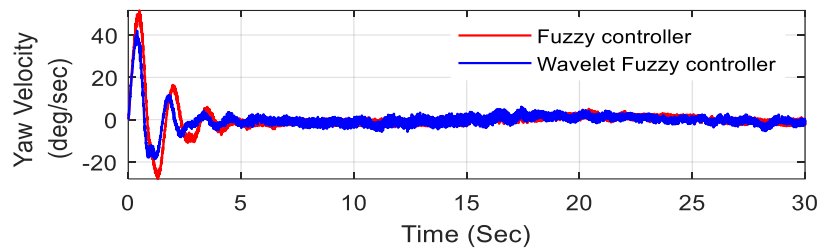
(a) Pitch angle



(b) Pitch velocity



(c) Yaw angle



(d) Yaw velocity

Figure 3.5. Pitch and yaw output with wavelet fuzzy control of helicopter in real-time

Further, from the results it can be identified that, for the action of the conventional controller on the system, high oscillation starts for pitch and yaw angle movement throughout the running conditions. Besides, the action of the wavelet fuzzy controller regulates the oscillations in pitch and yaw angle to the minimum value. Initially, the variation in oscillation with conventional fuzzy controller goes from 9.7 degrees to -4.3 degree, but with the action of the wavelet fuzzy controller, reduces the oscillations from 9.2 degree to -4.7 degree for pitch angle. For yaw angle, the oscillations with conventional fuzzy controller vary from 28 degree to -8.1 degrees, but with the action of the wavelet fuzzy controller, the oscillations are reduced from 20 degrees to -5.1 degrees. On the other side, with the conventional fuzzy controller, the pitch velocity is extremely low at the time of start and unable to give enough thrust to helicopter body and take time to settle the trajectory at the same time the yaw velocity should be low due to coupling effect. Further, while operating with the developed controller the yaw velocity is minimized to 42 degree/sec. The pitch and yaw motor provide the desired velocity from the controlled output. It clearly demonstrates the good performance and robustness property of the proposed controller in trajectory tracking operations in real-time as well.

Further, the RMSE for both the controllers is provided in Table 3.4. Besides, the settling time ( $t_s$ ) and steady-state error ( $e_{ss}$ ) are minimum in case of wavelet fuzzy controller which is 6.87sec, 2.413cm for pitch control & 5.5sec, 6.143cm respectively for yaw control as shown in Table 3.5.

Table 3.4. Root mean square error for wavelet fuzzy control of helicopter in real-time

Controller	Root mean square error	
	Pitch angle (deg)	Yaw angle (deg)
Fuzzy Controller	5.812 deg	9.51 deg
Wavelet Fuzzy	2.871 deg	6.381 deg

Table 3.5. Time response analysis for wavelet fuzzy control of helicopter in real-time

Controller	Pitch Response		Yaw Response	
	Settling time ( $t_s$ ) (sec)	Steady-state error ( $e_{ss}$ ) (cm)	Settling time ( $t_s$ ) (sec)	Steady-state error ( $e_{ss}$ ) (cm)
Fuzzy Controller	7.32 sec	3.763 cm	19.93 sec	12.31 cm
Wavelet Fuzzy	6.87 sec	2.413 cm	5.5 sec	6.143 cm

### 3.1.5 Wavelet Fuzzy Control of Ball Balancer system

The performance of the developed wavelet fuzzy controller on the ball balancer system is assessed by comparing its operation with the action of conventional fuzzy controller in MATLAB/Simulink and Quarc real-time control integration. The complete setup is operated with an intel i3 quad core CPU with 3.6 GHz clock speed. As the plate arrangement is symmetrical to the two servo units, the control action on one unit affects the operation of the control action on the other unit operating them in a coupled environment. To begin with, the system is manually calibrated by adjusting the servo load gear at 0 degrees. Further, the camera is calibrated at an offset height of 40 and a width parameter of 100 to ensure the flycap view of plate and ball and to capture an image of size  $440 \times 440$  [66]. Once the calibration is achieved, a square input signal is given to the setup with a frequency of 0.08Hz and an amplitude of 3. Further, a relative study of the fuzzy and wavelet fuzzy controllers has been done on Simulink and hardware for validation purposes.

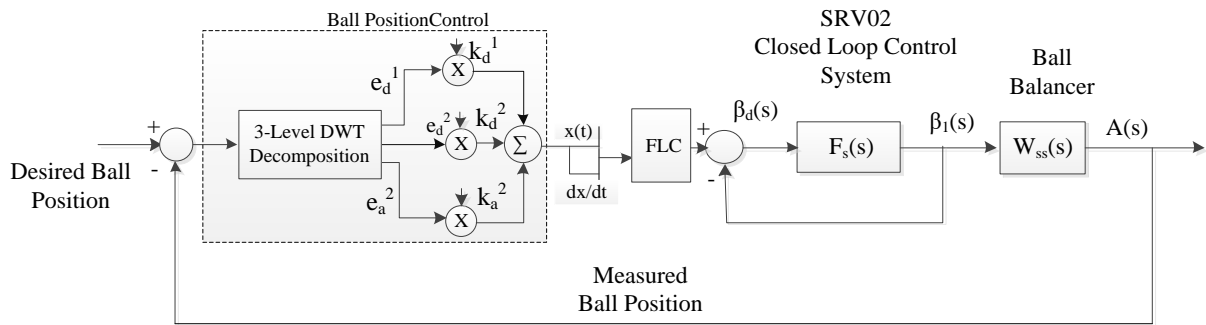
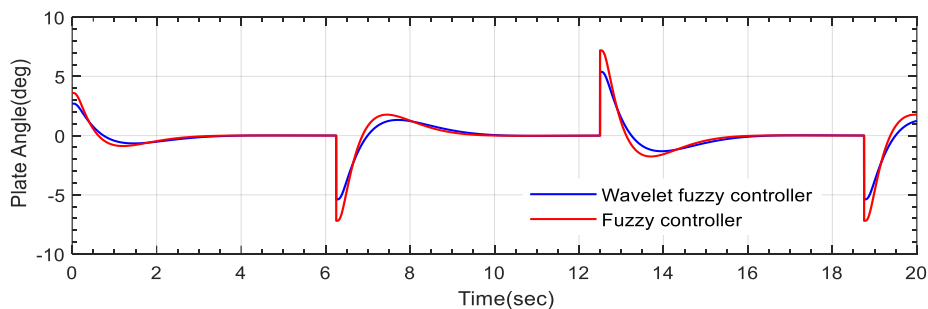


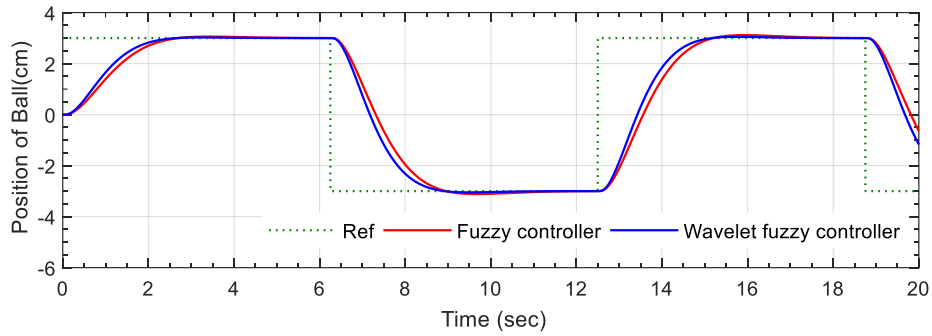
Figure 3.6. Wavelet fuzzy control for plate angle and ball position control

### 3.1.5.1 Simulation Analysis

In Simulink process, the contribution of WT make setup vulnerable to stable as system turn out to be unstable, this is ensured by time-domain analysis as the involvement of signal reconstruction filter out the unwanted frequency component from the signal and decrease the existing fluctuations. In this experiment, the number of wavelet filters ( $M$ ) are considered 12 due to the use three level decomposition and reconstruction, and the signal length ( $N$ ) is 1000 which is the sampling length of signal for one cycle. The results of the ball position, servo angle, and voltage of the ball balancer system for the action of improved wavelet fuzzy and the conventional fuzzy are shown in Figure 3.7. The main concentration is on the ball movement on the plate having minimum oscillations. From Simulink analysis of wavelet fuzzy, Figure 3.7(a), the plate angle of the x-axis is 2.7 degrees. This is less as compared to fuzzy with 3.8degree and causes the plate stabilization, and hence ball movement will be slow down and remains steady. The ball position from Figure 3.7(b) illustrates that discrepancy in the position of the ball during movement on plate surface will have less settling time in wavelet fuzzy will balance the ball rapidly.



(a) Plate angle



(b) Ball position

Figure 3.7. Ball balancer output with wavelet fuzzy control in Simulink

Table 3.6 shows the peak time, settling time, peak overshoot and steady-state error of wavelet fuzzy are 0.1sec, 2.42sec, 8.91e-06%, and 1.2e-06cm respectively which is less in comparison with fuzzy. As the peak time reduced, as well as the ball reaching time to its specified amplitude, minimized, reduction in settling time cause to balance the ball in less time as compared to fuzzy. As per the output waveform, it has been noticed that wavelet fuzzy control has a suitable response with the lowest error and overshoot carries the perfect ball balance on the plate without any vibrations. The RMSE value of the system output for wavelet fuzzy and fuzzy in terms of position and plate angle parameter using Simulink are shown in Table 3.7.

Table 3.6. Time response analysis for wavelet fuzzy control of ball balancer in Simulink

Controllers	Peak Time ( $t_p$ ) (sec)	Settling Time ( $t_s$ ) (sec)	Peak Overshoot ( $M_p$ ) (%)	Steady-state error ( $e_{ss}$ ) (cm)
Fuzzy controller	0.2sec	2.56sec	1.95%	0.0012cm
Wavelet fuzzy	0.1sec	2.42sec	8.91e - 06 %	1.2e - 06 cm

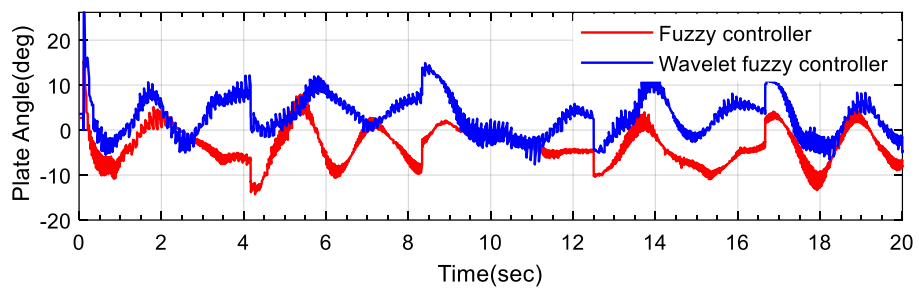
Table 3.7. Root mean square error for wavelet fuzzy control of ball balancer in Simulink

Controller	Root mean square error	
	Position (cm)	Angle (deg)
Fuzzy controller	2.1455 cm	3.3255 deg
Wavelet fuzzy	1.7586 cm	0.5931 deg

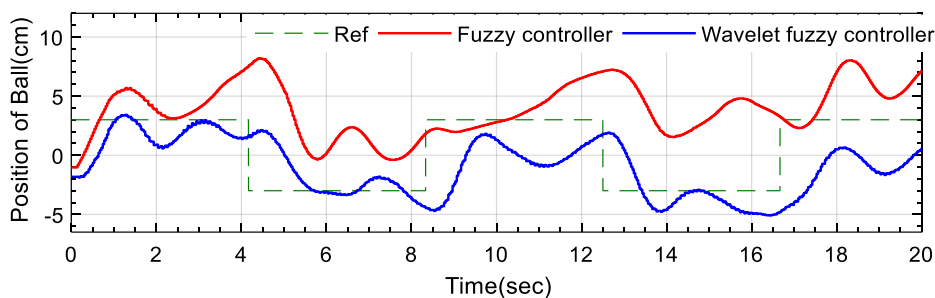
### 3.1.5.2 Real-time Results

The experimental setup discussed in Chapter 2 is used to assess the performance of the developed controller in real-time. Further, the controller is implemented with the real-time setup by performing the data exchange between the numerical simulation and the laboratory

setup. Initially, the simulated models in section 3.1.4.1 are calibrated for code generation with MATLAB/Simulink and Quanser software. Further, a C code generated in using the MEX function and interpreted to the hardware using the RS232 interface with the DAQ board. To achieve the efficient operation of the developed controller and simultaneous action of the ball balancer system for every rule fired by the fuzzy, Immediate I/O is adapted. This performs the hardware-in-the-loop-application programming interface (HIL-API) operation of the developed controller with the real-time setup. The results of the controller implementation in real-time are measured based on the momentary investigation. As the HIL-API provides an external interface for handling error known as the message API. This facilitates the interpretation of errors returned by different blocks and other APIs' functions. With this interface, the error between the simulation output and hardware results can be interpreted, and the corresponding control action can be activated to minimize the error. Further, based on the controller implementation and experimental procedure adapted, the ball position, plate angle, and operating voltage of the servo mechanism are observed as shown in Figure 3.8.



(a) Plate angle



(b) Ball position

Figure 3.8. Ball balancer output with wavelet fuzzy control in real-time

From the results in Figure 3.8(a) it is observed that the plate stabilization angle of wavelet fuzzy is maintained between  $-7^{\circ}$  to  $25^{\circ}$ , whereas the fuzzy has a stabilization angle varying between  $-15^{\circ}$  to  $+15$ . From Figure 3.8(b) it is observed that the wavelet fuzzy stabilizes the ball on the plate with zero oscillations even in the presence of external disturbances to achieve smooth operation. Further, the time response characteristics of the wavelet fuzzy and the fuzzy are

calculated for assessing the real-time operation of both the controllers and the corresponding results are shown in Table 3.8. The wavelet fuzzy response is observed to be better in terms of peak time, settling time, and peak overshoot, than the conventional fuzzy controller. Further, to support the performance assessment by the time response characteristics, the root mean square error (RMSE) is estimated for both the controllers in terms of position and plate angle parameter and tabulated in Table 3.9.

Table 3.8. Time response analysis for wavelet fuzzy control of ball balancer in real-time

<b>Controllers</b>	<b>Peak time</b> ( $t_p$ ) (sec)	<b>Peak overshoot</b> ( $M_p$ ) (%)	<b>Steady-state error</b> ( $e_{ss}$ ) (sec)
Fuzzy controller	1.98 sec	23.4%	4.1 cm
Wavelet fuzzy controller	1.83 sec	20.9%	1.55 cm

Table 3.9. Root mean square error for wavelet fuzzy control of ball balancer in real-time

<b>Controllers</b>	<b>Root mean square error</b>	
	Position (cm)	Angle (deg)
Fuzzy controller	5.1894 cm	3.4049 deg
Wavelet fuzzy controller	3.5680 cm	2.7517 deg

This development of wavelet fuzzy controller for ball balancer system calibrates the controller based on the error and change in error for the plate angle and ball position, and the control action is achieved by balancing the plate angle.

### 3.2 EVOLVING FUZZY BASED CONTROL SYSTEM

The fuzzy based approaches are developed to achieve trajectory tracking and position control with the helicopter and ball balancer system respectively. This minimizes the system error based on the inputs and the antecedent fuzzy sets of nonsingleton fuzzy logic controllers. But the inability of these techniques to perform while dealing with higher order uncertainties is considered as a major drawback. In another kind, the type-2 fuzzy logic controller (T2FLC) was widely adapted to cope up with the higher order uncertainties. Besides, the increasing fuzzy sets has improved the degree of freedom for T2FLC to handle uncertainties conveniently [126]. But the type reduction scenario made the general T2FLCs computationally intensive. In light of these issues, this section develops an evolving type-2 quantum fuzzy neural network (eT2QFNN) targeted at overcoming the drawbacks of conventional fuzzy systems and

achieving trajectory tracking and position control with the helicopter and ball balancer system respectively. The eT2QFNN has the advantage of overcoming the effects of parametric uncertainties in the systems with the help of a quantum rule growing scenario. Besides, the drawbacks of conventional control techniques with harmonic noises, and initializing weights for achieving trajectory tracking are overcome by the automatic generation of the rules in a single pass learning mode. Moreover, the use of quantum membership functions (QMF) tackles the problem of inadequacy and noise in conventional membership functions of fuzzy sets. Further, two learning policies are encompassed with the proposed approach, i.e., growing scenario and parameter adjustment, to deal with the drawbacks of self-organizing process in conventional fuzzy techniques.

### 3.2.1 Basic Control Scheme

#### 3.2.1.1 Proportional Derivative Controller

The use of proportional derivative (PD) controller with the helicopter and ball balancer systems is widely seen in the literature due to their design advantages in regulating the elevation, travel angles, and plate angles. Besides, the PD controller can be operated as a nonlinear feed forward controller which compensates for the gravitational torque in these systems. These advantages can also be seen with a PID controller, but the operation of these systems with the risk of integrator windup gives a large error between the measured and desired pitch and yaw angles. This results in large output voltages by the integrator which saturates the amplifier. Further, by the time the measured output reaches the desired response, the integrator builds up large energy causing large overshoot and oscillations in the response. Besides, the PD controller holds responsibility for the asymptotic stability of the helicopter and accommodates type 2 fuzzy neural network (T2FNN) with enough time for online learning of the system dynamics. The generalized representation of PD law for the control of a particular rotor model is described as

$$y_{PD} = K_P e + K_D \dot{e} \quad (3.43)$$

where  $e$ ,  $\dot{e}$  corresponds to the feedback error and its time derivative respectively between the desired and measured pitch or yaw angle value, and  $K_P$ ,  $K_D$  correspond to the proportional and derivative gains of the PD controller respectively. Further, while operating the helicopter and ball balancer systems with the PD controller, the disadvantages due to high control signal, and increasing accumulated error are observed. Besides, the differentiator leads to noise amplification which effects the operation of these systems and deviates them from following the reference. To overcome these drawbacks, the advantages of self-learning techniques are



identified to be widely applicable. As a part of the self-learning process, the interval type 2 fuzzy neural network (IT2FNN) techniques are adapted in this research.

### 3.2.1.2 Type-2 Fuzzy Neural Network (T2FNN)

The development of IT2FNN in this research adapts the antecedent type-2 and consequent crisp number (A2-C0) model [330]. The general structure of the A2-C0 fuzzy system for an  $i$   $j^{th}$  rule with two inputs is given as

$$\text{If } x_1 \text{ is } \tilde{A}_{1i} \text{ and } x_2 \text{ is } \tilde{A}_{2j}, \text{ Then } y_{NN} = \Omega_{ij} \quad \forall i = 1, \dots, I \text{ and } j = 1, \dots, J \quad (3.44)$$

where  $x_1$  and  $x_2$  corresponds to the input variables for the type-2 fuzzy,  $I$  and  $J$  correspond to the number of membership functions,  $\tilde{A}_{1i}$ ,  $\tilde{A}_{2j}$  are the fuzzy sets of IT2FNN and their corresponding membership functions are  $\tilde{\mu}_{1i}$ ,  $\tilde{\mu}_{2j}$ , respectively.  $y_{NN}$  defines the output variable of the fuzzy neural network and  $\Omega_{ij}$ 's are the adaptive crisp values. Further, the upper and the lower membership functions of  $\tilde{\mu}_{1i}$  are  $\bar{\mu}_{1i}$  and  $\underline{\mu}_{1i}$  and similarly for  $\tilde{\mu}_{2j}$  are  $\bar{\mu}_{2j}$  and  $\underline{\mu}_{2j}$ . The system output with IT2FNN can be written as [330]

$$y_{NN} = q \frac{\sum_{j=1}^J \sum_{i=1}^I \underline{w}_{ij} \Omega_{ij}}{\sum_{j=1}^J \sum_{i=1}^I \underline{w}_{ij}} + (1 - q) \frac{\sum_{j=1}^J \sum_{i=1}^I \bar{w}_{ij} \Omega_{ij}}{\sum_{j=1}^J \sum_{i=1}^I \bar{w}_{ij}} \quad (3.45)$$

where  $\underline{w}_{ij} = \underline{\mu}_{1i} \underline{\mu}_{2j}$ ,  $\bar{w}_{ij} = \bar{\mu}_{1i} \bar{\mu}_{2j}$ , and  $q$  reflects the weighting parameter which shares the contribution of the upper and the lower membership function.

### 3.2.2 Evolving type-2 quantum fuzzy neural network architecture

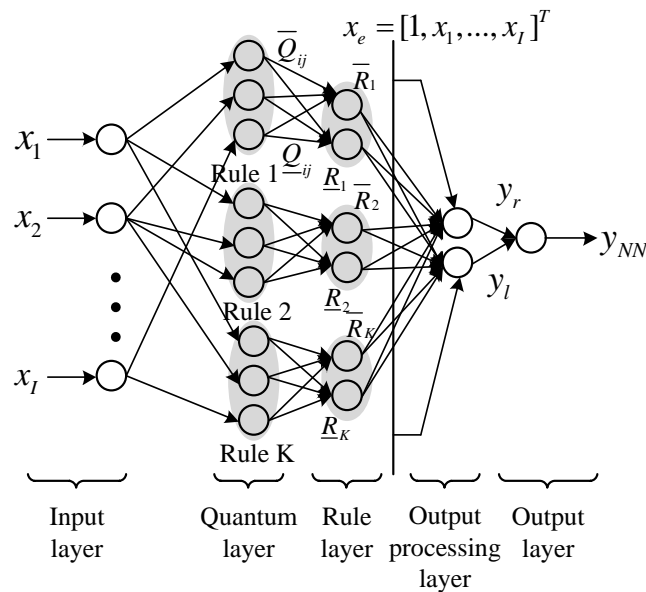


Figure 3.9. Architecture of evolving type-2 quantum fuzzy neural network [331]

In general, the most commonly used multi input single output IT2FLC structure has  $I$  inputs ( $x_1 \in X_1, \dots, x_I \in X_I$ ) and one output ( $y \in Y$ ) with  $K$  term nodes for each input. Further, any given property of fuzzy can be expressed with multiple fuzzy characteristic functions. This provides a choice for the expert system designer, based on the understanding about the problem which may sometimes lead to certain cases which has vague predicates. This can happen either due to the limited knowledge about the modelling of real-time system or due to insufficiency related to rule formulation. Considering these aspects, an arena of quantum computation is introduced in this research through the quantum fuzzy sets. The quantum fuzzy sets are defined as a sense superpositions of various fuzzy sets which can be seen either as an extension of quantum properties, or quantum predicates. This provides an advantage for development of a quantum fuzzy neural network approach which overcomes the drawbacks in conventional fuzzy sets. The complete architecture of evolving type-2 quantum fuzzy neural network (eT2QFNN) is shown in Figure 3.9. The layer 1 corresponds to the input layer representing the input nodes or variables. In layer 2, the quantum membership functions are defined for expressing the input variables as the fuzzy linguistic variables by calculating the quantum membership values. The layer 3 corresponds to a rule layer and the nodes in layer 3 are the compensatory rule nodes which are equated with the number of compensatory fuzzy sets of each external linguistic input variable. The links before and after layer 3 represent the preconditions and consequences of the of the rules respectively. The layer 4 is an output processing layer whose nodes are a linear function of the input variables. Generally, these nodes are known as the consequent nodes. Further, the last layer corresponds to an output layer and the nodes are defined as output nodes which acts as a defuzzifier and are recommended by layers 3 and 4. The rule premise comprises of interval type-2 quantum membership function (IT2QMF) as an antecedent to conventional IT2FNN and variable crisp numbers as the resultant part. For  $K$  term nodes, the fuzzy rules are expressed as follows:

$$R_j: \text{If } x_1 \text{ is } \tilde{Q}_{1j} \text{ and } x_i \text{ is } \tilde{Q}_{iK}, \text{ Then } y_j = X_e \tilde{\Omega}_j \quad (3.46)$$

where  $x_i$  is the  $i^{th}$  input,  $\tilde{Q}_{ij} = [\bar{Q}_{ij}, \underline{Q}_{ij}]$  is the set of upper and lower linguistic terms of quantum membership functions (QMFs),  $y_j$  is the regression output in  $j^{th}$  rule, and  $\tilde{\Omega}_j = [\bar{\Omega}_j, \underline{\Omega}_j]$  is the set of upper and lower weight parameters defined as  $\tilde{\Omega}_j \in \mathfrak{R}^{2(I+1)}$ . Here, the QMF is extended in to IT2FLC structure with uncertain jump functions for identifying uncertainties in the inputs and achieve efficient control. The output of IT2QMF for  $i^{th}$  input and  $j^{th}$  rule is given as:

$$\begin{aligned} \tilde{Q}_{ij}(x_{ij}, \beta, m_{ij}, \tilde{\theta}_{ij}) = & \frac{1}{n_s} \sum_{r=1}^{n_s} \left[ \left( \frac{1}{1 + \exp(-\beta x_i - m_{ij} + |\tilde{\theta}_{ij}^r|)} \right) U(x_i, -\infty, m_{ij}) + \right. \\ & \left. \left( \frac{\exp(-\beta(x_i - m_{ij} - |\tilde{\theta}_{ij}^r|))}{1 + \exp(-\beta(x_i - m_{ij} - |\tilde{\theta}_{ij}^r|))} \right) U(x_i, m_{ij}, \infty) \right] \end{aligned} \quad (3.47)$$

where  $m_{ij}$  is the mean of  $i^{th}$  input in  $j^{th}$  rule,  $\beta$  is the slope factor,  $\tilde{\theta}_{ij} = [\bar{\theta}_{ij}, \underline{\theta}_{ij}]$  is the uncertain jump position set defined as  $\tilde{\theta}_{ij} \in \mathfrak{R}^{2 \times I \times n_s \times K}$ , and  $n_s$  defines the number of grades. For any case, if the upper jump position ( $\bar{\theta}_{ij} = [\bar{\theta}_{1j}^1 \dots \bar{\theta}_{1j}^{n_s}; \dots; \bar{\theta}_{Ij}^1 \dots \bar{\theta}_{Ij}^{n_s}]$ ) is greater than the lower jump position ( $\underline{\theta}_{ij} = [\underline{\theta}_{1j}^1 \dots \underline{\theta}_{1j}^{n_s}; \underline{\theta}_{Ij}^1 \dots \underline{\theta}_{Ij}^{n_s}]$ ), which defines as  $\bar{\theta}_{ij} > \underline{\theta}_{ij}$ , the output of IT2QMF in eq. (3.47) leads to IT2 inference scheme producing footprint of uncertainties (FOU) [332]. Further, the union function  $U$  with  $U(x; a, b) = \begin{cases} 1, & \text{if } a \leq x_i < b \\ 0, & \text{otherwise} \end{cases}$  implies that  $x_i, m_{ij}, \infty$  have the properties of commutativity, associativity, monotonic, and continuous [333].

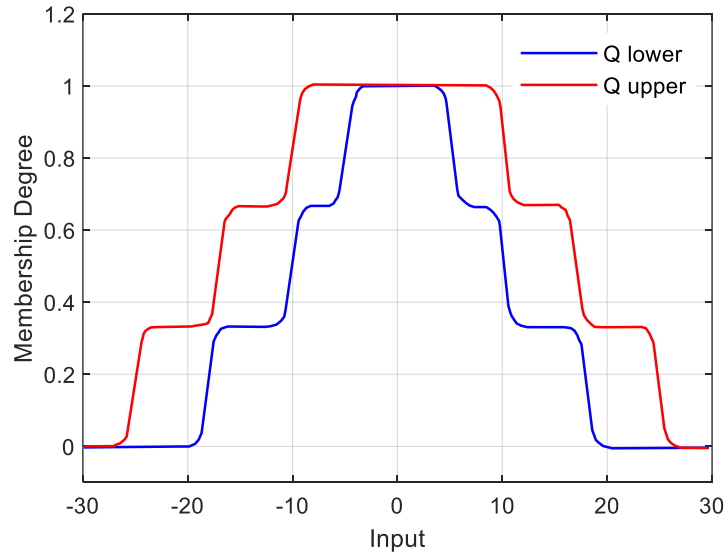


Figure 3.10. Interval type-2 quantum membership functions with 3 grades

Figure 3.10 depicts the IT2QMF with 3 grades depicting the FOU. A detailed operation of each layer in evolving type-2 quantum fuzzy neural network is discussed in detail as follows:

**Layer 1 (Input layer):** The input signals are fed into this layer where the data is directly propagated into the next layer without performing any computation. For an  $n^{th}$  observation, the input  $x_i$  is defined as  $X_n \in \mathfrak{R}^{1 \times I}$  and the output is defined as:

$$u_i = x_i \quad (3.48)$$

**Layer 2 (Quantum layer):** In this layer, the fuzzification process is performed. The QMFs calculate the degree of membership for  $X_n$  in each rule. The total number of rules initially defined are denoted as  $K$ . The outputs of the quantum layer are mathematically obtained as:

$$\bar{Q}_{ij} = \tilde{Q}_{ij}(x_i, \beta, m_{ij}, \bar{\theta}_{ij}) \quad (3.49)$$

$$\underline{Q}_{ij} = \tilde{Q}_{ij}(x_i, \beta, m_{ij}, \underline{\theta}_{ij}) \quad (3.50)$$

Layer 3 (*Rule layer*): This layer calculates the spatial firing strength by combining the membership degrees of  $j^{th}$  rule which is given as follows:

$$\tilde{R}_j = [\bar{R}_j, \underline{R}_j] \quad (3.51)$$

where  $\tilde{R}_j$  is the spatial firing strength,  $\bar{R}_j = [\bar{R}_1 \dots \bar{R}_K]$  is the set upper firing strength, and  $\underline{R}_j = [\underline{R}_1 \dots \underline{R}_K]$  is the set of lower firing strength. The product T-norm is adapted with the IT2QMF as given in (3.52) and (3.53) for performing this operation.

$$\bar{R}_j = \prod_{i=1}^l \bar{Q}_{ij} \quad (3.52)$$

$$\underline{R}_j = \prod_{i=1}^l \underline{Q}_{ij} \quad (3.53)$$

Layer 4 (*Output processing layer*): The upper and lower crisp outputs, i.e.  $y_r$  and  $y_l$  respectively are calculated in this layer. The type reduction process is carried out by employing design factors  $[q_r, q_l]$  for converting IT2 variables in to type 1 variable. The design factors control the upper and lower membership function proportion in such a way that  $q_r > q_l$ . This process of type reduction requires lesser epochs when compared to conventional type reduction techniques [334]. Here, the decoupled extended Kalman filter [334] is used to adjust the design factor such that the upper and lower crisp outputs keep adapting to the changes in the inputs.

The upper and lower crisp outputs are calculated as:

$$y_r = \frac{(1-q_r) \sum_{j=1}^K \underline{R}_j \bar{\Omega}_j x_e^T + q_r \sum_{j=1}^K \bar{R}_j \bar{\Omega}_j x_e^T}{\sum_{j=1}^K (\bar{R}_j + \underline{R}_j)} \quad (3.54)$$

$$y_l = \frac{(1-q_l) \sum_{j=1}^K \underline{R}_j \underline{\Omega}_j x_e^T + q_l \sum_{j=1}^K \bar{R}_j \underline{\Omega}_j x_e^T}{\sum_{j=1}^K (\bar{R}_j + \underline{R}_j)} \quad (3.55)$$

where  $\bar{\Omega}_j = [\bar{w}_{ij}, \dots, \bar{w}_{(l+1)j}]$  is the weight parameter of upper layer for  $j^{th}$  rule,  $\underline{\Omega}_j = [\underline{w}_{ij}, \dots, \underline{w}_{(l+1)j}]$  is the weight parameter of lower layer for  $j^{th}$  rule, and  $x_e = [1, x_1, \dots, x_l]$  is the extended input vector. The 1 in extended input prevents untypical gradient [335] and intercepts the rule consequent.

Layer 5 (*Output layer*): This layer calculates the crisp output of the network by summing the upper and lower crisp outputs obtained at the output processing layer as follows:

$$y_{NN} = y_r + y_l \quad (3.56)$$

### 3.2.2.1 Learning algorithm

The eT2QFNN works on an online learning policy presented in algorithm 1. The learning mechanism consists of two scenarios, the rule growing criteria and the parameter update scenario. These scenarios are executed for every iteration.

#### 3.2.2.1.1 Growing Criterion

Initially, the learning process has an empty rule base and the network structure keeps on updating as the input data changes. This eT2QFNN automatically evolves its fuzzy rules by forming a hypothetical rule from a newly obtained data sample. The hypothetically formed rules needs to be significantly evolved before adding it to the network structure. For a  $j^{th}$  rule, the significance is evaluated using  $L^2 - norm$  weighted by input density function  $p(x)$  as follows:

$$E_{sig}(j) = \|\omega_j\| \left( \int_{R^I} \exp\left(-\frac{2\|X-m_j\|^2}{\sigma_j^2}\right) p(X) dX \right)^{1/2} \quad (3.57)$$

From (3.57) it can be identified that the significance of the new rule is greatly dependent on  $p(X)$ , but this is hard to calculate in priori for practical problems as the data distribution is unknown. To overcome this a Gaussian mixture model (GMM) is formulated by Chellappa *et al.* [336]. This approximates the complex changes in data as

$$\hat{p}(X) = \sum_{h=1}^H \alpha_h \mathcal{N}(X; v_h, \Sigma_h) \quad (3.58)$$

where  $H$  is the number of mixture model,  $\alpha_h$  is the mixing coefficient,  $\mathcal{N}(X; v_h, \Sigma_h) = \exp\left(-\frac{1}{2}(X - v_h)^T \Sigma_h^{-1} (X - v_h)\right)$  is the Gaussian function of  $X$  with  $v_h \in \mathfrak{R}^I$  as mean vector and  $\Sigma_h \in \mathfrak{R}^{I \times I}$  as variance matrix.

Further, the  $j^{th}$  rule significance is estimated by substituting (3.58) to (3.57) and solving the standard operations to yield the following result:

$$\hat{E}_{sig}(j) = \|\omega_j\| \left( \pi^{\frac{I}{2}} \det(\Sigma_j)^{\frac{1}{2}} N_j A^T \right)^{1/2} \quad (3.59)$$

where  $\Sigma_j = \text{diag}(\sigma_{1,j}^2, \dots, \sigma_{I,j}^2)$  is the positive definite weighting matrix,  $A = [\alpha_1, \dots, \alpha_H]$  is the GMM mixing coefficients vector and  $N_j$  is further represented as:

$$N_j = \left[ \mathcal{N}\left(m_j - v_1; 0, \frac{\Sigma_j}{2} + \Sigma_1\right), \mathcal{N}\left(m_j - v_2; 0, \frac{\Sigma_j}{2} + \Sigma_2\right), \dots, \mathcal{N}\left(m_j - v_H; 0, \frac{\Sigma_j}{2} + \Sigma_H\right) \right] \quad (3.60)$$

Here,  $m_j = [m_{1,j}, \dots, m_{I,j}]$  is the mean vector of  $j^{th}$  rule.

Further, to improve the estimating accuracy of the IT2QMF, the interval type-2 Gaussian membership functions (IT2GMF) are adapted as follows:

$$\tilde{Q}_{i,j}(x_i, \beta, m_{i,j}, \tilde{\theta}_{i,j}) \approx \tilde{\mu}_{i,j} = \exp\left(-\frac{(x_i - m_{i,j})^2}{\tilde{\sigma}_{i,j}}\right) \quad (3.61)$$

$$\begin{aligned} \tilde{\sigma}_{i,j} &= [\underline{\sigma}_{i,j}, \bar{\sigma}_{i,j}] \\ \underline{\sigma}_{i,j} &= \min \underline{\theta}_{i,j}; \quad \bar{\sigma}_{i,j} = \min \bar{\theta}_{i,j} \end{aligned} \quad (3.62)$$

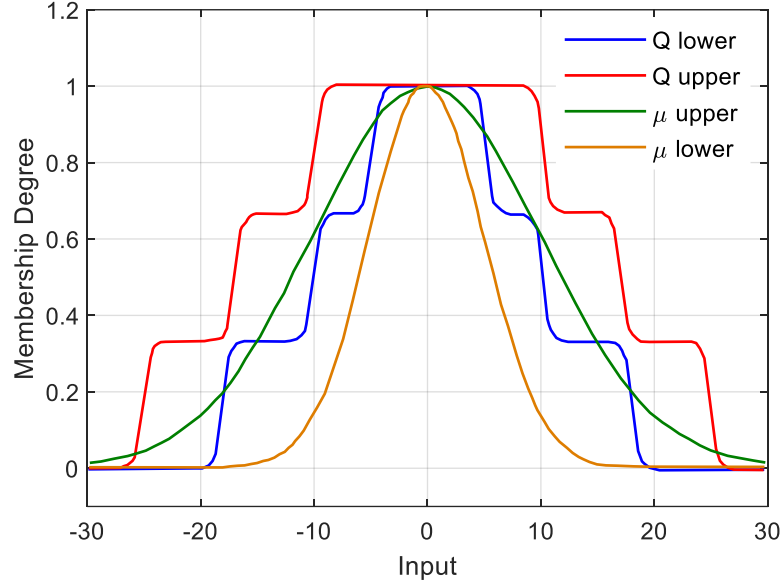


Figure 3.11. Comparison of interval type-2 quantum membership functions and Gaussian membership functions

In this process, both the means of IT2GMF and IT2QMF are defined equal and the minimum value of  $\tilde{\theta}_{i,j}$  is used to define the width of upper and lower IT2GMF. The illustration of IT2GMF in estimating the significance of IT2QMF is given in Figure 3.11. By following the above discussed criteria, it can be observed that the IT2GMF fits within the area of IT2QMF resulting in good approximation for IT2QMF.

By applying the modified IT2QMF estimation, the rule significance can be estimated using (3.63) and the design factor given by:

$$\hat{E}_j = |\hat{E}_{j,r}| + |\hat{E}_{j,l}| \quad (3.63)$$

This adds the hypothetical rule as a new rule  $R_{K+1}$  to the network if it contributes over the existing rules. The rule growing criteria is mathematically given as:

$$E_{K+1} \geq \rho \sum_{j=1}^K E_j \quad (3.64)$$

where  $\rho \in (0,1]$ , is a vigilance parameter [337].

### 3.2.2.1.2 Parameter Update

The process of adding a hypothetical rule to a network structure is based of two strategies. 1) If the hypothetical rule satisfies the conditions of network structure. This process is call initialization of fuzzy rule. 2) If the hypothetical rule doesn't satisfy the conditions of network

structure. Here, the parameters of the network are adjusting according to the change in input data. This strategy is winning rule update strategy. A detailed overview of these strategies is given as follows:

#### A. Initializing Fuzzy Rules:

The growing criterion in the previous section estimated the significance of a hypothetical rule according to the change in input data. For a data at  $n^{th}$  time step  $X_n$ , a new mean is assigned to IT2QMF as

$$m_{K+1} = X_n \quad (3.65)$$

This resulted in a new jump position which is achieved by distance-based formulation given in [331] as:

$$\begin{aligned} \bar{\theta}_{i,K+1} &= \frac{1}{\left(\frac{n_s+1}{2}\right)} \cdot r \cdot \bar{\sigma}_{i,K+1}, \\ \underline{\theta}_{i,K+1} &= \frac{1}{\left(\frac{n_s+1}{2}\right)} \cdot r \cdot \underline{\sigma}_{i,K+1} \end{aligned} \quad (3.66)$$

But, the application of mixed mean of GMM  $\hat{v}$  modified the distance, and the new distance is given as:

$$\begin{aligned} \bar{\sigma}_{i,K+1} &= |X_n - \hat{v}|, \underline{\sigma}_{i,K+1} = \delta_1 \cdot \bar{\sigma}_{i,K+1} \\ \hat{v} &= \sum_{h=1}^H \alpha_h \cdot v_h \end{aligned} \quad (3.67)$$

The updated jump positions are given as follows:

$$\begin{aligned} \bar{\theta}_{i,1} &= \frac{1}{\left(\frac{n_s+1}{2}\right)} \cdot r \cdot \bar{\sigma}_{i,1}, \\ \underline{\theta}_{i,1} &= \frac{1}{\left(\frac{n_s+1}{2}\right)} \cdot r \cdot \underline{\sigma}_{i,1} \end{aligned} \quad (3.68)$$

As the GMM approximates the mean and variance for complex changes in inputs, the eT2QFNN uses the diagonal positive definite matrix of mixed variance as follows:

$$\begin{aligned} \bar{\sigma}_{i,1} &= \hat{\sigma}_i, \underline{\sigma}_{i,1} = \delta_1 \bar{\sigma}_{i,1} \\ \hat{\Sigma} &= \sum_{h=1}^H \Sigma_h \cdot v_h, \hat{\Sigma} = \text{diag}(\hat{\sigma}_1^2, \dots, \hat{\sigma}_I^2) \end{aligned} \quad (3.69)$$

where,  $\delta_1$  creates the footprint of uncertainty (FOU).

Here, the consequent rule parameters of the hypothetical rule are like the winning rule as given by (3.70). The process of obtaining winning rule is discussed in further sections. Finally, as per (3.64) if the hypothetical rule satisfies the condition of the network structure, a new rule  $R_{K+1}$  is added and its covariance matrix is formulated as in (3.71). Further, to accommodate new rules, the covariance matrix needs to be adjusted [338], which is achieved by multiplying  $\left(\frac{K^2+1}{K^2}\right)$  with the covariance matrices of other rules as given in (3.72).

$$\tilde{\Omega}_{K+1} = \tilde{\Omega}_{jw} \quad (3.70)$$

$$P_{K+1}(n) = I_{Z \times Z} \quad (3.71)$$

$$P_j(n) = \left( \frac{K^2+1}{K^2} \right) P_j(n-1) \quad (3.72)$$

*B. Winning rule Update:*

If the hypothetical rules don't satisfy the conditions of network in (3.64), the performance of eT2QFNN needs to be maintained by adjusting the network parameters according to the changes in input data. This is achieved by formulating a winning rule which is dependent on the highest spatial firing strength. The spatial firing strength assesses the satisfaction rate of the rule antecedent part according to changes in the input data. The mathematical formulation for winning rule update is given as follows:

$$j_w = \arg \max_{j=1, \dots, K} \hat{P}(R_j|X) \quad (3.73)$$

$$\tilde{R}_j = (\bar{R}_j + \underline{R}_j)/2 \quad (3.74)$$

Previously, the decoupled extended Kalman filter is used to adjust the design factor such that the upper and lower crisp outputs keep adapting to the changes in the inputs. This helps in grouping the local parameters which results in formation of a block diagonal covariance matrix  $P(n)$  (3.75).

$$\tilde{P}(n) = \begin{bmatrix} \mathbf{P}_1(n) & \dots & \mathbf{0} & \dots & \mathbf{0} \\ \vdots & \ddots & & & \vdots \\ \mathbf{0} & & \mathbf{P}_j(n) & & \mathbf{0} \\ \vdots & & & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{0} & \dots & \mathbf{P}_K(n) \end{bmatrix} \quad (3.75)$$

For every time step, only one block covariance matrix is updated i.e.,  $P_{j_w}(n)$ . This property of localizing parameters enhances the efficiency by maintaining the robustness of the system. Remaining mathematics corresponding to decoupled extended Kalman filter are given in (3.76)- (3.78).

$$G_{j_w}(n) = P_{j_w}(n-1)H_{j_w}(n)[\eta I_{M \times M} + H_{k_w}^T(n)P_{j_w}(n-1)H_{j_w}(n)]^{-1} \quad (3.76)$$

$$P_{j_w}(n) = [I_{Z \times Z} - G_{j_w}(n)H_{j_w}^T(n)]P_{j_w}(n-1) \quad (3.77)$$

$$\vec{\theta}_{j_w}(n) = \vec{\theta}_{j_w}(n-1) + G_{j_w}(n)[t(n) - y(n)] \quad (3.78)$$

where,  $G_{j_w}(n)$  is the Kalman gain matrix,  $H_{k_w}$  is the Jacobian matrix [339],  $\vec{\theta}_{j_w}(n)$  is the parameter vector at  $n^{th}$  iteration.

---

**Algorithm 3.1:** eT2QFNN Learning Policy

---

**Define:** inputs- $X_n = [x_1, \dots, x_I]^T$ , outputs- $T_n = [t_1, \dots, t_M]^T$ , and  $n_s$

**\\Learning Policy\\**

**\\Phase 1:Growing Mechanism\\**

---



---

**If  $K = 0$  then**

Begin first rule using (3.65), (3.68), and (3.69)

**else**

The existing IT2QMF is approximated using (3.61)

Begin hypothetical rule  $R_{K+1}$  using (3.54-3.56), (3.59)

**for  $j = 1$  to  $K + 1$**

Calculate the rule significance  $\hat{E}_j$  using (3.63)

**end for**

**If  $E_{K+1} \geq \rho \sum_{j=1}^K E_j$  then**

$K = K + 1$

**end if**

**end if**

**\\Phase 2: Parameter Adjustment \\**

**If  $K(n) = K(n - 1)$  then**

Determine the spatial firing using (3.52) and (3.53)

Identify the winning rule  $j_w$  using (3.73)

Perform decoupled extended Kalman filter on  $R_{j_w}$  using (3.76) and (3.78)

Update covariance matrix according to the winning rule using (3.77)

**else**

Check the consequent weight  $\tilde{\Omega}_{K+1}$  of new rule using (3.70)

Calculate the covariance matrix using (3.71)

**for  $j = 1$  to  $K - 1$  do**

$$P_j(n) = P_j(n - 1) \left( \frac{K^2 + 1}{K^2} \right)$$

**end for**

**end if**

---

### 3.2.2.2 Adaptive law

To compensate the nonlinearity of the robotic systems operating with the PD controller, the eT2QFNN developed in Section 3.2.2 is trained with a nonlinear regulator. In this research, a sliding theory-based adaption law is equipped to develop a sliding surface  $s_p(e, \dot{e})$  as

$$s_p(e, \dot{e}) = \dot{e} + \chi e, \chi > 0 \quad (3.79)$$

The design process of the control structured is summarized with a theorem as follows:

**Theorem:** The consequent part parameters of the eT2QFNN with the adaptation laws are chosen as:

$$\dot{\mathcal{F}}(t) = -K_D \mathcal{P}(t) \tilde{\mathcal{W}}(t) s_p(t), \mathcal{F}(0) = \mathcal{F}_0 \in R^{n \times 1} \quad (3.80)$$

where  $P(t)$  is recursively updated as

$$\dot{\mathcal{P}}(t) = -\mathcal{P}(t) \tilde{\mathcal{W}}(t) \tilde{\mathcal{W}}^T(t) \mathcal{P}(t), \mathcal{P}(0) = \mathcal{P}_0 \in R^{n \times n} \quad (3.81)$$

In a compact set where the system under control is of second-order, the sliding theory-based adaption law guarantees the stability of the learning process.

**Proof:** The proof of the above state is initiated by identifying the control signals associated with the systems. The output of the PD controller from (3.43) is given as

$$y_{PD}(t) = K_P e(t) + K_D \dot{e}(t) \quad (3.82)$$

and can be rewritten as follows:

$$y_{PD}(t) = K_D \left( \dot{e}(t) + \frac{K_P}{K_D} e(t) \right) \quad (3.83)$$

Further, the overall control action is defined by

$$Y = -y_{NN}(t) + y_{PD}(t) \quad (3.84)$$

The representation of the plant model with sliding surface in (3.79) is given by

$$s_P(t) = \dot{e}(t) + \chi e(t) \quad (3.85)$$

where  $\chi = \frac{K_P}{K_D}$ . Considering (3.82) and (3.85), the control output of PD is defined as

$$y_{PD}(t) = K_D s_P(t) \quad (3.86)$$

Furthermore, the control output of eT2QFNN from (3.56) is summarized as

$$y_{NN}(t) = (1 - q_r) \sum_{j=1}^K \tilde{\tilde{R}}_j \bar{\Omega}_j + (1 - q_l) \sum_{j=1}^K \tilde{R}_j \underline{\Omega}_j \quad (3.87)$$

and rewritten as

$$y_{NN}(t) = \sum_{j=1}^K \left( (1 - q_r) \tilde{\tilde{R}}_j \bar{\Omega}_j + (1 - q_l) \tilde{R}_j \underline{\Omega}_j \right) \quad (3.88)$$

Further,  $\tilde{\mathcal{W}}(t)$  and  $\mathcal{F}(t)$  as defined as

$$\tilde{\mathcal{W}}(t) = \left[ (1 - q_r) \tilde{\tilde{R}}_1(t) + (1 - q_l) \tilde{R}_1(t) \dots (1 - q_r) \tilde{\tilde{R}}_K(t) + (1 - q_l) \tilde{R}_K(t) \right]^T \quad (3.89)$$

and

$$\mathcal{F}(t) = \left[ \bar{\Omega}_1 + \underline{\Omega}_1 \dots \bar{\Omega}_K + \underline{\Omega}_K \right]^T \quad (3.90)$$

From (3.89) and (3.90), (3.88) can be summarized as

$$y_{NN}(t) = \mathcal{F}^T(t) \tilde{\mathcal{W}}(t) \quad (3.91)$$

Hence, the cost function with the adaption law can be defined as

$$\mathcal{J}(t) = \int_0^t s_P^2(\tau) d\tau \quad (3.92)$$

$$= \frac{1}{K_D^2} \int_0^t (y(\tau) + y_{NN}(\tau))^2 d\tau \quad (3.93)$$

$$= \int_0^t \left( y(\tau) + \mathcal{F}^T(t) \tilde{\mathcal{W}}(\tau) \right)^2 d\tau \quad (3.94)$$

Considering the formulations in [340], the gradient of cost function  $\mathcal{J}$  concerning  $\mathcal{F}$  is derived as

$$\nabla_{\mathcal{F}} J(t) = 0 \quad (3.95)$$

$$\Rightarrow \int_0^t \tilde{\mathcal{W}}(\tau) y(\tau) d\tau + \mathcal{F}(t) \int_0^t \tilde{\mathcal{W}}(\tau) \tilde{\mathcal{W}}^T(\tau) d\tau = 0 \quad (3.96)$$

$$\Rightarrow \mathcal{F}(t) = - \left[ \int_0^t \tilde{\mathcal{W}}(\tau) \tilde{\mathcal{W}}^T(\tau) d\tau \right]^{-1} \int_0^t \tilde{\mathcal{W}}(\tau) y(\tau) d\tau \quad (3.97)$$

Further,  $\mathcal{P}$  is defined as

$$\mathcal{P}(t) = \left[ \int_0^t \tilde{\mathcal{W}}(\tau) \tilde{\mathcal{W}}^T(\tau) d\tau \right]^{-1} \quad (3.98)$$

$$\Rightarrow \mathcal{P}^{-1}(t) \mathcal{F}(t) = - \int_0^t \tilde{\mathcal{W}}(\tau) y(\tau) d\tau \quad (3.99)$$

$$\Rightarrow \mathcal{P}^{-1}(t) = \int_0^t \tilde{\mathcal{W}}(\tau) \tilde{\mathcal{W}}^T(\tau) d\tau \quad (3.100)$$

$$\Rightarrow \mathcal{P}^{-1}(t) \mathcal{P}(t) \mathcal{P}^{-1}(t) = - \tilde{\mathcal{W}}(t) \tilde{\mathcal{W}}^T(t) \quad (3.101)$$

$$\Rightarrow \dot{\mathcal{P}}(t) = - \mathcal{P}(t) \tilde{\mathcal{W}}(t) \tilde{\mathcal{W}}^T(t) \mathcal{P}(t) \quad (3.102)$$

From (3.102), it is observed that  $\dot{\mathcal{P}}(t)$  is negative definite and the  $\mathcal{P}(t) \in \ell_\infty$  decreases over time. By performing time derivative of (3.99) and by adapting the algebraic manipulations, the consequent part parameters of the controller are obtained as

$$\begin{aligned} \dot{\mathcal{F}}(t) &= \dot{\mathcal{P}}(t) \mathcal{P}^{-1}(t) \mathcal{F}(t) - \mathcal{P}(t) \tilde{\mathcal{W}}(t) y(t) \\ &= - \mathcal{P}(t) \tilde{\mathcal{W}}(t) \tilde{\mathcal{W}}^T(t) \mathcal{F}(t) - \mathcal{P}(t) \tilde{\mathcal{W}}(t) y(t) \\ &= - \mathcal{P}(t) \tilde{\mathcal{W}}(t) y \left( \tilde{\mathcal{W}}^T(t) \mathcal{F}(t) + y(t) \right) \\ &= -K_D \mathcal{P}(t) \tilde{\mathcal{W}}(t) s_p(t) \end{aligned} \quad (3.103)$$

### 3.2.3 Stability analysis

The developed controller is adapted along conventional PD controller which is initially defined in (3.43). As the eT2QFNN is an approximator of general function, there exists  $\mathcal{F}^*$  such that:

$$y(t) = \mathcal{F}^{*T} \tilde{\mathcal{W}}(t) \quad (3.104)$$

Defining  $\tilde{\mathcal{F}}(t)$  as

$$\tilde{\mathcal{F}}(t) = \mathcal{F}(t) - \mathcal{F}^* \quad (3.105)$$

and

$$s_p(t) = \tilde{\mathcal{F}}^T(t) \tilde{\mathcal{W}}(t) \quad (3.106)$$

it is identified that, for the inputs to the fuzzy system  $e(t)$  and  $\dot{e}(t)$ , the approximation of general function is a second order one.

**Lemma:**

$$\frac{d \left( \mathcal{P}^{-1} \tilde{\mathcal{F}}(t) \right)}{dt} = \dot{\mathcal{P}}^{-1}(t) \tilde{\mathcal{F}}(t) + \mathcal{P}^{-1}(t) \dot{\tilde{\mathcal{F}}}(t)$$

$$\begin{aligned}
&= -\mathcal{P}^{-1}(t)\dot{\mathcal{P}}(t)\mathcal{P}^{-1}(t)\tilde{\mathcal{F}}(t) + \mathcal{P}^{-1}(t)\dot{\tilde{\mathcal{F}}}(t) \\
&= \tilde{\mathcal{W}}(t)\tilde{\mathcal{W}}^T(t)\tilde{\mathcal{F}}(t) - \tilde{\mathcal{W}}(t)s_p(t) \\
&= \tilde{\mathcal{W}}(t)s_p(t) - \tilde{\mathcal{W}}(t)s_p(t) \\
&= 0
\end{aligned} \tag{3.107}$$

This indicates  $\mathcal{P}^{-1}(t)\tilde{\mathcal{F}}(t)$  doesn't change over the time and hence  $\mathcal{P}^{-1}(t)\tilde{\mathcal{F}}(t) = \mathcal{P}^{-1}(0)\tilde{\mathcal{F}}(0), \forall t > 0$ .

$$\lim_{t \rightarrow \infty} \tilde{\mathcal{F}}(t) = \lim_{t \rightarrow \infty} \mathcal{P}(t)\mathcal{P}^{-1}(0)\tilde{\mathcal{F}}(0) \tag{3.108}$$

Since  $\mathcal{P}(t)$  is decreasing and  $\tilde{\mathcal{F}}(t) \in \ell_\infty, \mathcal{F}(t) \in \ell_\infty$ . Considering the following Lyapunov function:

$$V = \frac{1}{2}\tilde{\mathcal{F}}^T(t)\mathcal{P}^{-1}\tilde{\mathcal{F}}(t) \tag{3.109}$$

$$\begin{aligned}
V &= \tilde{\mathcal{F}}^T(t)\mathcal{P}^{-1}\dot{\tilde{\mathcal{F}}}(t) + \frac{1}{2}\tilde{\mathcal{F}}^T(t)\dot{\mathcal{P}}^{-1}\tilde{\mathcal{F}}(t) \\
&= \tilde{\mathcal{F}}^T(t)\tilde{\mathcal{W}}(t)s_p(t) - \frac{1}{2}\tilde{\mathcal{F}}^T(t)\mathcal{P}^{-1}(t)\mathcal{W}(t)\mathcal{W}^T(t)\tilde{\mathcal{F}}(t) \\
&= s_p^2(t) - \frac{1}{2}s_p^2(t) \\
&= \frac{1}{2}s_p^2(t)
\end{aligned} \tag{3.110}$$

To perform the stability analysis of the developed controller, some assumptions needs to be considered.

*Assumption 1.* The inputs of eT2FNN and their derivatives are bounded [341], [342].

$$|e(t)| \leq B_e, |\dot{e}(t)| \leq \dot{B}_e \quad \forall t \tag{3.111}$$

where  $B_e$  and  $\dot{B}_e$  are positive scalars.

This assumption with the adaptive law guarantees that the center of the quantum membership function  $c$ , standard deviation  $\sigma$  of the membership functions which is considered to be an interval, and the adaptive crisp values  $\Omega_{ij}$  are bounded as:

$$\begin{aligned}
\underline{B}_\sigma &\leq \|\underline{\sigma}_1\| \leq \bar{B}_\sigma, \quad \underline{B}_\sigma \leq \|\underline{\sigma}_2\| \leq \bar{B}_\sigma, \quad \|c_1\| \leq B_c, \quad \|c_2\| \leq B_c \\
\underline{B}_\sigma &\leq \|\bar{\sigma}_1\| \leq \bar{B}_\sigma, \quad \underline{B}_\sigma \leq \|\bar{\sigma}_2\| \leq \bar{B}_\sigma, \quad \|\bar{c}_1\| \leq B_c, \quad \|\bar{c}_2\| \leq B_c, \quad \|\bar{\Omega}_{ij}\| \leq B_\Omega
\end{aligned}$$

where  $\underline{B}_\sigma, \bar{B}_\sigma, B_c$ , and  $B_\Omega$  are positive scalars.

*Assumption 2.* The outputs of the eT2FNN controller are bounded [341], [342].

$$|y_{NN}(t)| \leq B_y, |\dot{y}_{NN}(t)| \leq \dot{B}_y, \forall t \tag{3.112}$$

$$|y_{PD}(t)| \leq B_y, |\dot{y}_{PD}(t)| \leq \dot{B}_y, \forall t \tag{3.113}$$

$$|y(t)| \leq B_y, |\dot{y}(t)| \leq \dot{B}_y, \forall t \tag{3.114}$$

where  $y$  and  $\dot{y}$  are known positive constants.

In assumptions 1 and 2, the inputs and outputs of the eT2QFNN are bounded as the learning rate considered in this research is adaptive and does not need to be known a priori. This is an obvious superiority of the current approach with respect to conventional type 2 fuzzy neural network and interval type 2 fuzzy logic control approaches in which the upper bounds of the states of the system should be known a priori in order to choose an appropriate value for the learning rate.

Further, it is well known that the sliding surface suffers from high-frequency oscillations in the control input, which are called chattering. The following are the two common methods used to eliminate chattering:

- 1) using a saturation function to replace the signum function;
- 2) inserting a boundary layer so that an equivalent control replaces the corrective one when the system is inside this layer.

*Assumption 3.* An inequality is considered for positive design parameters [343]. Generally, all the feasible design parameters may have equality as well as inequality constraints which can be referred as feasible region. As the problem under consideration in this research is trajectory tracking and position control, i.e., to move the system precisely by the eT2QFNN controller to perform the desired operation, it must be treated as an equality constraint. A feasible design must satisfy precisely all equality constraints. But, the experiments on control of two degree of freedom operation of the systems has shown that force feedback sometimes produces an undesirable chattering behaviour, where the plant repeatedly makes and breaks contact with the constraint surface. This behaviour is an example of a limit cycle, and is likely caused by the nonlinearity in the system dynamics introduced by the unilateral (i.e., inequality) constraint. Further, it is identified that the feasible region with respect to an inequality constraint is much larger than that with respect to the same constraint expressed as equality. Hence to scale these inequality constraints by a positive constant, the assumption 3 is made. This assumption shrinks the inequality of the feasible region, and the number of possible designs that can optimize the cost function is reduced improving the time of convergence for optimal solution, and achieving stability.

$$\Sigma > \dot{y} \tag{3.115}$$

Considered that the system inputs and the controller actions are bounded to positive constants, the Lyapunov function candidate in (3.110) is solved as

$$V = \frac{1}{2K_D^2} y_{PD}^2 \tag{3.116}$$

By applying differentiation, the (3.116) is formulated as:

$$\dot{V} = \frac{1}{K_D^2} y_{PD} \dot{y}_{PD} = \frac{1}{K_D^2} y_{PD} (\dot{y} + \dot{y}_{NN}) \quad (3.117)$$

This guarantees the conditions  $V > 0$  and  $\dot{V} < 0$ . For an arbitrary initial condition  $e(0) \neq 0$ , the output  $y_{PD}(t)$  and the error  $e(t)$  will approach zero within finite time  $t_f$ . Therefore, from Lyapunov stability theorem [344], the closed loop system achieves global stability, when the controller closely tracks the desired trajectory. This also achieves guaranteed position convergence and minimized tracking error.

### 3.2.4 Evolving Fuzzy Control for Helicopter system

In this section, the eT2QFNN method is evaluated for the 2DoF helicopter system with coupling among output channels. The trajectory of 2DoF helicopter has been defined by pitch angle and yaw angle movements. The main rotor lift is responsible for change in pitch and pulling forces of tail rotor are responsible for yaw angle movements. The objective is to decouple the two typical output channels such that the output responses follows a desired trajectory. From the dynamic model of the 2DoF helicopter system it has been identified that the pitch and yaw axes can be controlled by appropriately selecting the voltages of the pitch and yaw motors, respectively. Due to the linear dependency of each rule on the input variables, the developed controller is considered to be ideal for acting as an interpolating supervisor for the PD controller. From the block diagram in Figure 3.12, desired and measured pitch and yaw position caused to generate an error which is further applied to fuzzy input. Here, the pitch error  $e_p(t) = \alpha_d - \alpha$ , derivative of pitch error is  $\dot{e}_p(t)$ , yaw error is  $e_y(t) = \varphi_d - \varphi$ , and derivative of yaw error is  $\dot{e}_y(t)$ . Since the principle idea of developing the eT2QFNN is based on QMF with uncertain jump positions, the controller benefits from flexible rule formulation with nine initial rules. Further these initial membership functions are designed in a closed interval with universe of discourse lying between  $[-0.2, 0.2]$ . This range is further evolved during finding optimal position. Apart from the rules and universe of discourse, the vigilance parameter for performing rule growing mechanism is set at 0.65 for simplicity, and the FOU is created at 0.7. At the initial instant, two control rules are added at  $e_p(t)$  and  $e_y(t)$  corresponding to the criterion discussed in the algorithm. Further, the rule growing process inherently develops the rules based on the heuristic motion of the helicopter system.

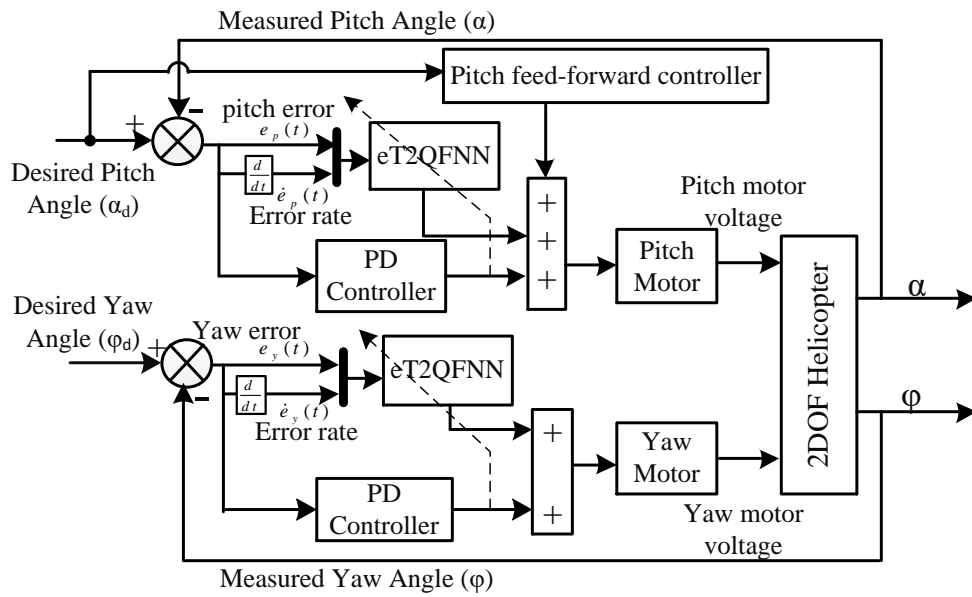
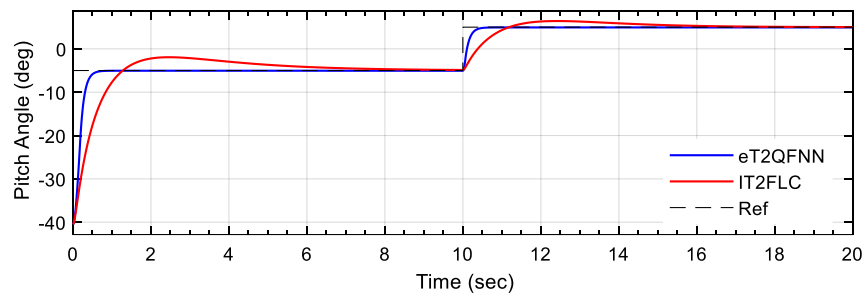


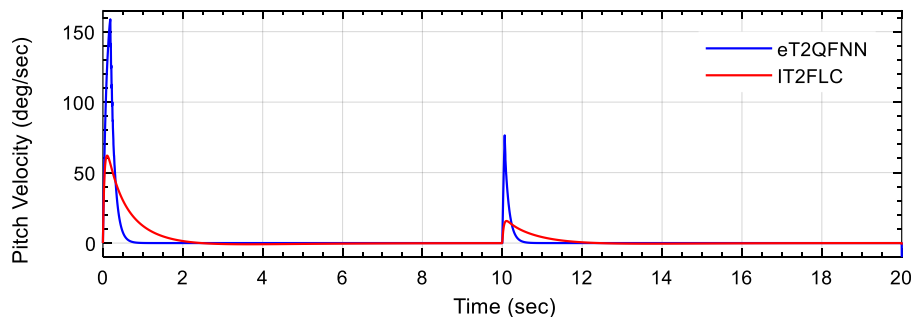
Figure 3.12. Evolving type 2 fuzzy neural network controller for helicopter control

### 3.2.4.1 Simulation Results

The response of pitch and yaw motion of the 2DoF helicopter with eT2QFNN controller is analyzed using Simulink/MATLAB software. The reference trajectory of amplitude 5 is considered for pitch and yaw path tracking assessment. To replicate the behavior of disturbance conditions during the simulation process, an automated white gaussian noise of 20db is considered along with the inputs driving the system. To assess the superiority of the developed controller, the results obtained are compared with conventional IT2FLC applied with the same setup. The corresponding results are shown in Figure 3.13.



(a) Pitch angle



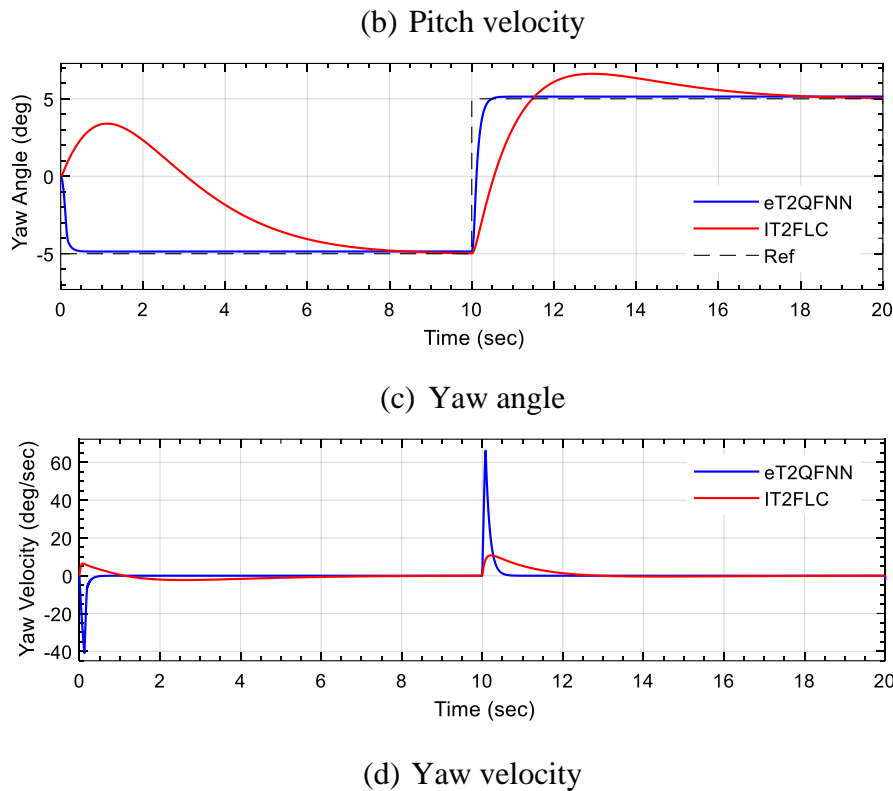


Figure 3.13. Pitch and yaw output with evolving type 2 quantum fuzzy neural network control of helicopter in Simulink

From Figure 3.13, it is observed that, the pitch angle for eT2QFNN in Figure 3.13(a) reach an angle of  $-5.1^{\circ}$  at 0.91 sec for the first negative step. The control action is so strong that it can uplift the body within 0.4 sec and able to hover very fast. The helicopter body is at rest position and tilted at  $-40$  degree and so pitch motor required strong thrust at the start to uplift the body. Due to strong control action at 0.4 sec the pitch velocity in Figure 3.13(b) reaches to its highest value of 159 degree/sec and will be able to generate enough force as per requirement. The pitch voltage also stops varying after reaching its stable state and gives no oscillations. The yaw angle in Figure 3.13(c) for eT2QFNN controller reaches to its trajectory of angle  $-4.8^{\circ}$  at 0.6sec. To maintain the coupling between pitch and yaw rotor, the pitch velocity is comparatively less, and the yaw propeller are rotating with yaw velocity as shown in Figure 3.13(d). The variation is observed to be around  $-41$ degree/sec at 0.2 sec. When compared to the conventional IT2FLC, the proposed controller depicted fast response to variation in trajectory with less settling time.

Further, the root mean square error (RMSE) has been calculated for the measured pitch angle with the reference sine wave trajectory as shown in Table 3.10 using Simulink. Besides, Table 3.11 points out a comparative assessment of controller's performance in terms of time response characteristics for pitch and yaw angle of helicopter system.



Table 3.10. Root mean square error for evolving type 2 quantum fuzzy neural network control of helicopter in Simulink

Controller	Root mean square error	
	Pitch angle (deg)	yaw angle (deg)
IT2FLC	2.8167 deg	4.9176 deg
eT2QFNN	2.1987 deg	0.1976 deg

Table 3.11. Time response analysis for evolving type 2 quantum fuzzy neural network control of helicopter in Simulink

Controller	Pitch Response		Yaw Response	
	Settling time ( $t_s$ ) (sec)	Steady-state error ( $e_{ss}$ ) (cm)	Settling time ( $t_s$ ) (sec)	Steady-state error ( $e_{ss}$ ) (cm)
IT2FLC	6.15 sec	0.81 cm	7.92 sec	6.93 cm
eT2QFNN	0.56 sec	0.13 cm	0.61 sec	0.27 cm

From the observation of the time response analysis, settling time  $t_s$  and steady-state error  $e_{ss}$  are minimum in case of wavelet fuzzy which are 0.56 sec and 0.13 cm for pitch control & 0.61 sec and 0.27 cm for yaw control. Further, the action of the developed controller is tested in laboratory on a 2DoF helicopter test rig.

#### 3.2.4.2 Real-time Results

To analyse the action of the controller on the real-time operation of the helicopter system, the hardware in loop-application programming interface (HIL-API) associated with the Quanser helicopter setup is used. Initially, the simulated models in section 3.2.4.1 are calibrated for code generation with MATLAB/Simulink and Quanser software. Further, a C code generated in using the MEX function and interpreted to the hardware using the RS232 interface with the DAQ board. To achieve the efficient operation of the developed controller and simultaneous action of the helicopter system for every rule fired by the eT2QFNN for PD controller, Immediate I/O is adapted. This performs the hardware-in-the-loop-application programming interface (HIL-API) operation of the developed controller with the real-time setup. The corresponding pitch and yaw characteristics for the real-time operation of the helicopter are shown in Figure 3.14. Similar to the simulation process, the performance of the developed approach is assessed and the results are compared with the action of conventional interval type 2 fuzzy logic controller developed for the same helicopter setup.

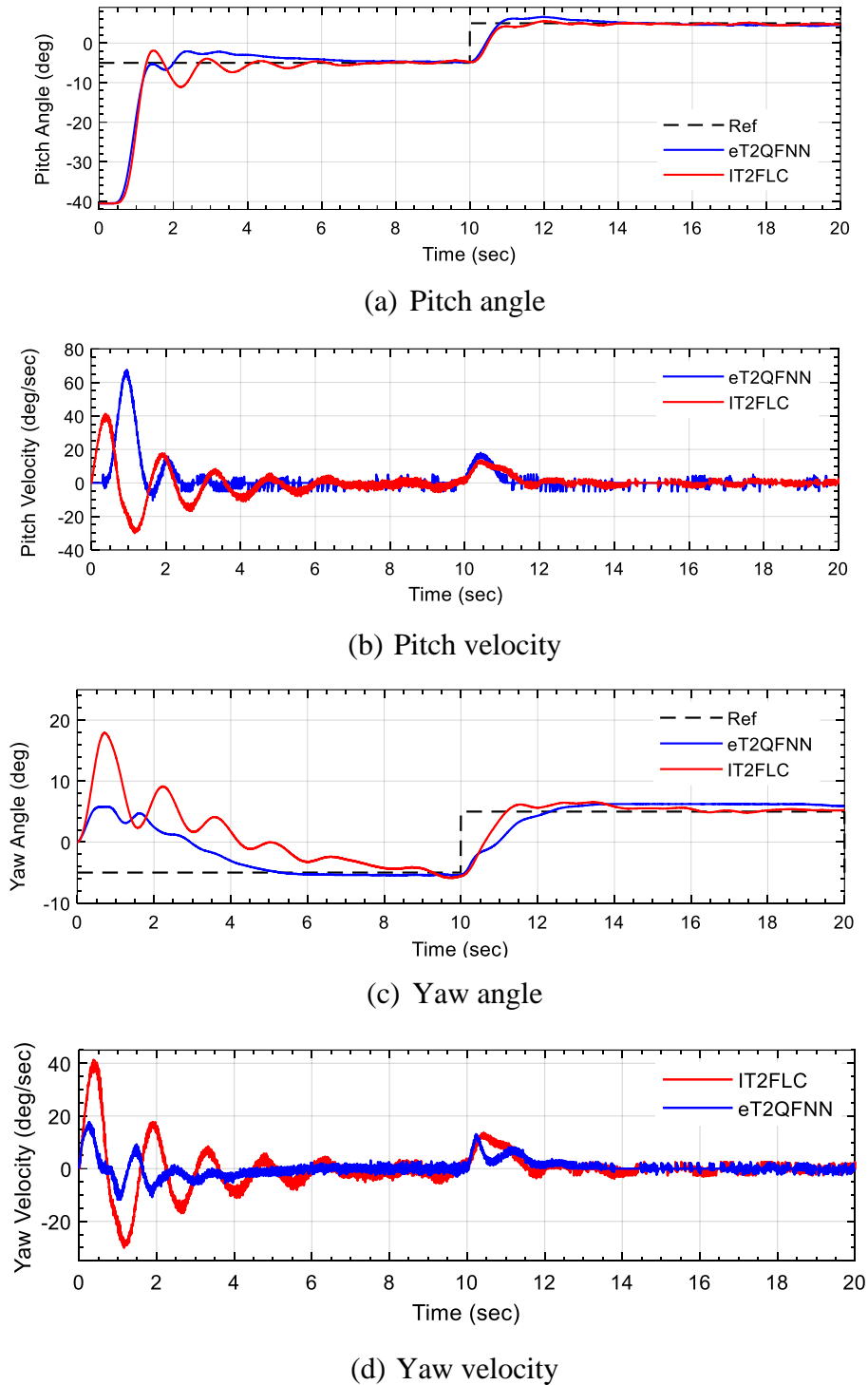


Figure 3.14. Pitch and yaw output with evolving type 2 quantum fuzzy neural network control of helicopter in real-time

Figure 3.14 shows output response of the pitch angle movement, yaw angle movement, pitch velocity, and yaw velocity for developed controller on square trajectory. To assess the superiority of the developed controller, the action of IT2FLC on the same setup is carried out and the results obtained are compared. The pitch angle due to action of eT2QFNN in Figure 3.14(a) reaches an angle of  $-5.1^{\circ}$  at  $t = 1.5 \text{ sec}$  for the first time and touches the reference

trajectory. Further it settles to the angle of  $-4.9^{\circ}$  at  $t = 5.41 \text{ sec}$  and stabilizes the system. The pitch velocity due to action of eT2QFNN in Figure 3.14(b) reaches to its highest value of 69 degree/sec at  $t = 0.8 \text{ sec}$ . The velocity reduces to 69 degree/sec in real-time as no disturbances were considered in simulation. Similarly, the yaw angle variation due to action of eT2QFNN in Figure 3.14(c) settles at trajectory of angle  $-5.8^{\circ}$  at  $t = 5.6 \text{ sec}$  and maintains the constant trajectory. As mentioned earlier, to keep the proper coupling between pitch and yaw rotor, the pitch velocity is comparatively less than the yaw velocity depicted in Figure 3.14(d). The yaw velocity goes up to 18.6 degree/sec at  $t=0.6 \text{ sec}$ . In a similar observation, the IT2FLC underperformed while controlling both the pitch and yaw motors.

Further, the RMSE for both the controllers is provided in Table 3.12. Besides, the settling time ( $t_s$ ) and steady-state error ( $e_{ss}$ ) are minimum in case of eT2QFNN controller which is 6.75 sec, 5.97 cm for pitch control & 5.1 sec, 3.17 cm respectively for yaw control as shown in Table 3.13.

Table 3.12. Root mean square error for evolving type 2 quantum fuzzy neural network control of helicopter in real-time

Controller	Root mean square error	
	Pitch angle (deg)	Yaw angle (deg)
IT2FLC	7.6516 deg	13.2168 deg
eT2QFNN	3.1698 deg	4.1689 deg

Table 3.13. Time response analysis for evolving type 2 quantum fuzzy neural network control of helicopter in real-time

Controller	Pitch Response		Yaw Response	
	Settling time ( $t_s$ ) (sec)	Steady-state error ( $e_{ss}$ ) (cm)	Settling time ( $t_s$ ) (sec)	Steady-state error ( $e_{ss}$ ) (cm)
IT2FLC	7.23 sec	8.73 cm	9.72 sec	13.35 cm
eT2QFNN	6.75 sec	5.97 cm	5.1 sec	3.17 cm

The comparison between action of both the controllers on the test rig depicted that IT2FLC has large settling time, and high oscillations, whereas the proposed controller depicted fast response to variation in trajectory with less settling time.

### 3.2.5 Evolving Fuzzy Control for Ball Balancer System

The numerical simulation of the 2DoF ball balancer model is developed using MATLAB/Simulink software. Since the arrangement of the plate on the two servo units is symmetrical, the action of the controller on one servo unit has an impact on the action of the controller on the other servo unit. This operates them in a coupled environment irrespective of the aspect that both the controllers are designed in a decoupled environment.

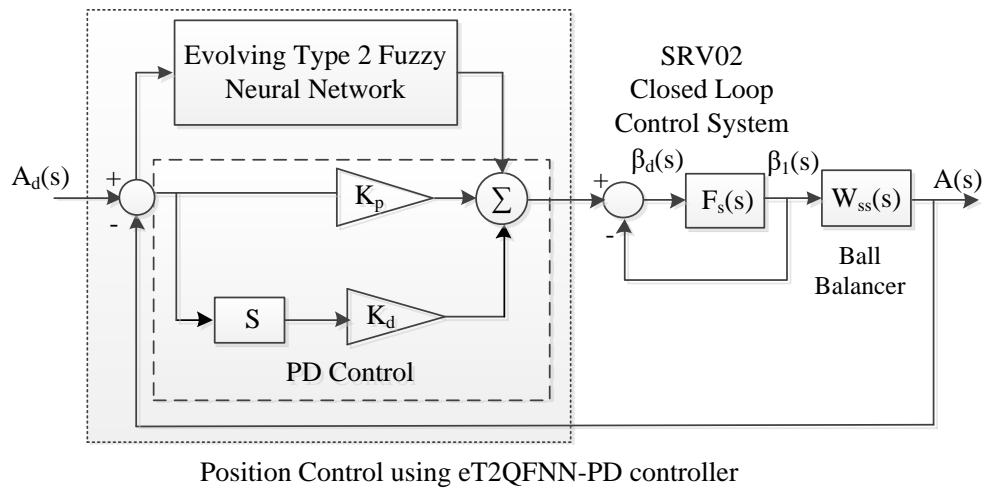
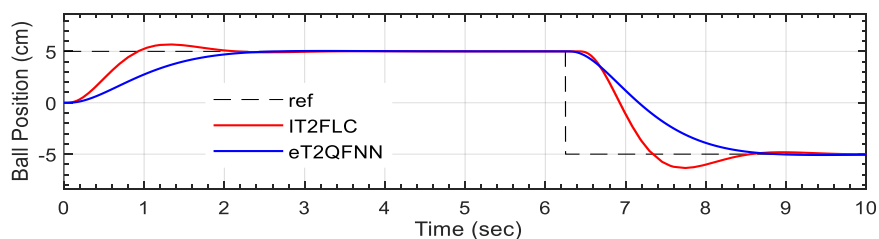


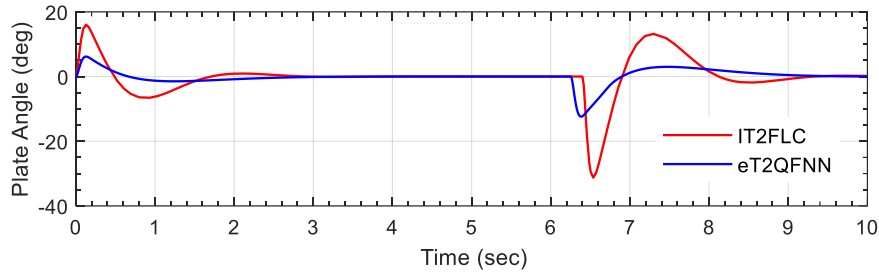
Figure 3.15. Evolving type 2 fuzzy neural network control for ball balancer system

#### 3.2.5.1 Simulation Analysis

Initially, the system is operated to control the position of the ball on a plate in a square trajectory. This is achieved by controlling the ball balancer with eT2QFNN by providing a square input signal with a frequency of 0.08Hz and amplitude of 3 as the reference trajectory. The eT2QFNN operates by measuring the error between the desired and measured ball position to optimize the values of the PD controller. The results obtained are then compared with the action of the classical IT2FLC control on the same simulation, operating for the same trajectory to assess the performance of the eT2QFNN controller. The results of the ball position, servo angle, and voltage of the ball balancer system for the action of eT2QFNN controller and the IT2FLC controller are shown in Figure 3.16.



(a) Ball position



(b) Plate angle

Figure 3.16. Ball balancer output with evolving type 2 quantum fuzzy neural network control in Simulink

Figure 3.16 (a) shows the results of the position of the ball on the x-axis for both IT2FLC and the eT2QFNN control. The results identify that the ball position settles around 2.1 sec at 5 cm with the action of eT2QFNN controller. In this case, the eT2QFNN has a minimum final position and reaches the desired value in very less time. Further, in Figure 3.16(b), the servo angle response of the ball on the x-axis shows that variation in the control angle is provided by the servo motor. The minimum control angle defines the accuracy of a controller in achieving balancing control for the ball balancer system. In this case, the eT2QFNN has less control angle when compared with the IT2FLC. This will move the plate very slowly while balancing the ball and helps in achieving a stable response for the system. Hence, the eT2QFNN has better performance when compared with the IT2FLC control. Further, to assess the performance of the eT2QFNN controller, the time domain specifications are calculated and the results are shown in Table 3.14. From the result, it is observed that the peak overshoot of IT2FC is 13.5%, which causes huge oscillations and made the ball difficult to balance on the plate. On the other side as per the results observed in the graph, it's observed that eT2QFNN has a fine response to peak overshoot 0.827% and shows the excellent balance of ball on the plate without any oscillation.

Table 3.14. Time response analysis for evolving type 2 quantum fuzzy neural network control of ball balancer in Simulink

<b>Controllers</b>	<b>Peak time</b> ( $t_p$ ) (sec)	<b>Settling time</b> ( $t_s$ ) (sec)	<b>Peak overshoot</b> ( $M_p$ ) (%)	<b>Steady-state error</b> ( $e_{ss}$ ) (cm)
IT2FLC	0.19sec	2.8 sec	13.5%	0.00472 cm
eT2QFNN	0.14 sec	2.2 sec	0.827%	0.000648 cm

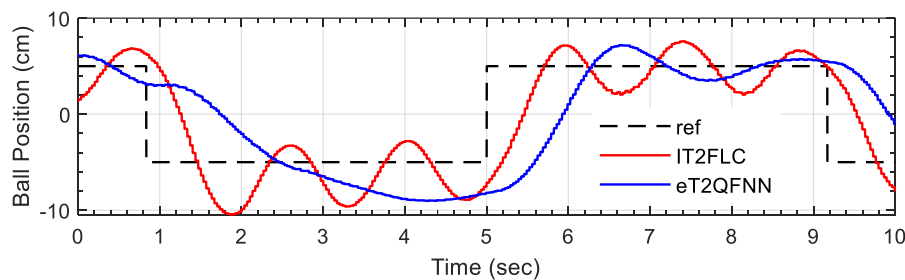
Further, the RMSE value of IT2FLC and eT2QFNN control actions on the operation of ball balancer for square trajectory are calculated in terms of position and plate angle parameter. The results are tabulated as shown in Table 3.15.

Table 3.15. Root mean square error for evolving type 2 quantum fuzzy neural network control of ball balancer in Simulink

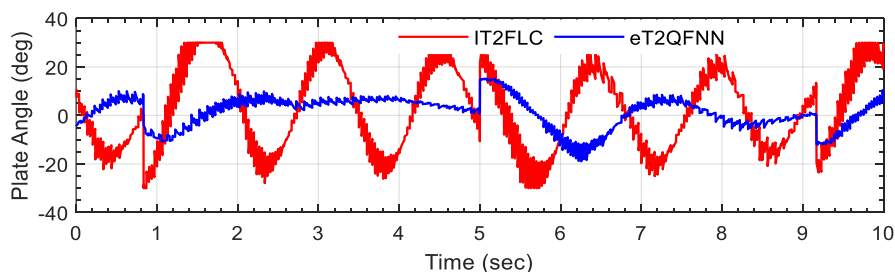
Controllers	Root mean square error	
	Position (cm)	Angle (deg)
IT2FLC	2.1455 cm	3.3255 deg
eT2QFNN	1.7586 cm	0.5931 deg

### 3.2.5.2 Experimental Analysis

The hardware study is initiated with the same parameters as in the simulation analysis. Further, the developed eT2QFNN controller and the IT2FLC are implemented on the same setup, and the output responses of each controller are identified. The ball position, plate angle, and operating voltage of the servo mechanism are observed as shown in Figure 3.17.



(a) Ball position



(b) Plate angle

Figure 3.17. Ball balancer output with evolving type 2 quantum fuzzy neural network control in real-time

From the results in Figure 3.17 (a) it is observed that the eT2QFNN provides stabilized operation for the ball balancer system by balancing the ball on the plate with almost zero oscillation. The results in Figure 3.17 (b) show that the plate stabilization angle of eT2QFNN is maintained between 15 to 20 degrees, whereas the IT2FLC has a stabilization angle varying between  $-30$  to  $35$  degree. This identifies that the eT2QFNN has smooth operation even in the presence of external disturbances that affect the movement of the ball during real-time experiments. Besides, the time-domain characteristics of the eT2QFNN and the IT2FLC are

calculated to assess their operation in the real-time experiment. As of Table 3.16, the response of eT2QFNN is better than the IT2FLC for peak time, settling time, and peak overshoot.

Table 3.16. Time response analysis for evolving type 2 quantum fuzzy neural network control of ball balancer in real-time

<b>Controllers</b>	<b>Peak time</b> ( $t_p$ ) (sec)	<b>Peak overshoot</b> ( $M_p$ ) (%)	<b>Steady-state error</b> ( $e_{ss}$ ) (cm)
IT2FLC	1.37 sec	22.9%	1.151 cm
eT2QFNN	0.89 sec	12.4%	0.487 cm

Similarly, the RMSE value for the action of IT2FLC and eT2QFNN controller on the operation of real-time ball balancer setup for square trajectory are calculated in terms of position and plate angle parameter. The results are tabulated as shown in Table 3.17.

Table 3.17. Root mean square error for evolving type 2 quantum fuzzy neural network control of ball balancer in real-time

<b>Controllers</b>	<b>Root mean square error</b>	
	Position (cm)	Angle (deg)
IT2FLC	5.1894 cm	3.4049 deg
eT2QFNN	3.5680 cm	2.7517 deg

### 3.3 CONCLUSION

In this chapter, the control of benchmark systems is achieved through hybrid and evolving intelligent control approaches. Initially, the wavelet fuzzy control is developed for helicopter and ball balancer systems. The advantage of wavelet transforms while analysing the non-stationary signals provides an upper hand while developing the controller. Besides, the wavelet transforms investigate the input signals which extricates and distinguish the segments of the frequency signal in a different period and represent them to another frame. Moreover, the use of discrete wavelet transforms denoises the signal by decomposing and reconstructing the signal using dilated and shifted forms. This helps in tuning the weights of fuzzy logic controller rules after signal reconstruction to achieve required control with the robotic system. This overcomes the drawbacks of transparent interpretation of choosing fuzzy rules, achieved position, balancing control, and improved transient response compared to conventional techniques.

Similarly, the evolving type2 quantum fuzzy neural network control scheme is developed for achieving balancing and attitude tracking control in ball balancer and helicopter systems respectively. The controller generated the required rules through rule growing mechanism and

parameter adjustment learning scenario. Further, a sliding surface based adaptive law is equipped to compensate the nonlinearity of the systems operating with eT2QFNN and PD controller. Simulation and experimental analysis are conducted with the developed approaches and modelled plants for offline path tracking for a predefined trajectory with the systems. The corresponding results are compared with the IT2FLC technique which depicted the robustness of the proposed approach.



## Chapter 4. PROBABLISTIC ALGORITHMS FOR CONTROL OF 2DOF SYSTEMS

The effect of unknown disturbances during the control of non-holonomic systems have been the subject of research due to their usefulness in various applications. As a result, the modeling and control of these systems is dealt in a theoretical way by achieving steady state operation through a feedback control loop. From the literature it is observed that the conventional process of controlling 2DoF systems while considering the parametric uncertainties lacked in the areas of selecting the bias and balancing the prognostic variables. The discussions depict that the uncertainty in the system played a big role, and it is necessary to bring a solution to this problem. For that matter, approximation and relaxation methods have been proposed considering their conservative nature. Besides, to overcome the above-mentioned drawbacks, and achieve the stability and performance requirements, advanced control techniques which depend upon probabilistic control and randomization are being implemented in this area of research [153], [345], [346]. Initially, these methods have assumed that the parametric uncertainty affects the system in a probabilistic pattern and then the performance level has been provided to check its ability. Further, the mathematical model required to design these algorithm-based controls is obtained by numerical linearization of the full order nonlinear system [157]. Since, the explicit relationship between the state- space matrices and the uncertain parameters should be available all the time, linearization process subjected to repetition at the time of variation in the uncertain parameters is adapted. Considering these aspects, this chapter proposes an optimized probabilistic control approach to design a control algorithm with reference to randomization for gain matrix calculation and for closed-loop system analysis as well [347]. These probabilistic control algorithms take less computational time and easy to implement [153] and at the same time have robust boundary conditions and are less conservative at the cost of probability risk failure. This approach is not only limited to control engineering but also explores the general engineering design, robust optimization where the environment is directly affected by the uncertainty. Other than analysis of system, the probabilistic methodology reveals its maximum capacity with respects to control frameworks.

## 4.1 RANDOMIZED PROBABILISTIC ALGORITHM BASED CONTROL DEVELOPMENT FOR 2DOF HELICOPTER SYSTEM

### 4.1.1 Randomized Algorithms

The randomized algorithms are defined as an optimization technique where the actual system uncertainties initiate the randomization process while the deterministic decision parameters remain unconsidered. Usually, this randomness is generated in a system due to the involvement of stochastic uncertainties which produce various results for same input at different runs. This indicates a probabilistic property which defines the system as probably approximately correct (PAC). The generalized model representing different uncertainty sources that may affect the operation of a dynamic system is shown in Figure 4.1. This standard approach is frequently used in modern control theory and is termed as the  $M - \Delta$  model.

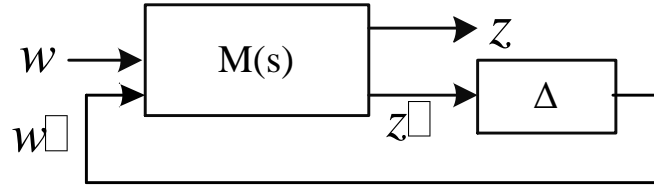


Figure 4.1. Generalized  $M - \Delta$  model for uncertainty representation

Where,  $M \in \mathcal{RH}_{\infty}^{c,r}$  space with real coefficients  $c, r$  and corresponds to the transfer matrix of the system. This includes the extended part of the system which consist of reference signals, disturbances, and noise ( $w$ ), along with a controller which represents tracking errors, and controlled signals ( $z$ ). Further,  $\Delta \in \mathcal{RH}_{\infty}^{r\Delta, c\Delta}$  corresponds to all the time invariant uncertainties affecting the system also known as a random matrix supported by a structured set  $\beta_{\mathbb{D}}$ . Considering the  $M - \Delta$  configuration, the measurable performance function analysis is defined by

$$J(\Delta): \mathbb{D} \rightarrow \mathbb{R} \tag{4.1}$$

The above function considers various performance requirements at an associated level  $\gamma$  with a structured uncertainty set  $\mathbb{D}$ . Further, this configuration along with random uncertainty can be used to verify the probabilistic performance of the system in both the normal operation and worst-case operation. These scenarios are defined as follows:

*Probabilistic performance verification:* Consider a random matrix  $\Delta$  with density  $f_{\Delta}(\Delta)$  supported by  $\beta_{\mathbb{D}}$  for verification of probabilistic performance under normal operation using randomized algorithms. The assigned probability levels are given by  $\epsilon \in (0, 1), \delta \in (0, 1)$ . For

any given performance function  $J(\Delta): \mathbb{D} \rightarrow \mathbb{R}$  at an associated level  $\gamma$ , the randomized algorithm provides an estimate  $\hat{p}_N(\gamma)$  of probability performance with least probability of  $1 - \delta$  that is within  $\epsilon$  from  $p(\gamma)$ . This is known as the probability of violation.

$$p(\gamma) = P_R\{J(\Delta) \leq \gamma\} \quad (4.2)$$

Further, the estimate should be constructed considering a finite number  $N$  of  $\Delta \in \beta_{\mathbb{D}}$  random samples.

*Probabilistic worst-case performance:* Similar to the probabilistic performance verification, the probabilistic worst-case performance also follows the same conditions. Further, for any given performance function  $J(\Delta): \mathbb{D} \rightarrow \mathbb{R}$  at an associated level  $\gamma$ , the randomized algorithm provides a performance level  $\hat{\gamma}_N \leq \sup_{\Delta \in \beta_{\mathbb{D}}} J(\Delta)$  with least probability of  $1 - \delta$  such that

$$P_R\{J(\Delta) \leq \hat{\gamma}_N\} \geq 1 - \epsilon \quad (4.3)$$

Further, the performance level  $\hat{\gamma}_N$  should be constructed considering a finite number  $N$  of  $\Delta \in \beta_{\mathbb{D}}$  random samples.

#### 4.1.2 Randomized Algorithm for Probabilistic Controller Synthesis

For a fixed plant  $G(s)$ , the controller  $K(s, \theta)$  is designed to synthesize the interconnection as shown in Figure 4.2. Here, the system stability and performance are achieved in the presence of uncertainty, and  $\theta \in \mathbb{R}^{n_\theta}$  corresponds to the design parameter vector. The terms  $u$  and  $y$  correspond to the inputs and outputs of the plant respectively.

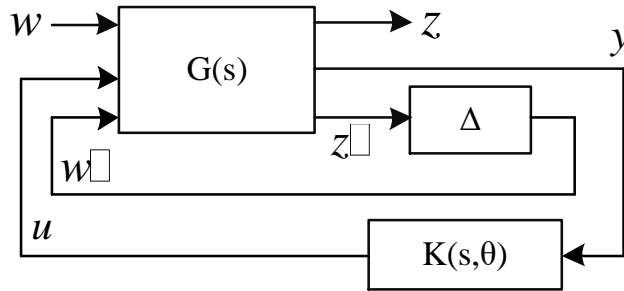


Figure 4.2. Controller design for a fixed plant

Generally, the randomized algorithms are based on interplay of deterministic optimization and random sampling in design parameter and uncertainty space respectively. Considering the performance function in (4.1), the constraints of design and performance of uncertain systems can be rewritten in the form of inequality as

$$J(\Delta, \theta) \leq \gamma \quad (4.4)$$

where the performance function for design  $J(\Delta, \theta): \mathbb{D} \times \mathbb{R}^{n\theta} \rightarrow \mathbb{R}$  is scalar valued. Therefore, for a fixed controller parameter  $\theta$ , the condition in (4.1) is considered as a special case of (4.4). Hence the probability of violation and reliability of the design can be defined as follows:

*Probability of violation and reliability:* Consider the accuracy  $\epsilon \in (0,1)$  and performance level  $\gamma > 0$ . Given  $\theta \in \mathbb{R}^{n\theta}$ , the probability of violation for the design is given by

$$V(\theta) \doteq P_R\{J(\Delta, \theta) > \gamma\} \quad (4.5)$$

*Probability of violation:* From the above conditions, the reliability of design  $\theta$  is

$$R(\theta) \doteq 1 - V(\theta) \quad (4.6)$$

For  $\theta$  to be probabilistic and reliable,  $V(\theta) \leq \epsilon$ , or equivalent to  $R(\theta) \geq 1 - \epsilon$  for a given accuracy  $\epsilon \in (0,1)$ .

#### 4.1.2.1 Managing Uncertainty

A transfer function, linear matrix inequalities (LMI) or polynomial is referred as uncertain object if it reliant on complex and structured set of uncertain parameters. The uncertain parameters are considered as random variables for analysis of uncertain system through probabilistic methods. The uncertain transfer function of the system is based on its complexed uncertain parameters with the given variant distribution of randomness. The uncertainties involved in the system is defined separately with its structure. Once the helicopter system is linearized, and the uncertainty has been added to the model through its various parameters, the randomized algorithm is used to generate a large number of random samples in order to analyse and synthesize the system process. The uncertainty presented in the system has been handled in two ways. Firstly, a series of uncertainties is generated by a randomized algorithm. This uncertainty series is blend of vector, scalar, and matrices with predefined distributions in MATLAB. In the next approach, every uncertain parameter is defined in m-file as per system design parameters while adapting its composition rules. This m-file tracks a basic structure of user function. After describing the user function file, the necessary randomized algorithm comes into the action calling as per argument defined in the user function file. This method is quite easy to implement and provides fast speed on execution and usability.

#### 4.1.2.2 System design reliant on a randomized algorithm

The purpose of the probabilistic method is not only examining, create and performance verification of the system but also handle the synthesis and design of a robust controller for the system. With this advancement, the randomized algorithm creates a wide application in the

field of control as well. A randomized algorithm is an example of this approach where designing the controller has been done based on random sampling in the presence of uncertainty, and deterministic convex optimization.

Consider a design vector variable  $\theta \in \Theta \subseteq \mathbb{R}^{n_\theta}$ , and  $\Delta \in \mathbb{D}$  for the uncertain parameter of a system. The performance function for the fixed parameter, performance objective with inequality specification for desired performance level have already been explained in detail in the above section and, further will be used for optimal probabilistic feasible control design. Now for simplification all, design and performance constraints are expressed in terms of design inequality,  $f(\Delta, \theta) \leq 0$  where scalar valued function for design parameter is denoted by  $f(\Delta, \theta): \mathbb{D} \times \Theta \rightarrow \mathbb{R}$ . For determination of  $\theta$ , in a manner with  $J(\Delta, \theta) \leq \gamma$ , the function will be

$$f(\Delta, \theta) = J(\Delta, \theta) - \gamma \quad (4.7)$$

A system is a robust probabilistic design only if the design vector  $\theta$  follows the inequality  $f(\Delta, \theta) \leq 0$  for every  $\Delta$  randomness. The probability level is defined as  $p^* \in (0,1)$ . The minimal probability followed by a randomized algorithm is  $1-\delta$  and design parameter  $\theta_{pr} \in \Theta$  in the following way:

$$Prob\{\Delta \in \mathbb{D}: f(\Delta, \theta_{pr}) < 0\} \geq p^* \quad (4.8)$$

where  $\theta_{pr}$  corresponds to the design parameter reliant on a fixed number  $N$  for random samples of  $\Delta$ .

#### 4.1.2.3 Design of sequential randomized algorithm

Given the probability level  $p^* \in (0,1)$ , the minimal probability  $(1 - \delta)$  followed by the design parameter  $\theta_{pr} \in \Theta$  and the randomized algorithm is given by

$$Prob\{\Delta \in \mathbb{D}: f(\Delta, \theta_{pr}) < 0\} \geq p^* \quad (4.9)$$

and derived using Algorithm 4.1 as follows:

---

<b>Algorithm 4.1:</b> Robust probabilistic design with randomised algorithm [151]	
<i>Step 1:</i>	Initialize the algorithm by setting $k = 0$ and considering that $\theta_0 \in \Theta$ .
<i>Step 2:</i>	Calculate the sample size function $N(k)$ , such that
	$N(k) = \left\lceil \frac{\log_{10} \frac{\pi^2 (k+1)^2}{6\delta}}{\log_{10} \frac{1}{p^*}} \right\rceil \quad (4.10)$
<i>Step 3:</i>	Assess the probabilistic vision,
	<b>for</b> $i = 0$ to $N(k)$
	Generate a random vector sample $\Delta^{(i)}$ with uniform distribution
	<b>if</b>
	$f(\Delta^{(i)}, \theta_k) > 0$ , set $\Delta_k = \Delta^{(i)}$ and go to next step
	<b>else if</b>

$f(\Delta^{(i)}, \theta_k) < 0$ , return to initialization.

**end.**

*Step 4:* Construct a new candidate solution by updating  $\theta_{k+1}$

*Step 5:* Set  $k = k + 1$  and continue from Step 2.

There are two possible outcomes of this algorithm which completely dependent on probabilistic vision. If probabilistic vision is returning to its true value, then  $\theta_k$  is able to pass feasibility test on  $N(k)$  trails. Here  $\theta_k$  is defined as a feasible solution of the probabilistic approach. For sample size function  $N(k)$  shown in equation (4.10) with probability greater than  $1 - \delta$ , the final solution will be  $\theta_{pr}$  as per equation (4.9). Further, the ‘For’ loop breaks at  $i < N(k)$  in the probabilistic oracle only if oracle fails to return at its final solution along with phenomena violation  $\Delta_k: f(\theta_k, \Delta_k) > 0$ . Then the parameter solution  $\theta_k$  such that

$$\theta_{k+1} = alg(\theta_k, \Delta_k) \quad (4.11)$$

where  $alg$  corresponds either to the gradient expressed as ellipsoid or to a cutting plane probabilistic algorithm. For the implementation of the algorithm the function gradient ( $f$ ) with respect to  $\theta$  for fixed uncertainty. For further consideration, robust feasibility elaborated by LMI in the following manner:

Estimate  $\theta$  in a way that  $F^{(j)}(\Delta, \theta) \leq 0$  with  $j = 1, \dots, m$ , and  $\forall \Delta \in \mathbb{D}$ , where

$$F^{(j)}(\Delta, \theta) = F_0^{(j)}(\Delta) + \sum_{i=1}^n \theta_i F_i^{(j)}(\Delta), \quad (4.12)$$

and  $F_i^{(j)}(\Delta), i = 0, \dots, n$  are symmetric in nature with order of  $m \times m$  real matrices. These matrices are dependent on uncertainty vector  $\Delta \in \mathbb{D}$  in a nonlinear way. Further, its simplified as

$$f(\Delta, \theta) = \max_j \lambda_{max}(F^{(j)}(\Delta, \theta)) \quad (4.13)$$

Here, the randomized algorithm is collaborated with the LMI because of its easy handling of sub gradient of the function as shown in equation (4.13), and computation becomes simple.

This can be depicted by sub gradient of  $f(\Delta, \theta) = \lambda_{max}(F(\Delta, \theta))$  at  $\theta = \theta_k$  such that

$$g_{\Delta} \theta_k = [\xi_{max}^T F_1(\Delta) \xi_{max} \dots \xi_{max}^T F_n(\Delta) \xi_{max}]^T. \quad (4.14)$$

where  $\xi_{max}$  denotes the eigen vector, and  $F(\Delta, \theta_k)$  corresponds to the largest eigen value. This approach eliminates the manual calculations of the gradient.

#### 4.1.3 Reachability Analysis

Consider a state space representation  $A$  characterized for a continuous dynamical system by the means of a differential equation  $\dot{a} = f(a, u)$ . For any given set  $A_0 \subset A$ ,  $u$  corresponds to a set of admissible input signals, and all the uncertainties and potential disturbances between

$a_0 \in A_0$  may affect the system operation. Hence, to estimate the change in system performance along with the action of the controller, it is necessary to compute the effects of these uncertainties and disturbances. Further, these implications can be used to identify the effectiveness of system operation for every acceptable disturbance. Here, the accurate operation indicates that, either the terrible subsets of  $A$  have been avoided, or the complex chronological aspects have been possibly increased. Moreover, this helps in identifying the robustness of the system against multiple disturbances especially for the conditions where the parameter estimates are unknown. Traditionally, this process of analysing performance and robustness of the system were adapted with discrete-continuous time systems and were later involved with systems adhering to state variables. Further, it is identified that, to observe the accurate operation of the system under uncertainties and disturbances, every individual model adapts one input parameter along with an initial condition. Ideally, this process is repeated for all the values in the uncertainty set and for all the possible disturbances experienced by the system during its operation. This indicates the less feasibility of the approach for application with systems involving large uncertainty data sets or disturbances. In light of the above disadvantages, new methods with simplified procedure and which are capable of handling large disturbances needs to be adapted.

Further, it is identified that the reachability analysis [348], [349] is capable of handling uncertainties and large disturbances in simulations using the breadth first approach. This approach simulates the conventional process of identifying the robustness of the system against multiple disturbances by considering a continuous loop of the sample set (elements of uncertainty set or large disturbances) [350]. This eliminates the waiting time for starting a new test and improves the confidence in the estimations provided by the approach. Besides, the set-based approach also includes the information corresponding to the transient behaviour system apart from the stationary behaviour information in the conventional approaches.

In this research, the reachability analysis is carried out with the help of ellipsoidal method [351] on two different operating scenarios of the unmanned helicopter. These scenarios deal with the open loop and closed loop operation of the unmanned helicopter with uncertainties and disturbances using the developed controller. Further, to implement the reachability analysis with the ellipsoidal method, the ellipsoidal calculus is calculated for both the scenarios. The adapted ellipsoidal method for reachability analysis has advantages with, linear time space, and the quadratic state space. Moreover, this method represents the linear and nonlinear systems as reach sets based on the internal and external ellipsoids respectively. This estimates the open

loop and closed loop reach sets for verifying the performance, robustness, and safety of the system.

#### 4.1.4 Numerical Simulation

For the analysis of probabilistic method based on a randomized algorithm, the two degree of freedom (DoF) helicopter has been considered in the experiment. This helicopter system is not fixed but the parameters like a moment of inertia of pitch and yaw, the mass of the system have been considered as uncertain parameters. Further, the system is driven by a white gaussian noise along with the disturbances, and error generated with response to the reference signals. Here, the sequential randomized algorithm makes available a controller that stabilizes the uncertain system with prespecified probability. From a randomized algorithm with a probabilistic approach, a gain matrix is generated which represents a unique state-feedback controller. The block diagram of the probabilistic controller developed for the control of the helicopter system is shown in Figure 4.3. The feedforward control is integrated with a randomized algorithm based probabilistic controller (RABPC) for providing necessary gain to the pitch.

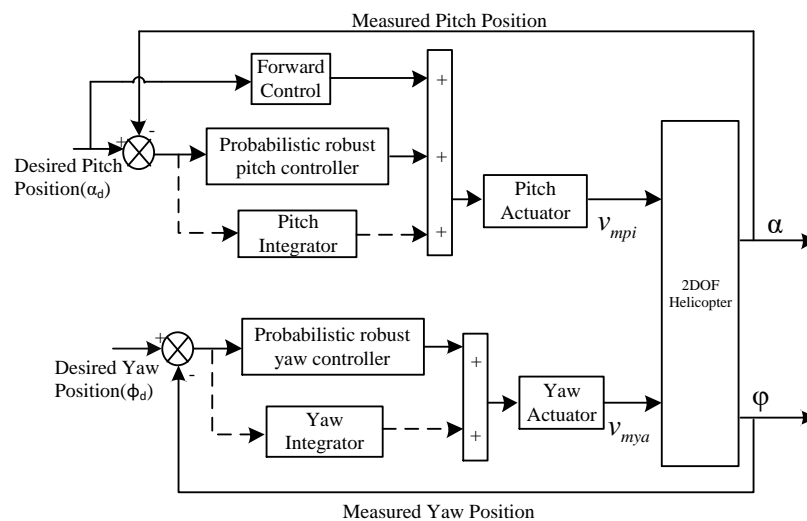


Figure 4.3. Probabilistic Control for Helicopter System.

For a probabilistic solution of the approximate Lyapunov inequalities, a non-deterministic model is adapted. This defines the system uncertainty which can be further used define the performance index for identifying the required gain matrix. For a standard quadratic cost function [352], the performance index is given as

$$J = \int_0^{\infty} x^T(t)Sx(t) + u^T(t)Ru(t)dt \quad (4.15)$$

Where  $S, R$  - weighting matrices and  $S = S^T > 0$ , and  $R = R^T > 0$ . The state feedback law is



$$U(t) = -R^{-1}B^T P^{-1}x(t) \quad (4.16)$$

Here,  $P = P^T > 0$  provides guaranteed stability with a common solution to the Lyapunov inequalities

$$A(\Delta)P + PA(\Delta)^T - 2B(\Delta)R^{-1}B(\Delta)^T < 0, P = P^T > 0, \quad \forall \Delta \in \mathbb{D} \quad (4.17)$$

Further, the effect of structured nonlinear uncertainty  $\Delta \in \mathbb{D}$  on the state matrices  $A(\Delta)$  and  $B(\Delta)$  is identified to estimate the solution for an infinite set of Lyapunov inequalities. Here the randomized algorithm-based sub gradient iteration is initiated. This sequential approach [153] involves two stages, checking the feasibility and uncertainty by a random vector sample  $\Delta_k \in \mathbb{D}$ , and defining the single Lyapunov inequality for each step by  $\Delta_k$  using the sub gradient update. Further, the controller gain synthesis using a randomized algorithm for probabilistic robust controller design is calculated using Algorithm 4.2.

---

**Algorithm 4.2:** Controller gain synthesis with with randomised algorithm [153]

---

*Step 1:* Initialize the process by defining maximum number of iterations to  $N_{max}$ ,  $k = 0$ , and by choosing a symmetrical initial condition  $P^1 = P^0 > 0$

*Step 2:* Initiate the probabilistic approach,  
**for**  $k = 1$  to  $N_{max}$   
Generate a random vector sample  $\Delta_k \in \mathbb{D}$  with uniform distribution  
**for**  
 $P = P^k$ ,  
if  $A(\Delta)P + PA(\Delta)^T - 2B(\Delta)R^{-1}B(\Delta)^T \geq 0$ , set  $\Delta_k = \Delta^{(i)}$  and go to step 3  
**else if**  
 $A(\Delta)P + PA(\Delta)^T - 2B(\Delta)R^{-1}B(\Delta)^T \leq 0$ , return  $P_{Rand} = P^k$ ,  
 $K_{Rand} = -R^{-1}B^T P_{Rand}^{-1}$  and exit.  
**end**  
**end.**

*Step 3:* A new candidate solution  $P^{k+1}$  is obtained according to sub gradient update based on  $P^k$  and  $\Delta_k$ .

*Step 4:* Iterate the steps by setting  $k = k + 1$  and continue from Step 2.

---

This achieves the Lyapunov solution based on random algorithm, such that for the helicopter system with integrator

$$P_{Rand,i} = \begin{bmatrix} 289.2741 & 0 & 0 & 0 & 0 & 0 \\ 0 & 226.9569 & 0 & 0 & 0 & 0 \\ 0 & 0 & 172.731 & 0 & 0 & 0 \\ 0 & 0 & 0 & 283.6118 & 0 & 0 \\ 0 & 0 & 0 & 0 & 50.0000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 50.0000 \end{bmatrix} \quad (4.19)$$

Further, the gain matrix of the controller for operating the 2DoF helicopter system is

$$K_{Rand,i} = \begin{bmatrix} 22.1V/rad & 2.08V/rad & 10.4V.s/rad & 1.66V.s/rad & 7.04V/(rad.s) & 0.696V/(rad.s) \\ -2.3V/rad & 22.2V/rad & -0.569V.s/rad & 14.4V.s/rad & -0.696V/(rad.s) & 7.04V/(rad.s) \end{bmatrix} \quad (4.20)$$

#### 4.1.4.1 Simulation Results and discussion

To assess the operation of the developed RABPC, the controller gain matrices obtained in the above section for unmanned helicopter operating are considered. The simulation analysis is carried out by considering a square trajectory to identify the effectiveness of the developed controller. It is observed that a simulation model of the helicopter system responded efficiently to RABPC controller. The results of the numerical simulation for the helicopter following the square trajectory are shown in Figure 4.4 (a-d).

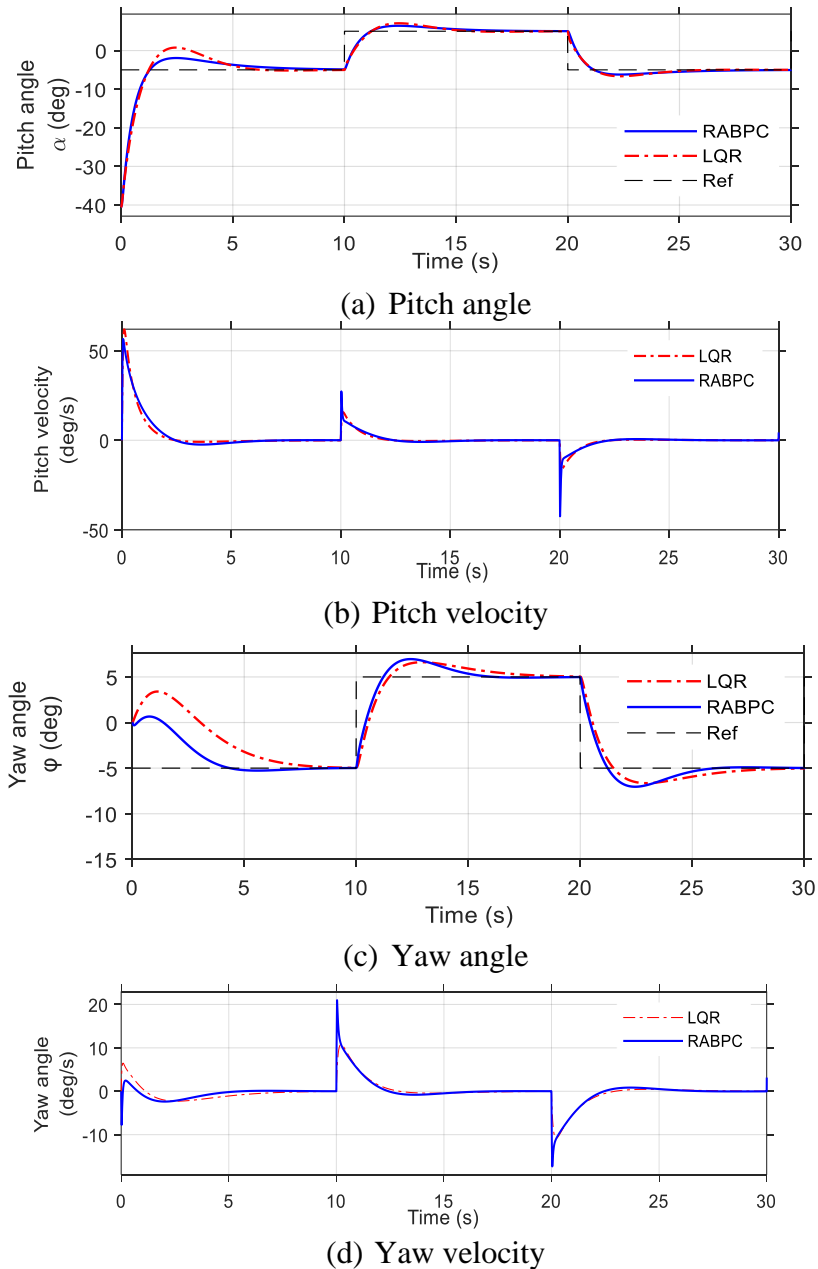


Figure 4.4. Pitch and yaw output with randomised algorithm based probabilistic control of helicopter in Simulink

The superiority of the developed RABPC is identified by comparing the results with the simulation results of the LQR controller implemented on the same system following same trajectory. The results in the figures identify the pitch and yaw responses based on angle, velocity, and voltage characteristics for RABPC and LQR controllers acting on helicopter system. For the helicopter system tracking the sine trajectory with RABPC, the pitch angle trajectory reaches -2.5 degree at 2.3 secs and settles with the desired trajectory before 7 secs. Similarly, the yaw angle reaches 2.2 degrees at 0.4 secs and settles with the trajectory within 7 s. Further, the pitch and yaw velocity are maintained at sufficient level for balancing the helicopter while following the desired trajectory. Besides, less transients in motor operating voltages are observed which identifies the effectiveness and steady state tracking of the helicopter. For the same scenario identified with the LQR controller, the system underperforms resulting in deviation from the path, oscillations, and high voltage transients. From the observations it is identified that, the developed RABPC controller outperforms the classical controller in all the characteristics. Further, the root mean square error has been calculated for the pitch and yaw angle of the helicopter system tracking the square trajectories under both RABPC and LQR controllers is shown in Table 4.1. Besides, a comparative assessment for the performance of pitch and yaw control by both the controllers on the simulated conditions of the helicopter system are shown in Table 4.2.

Table 4.1. Root mean square error for randomised algorithm based probabilistic control of helicopter in Simulink

Controller	Root mean square error	
	Pitch (deg)	Yaw (deg)
LQR	3.9165 deg	1.9217 deg
RABPC	3.1489 deg	1.6114 deg

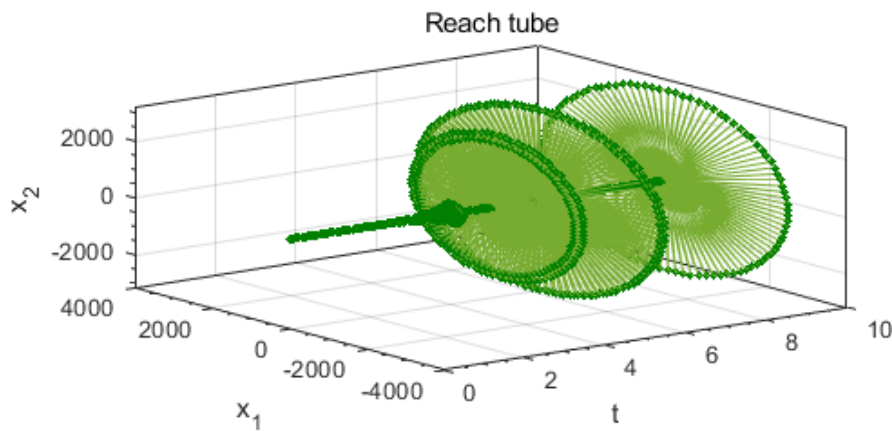
Table 4.2. Time response analysis for randomised algorithm based probabilistic control of helicopter in Simulink

Controller	Pitch		Yaw	
	Settling time ( $t_s$ ) (sec)	Steady state error ( $e_{ss}$ ) (cm)	Settling time ( $t_s$ ) (sec)	Steady state error ( $e_{ss}$ ) (cm)
LQR	6.3 sec	-0.522 cm	8.32 sec	1.38 cm
RABPC	6.1 sec	0.000209 cm	4.68 sec	0.452 cm

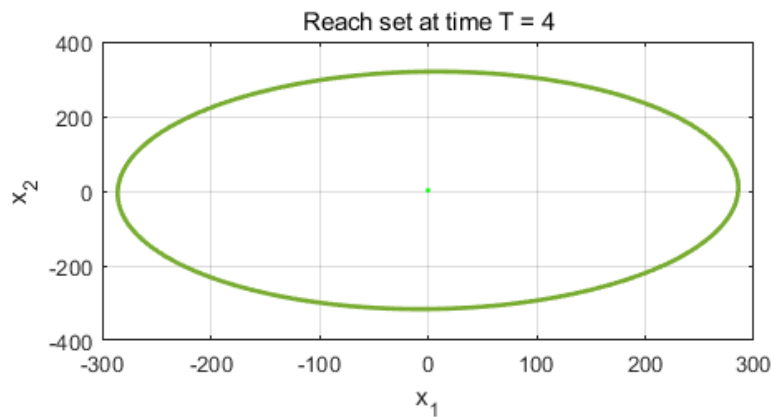
From the observation of the time response characteristics the average settling time, and steady-state error are minimum in case of RABPC which are 6.1 s, and 0.000209 cm for pitch control, and 4.68 s, and 0.452 cm for yaw control tracking the square trajectory.

#### 4.1.4.2 Results of reachability analysis

To test the robustness of the developed trajectory tracking approach, the reachability analysis for the nonlinear systems discussed in section 4.1.3 is performed. Initially, the testing phenomenon is achieved for open loop and closed loop system by defining the RABPC control bounds as per the ellipsoidal calculus for the helicopter system. Further, the system is linearized, and the time interval is defined for the obtaining the reach tube sets. Based on the initial conditions, the directions required for randomizing the uncertainties and initializing the reach set are specified. For a specified reach set, the orthogonal basis of subspace  $(x_1, x_2)$  are defined such that the projection of the reach set is obtained as shown in the Figure 4.5 and 4.6.

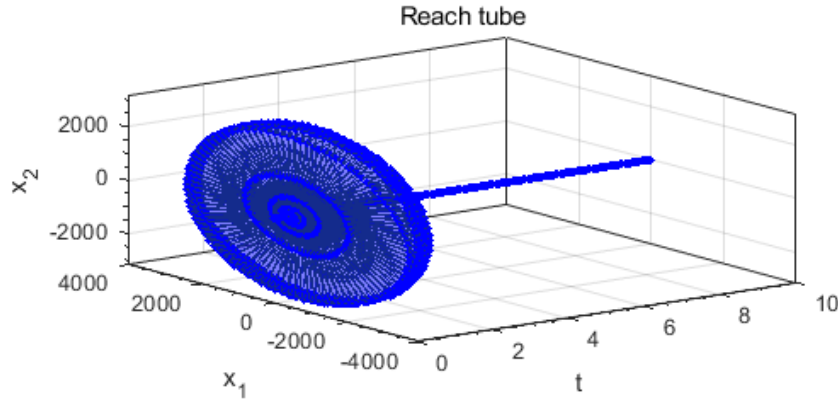


(a) Ellipsoidal reach tube projection on subspace  $t \in [0,4]$

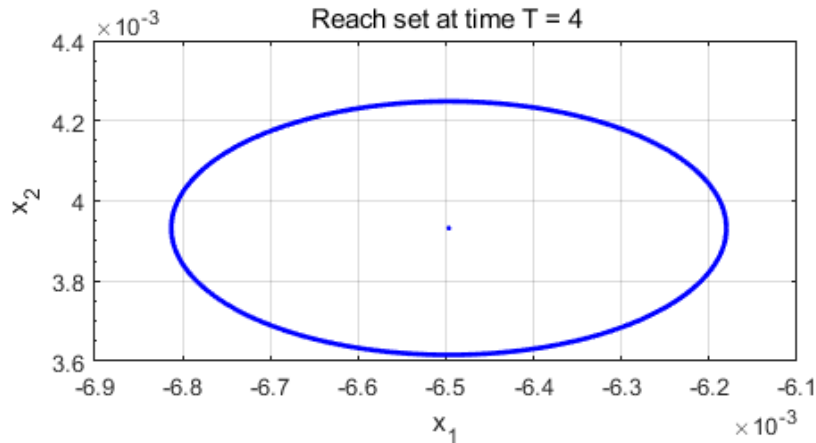


(b) Reach approximation at time  $t=4$  sec for open loop system

Figure 4.5. Reachability analysis for 2DoF helicopter system with LQR controller



(a) Ellipsoidal reach tube projection on subspace  $t \in [0,4]$



(b) Reach approximation at time  $t=4$  sec for open loop system

Figure 4.6. Reachability analysis for 2DoF helicopter system with RABPC controller

In Figure 4.5, the reachability set corresponding to the state space representation of the open loop system is identified. Figure 4.5(a) corresponds to whole ellipsoidal reach tube projection on subspace  $t \in [0, 4]$ , and Figure 4.5(b) corresponds to reach approximation at time  $t=4$  sec. Further, the reachability set corresponding to the state space equation after applying RABPC is given in Figure 4.6. The Figure 4.6(a) corresponds to whole ellipsoidal reach tube projection on subspace  $t \in [0, 4]$ , and the Figure 4.6(b) corresponds to reach approximation at time  $t=4$  sec. Further, it is identified that for every step  $k$  the control law steers the state of the system to the centre point of the ellipsoidal reach set. The closer the linearized system gets to the centre point, the smaller is the linearization error. In Figure 4.5(a), (b) the system is observed to be reaching the centre with large linearization error due to multiple disturbances, but in Figure 4.6(a), (b), the RABPC steers the system towards the centre point of ellipsoid reach set with less linearization error. This depicts the robustness of the developed controller on the nonlinear system. The over-approximated reachable set is contained in the target set  $T$  after 230-time steps and terminates in 463.0367 secs. The absolute and relative tolerance of the systems before and after the application of RABPC is observed to be  $1e - 05$  and  $1e - 06$  respectively.

#### 4.1.5 Real-time Experiment

The experimental system of 2 DoF helicopter system discussed in Chapter 2 is adapted with the controllers designed in Simulation. The interconnection between the real-time helicopter and the simulated controller is achieved with QUARC/MATLAB integration. The control algorithms have been implemented using MATLAB 2015b. To evaluate the performance of the controller's outline, the tracking control of RABPC is compared with the classical LQR.

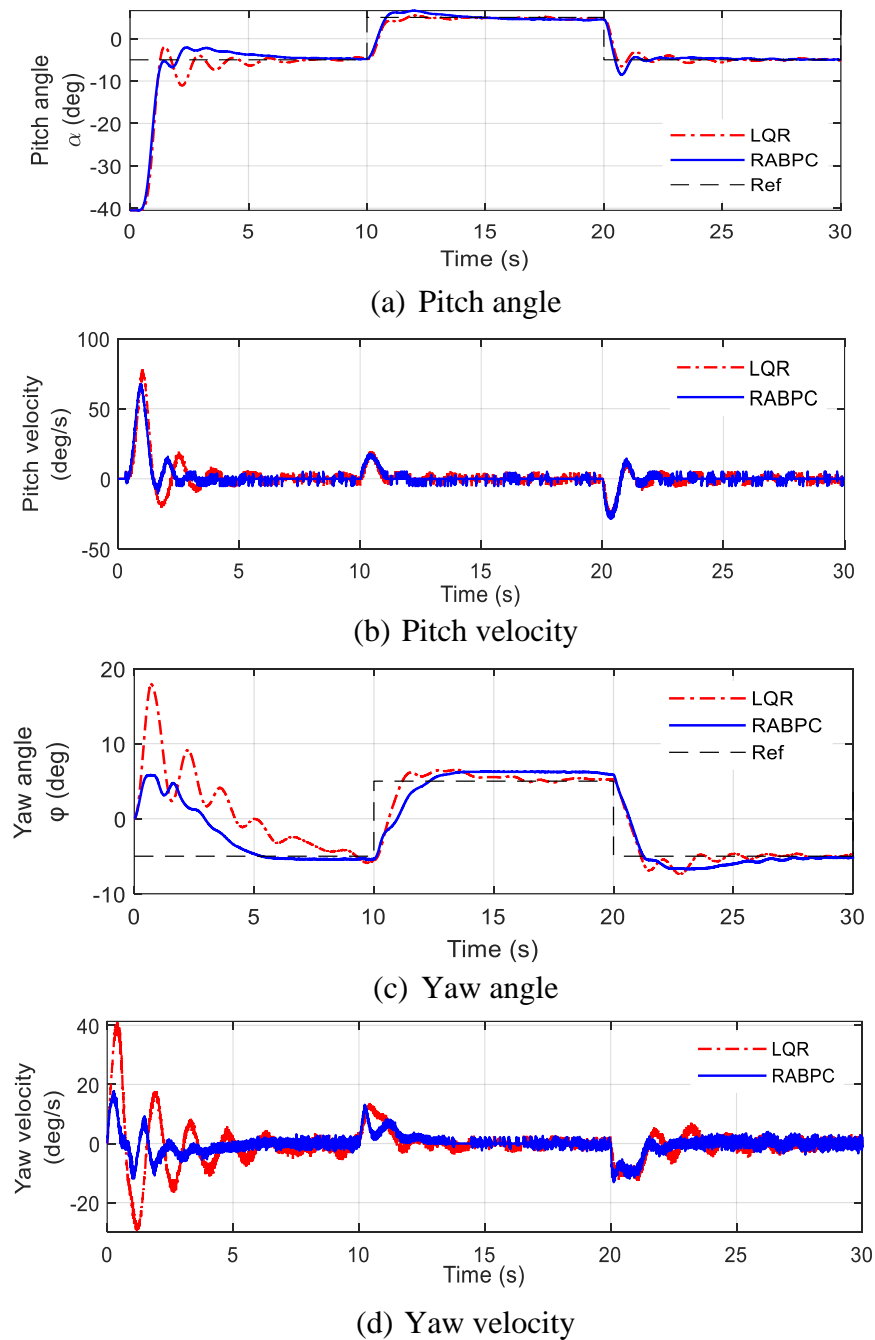


Figure 4.7. Pitch and yaw output with randomised algorithm based probabilistic control of helicopter in real-time

The results in the Figures 4.7 identify the pitch and yaw responses based on angle, velocity, and voltage characteristics for RABPC and LQR controllers acting on the real-time operating helicopter system with and without integrator following sine and square trajectories. For the helicopter system tracking the square trajectory with RABPC, the pitch angle trajectory reaches -5.1 degrees at 1.3 secs and follows the desired trajectory before 7 s. Similarly, the yaw angle reaches 4.8 degrees at 0.9 secs and settles along the desired trajectory within 6 s. Further, the pitch and yaw velocity are maintained at sufficient level for balancing the helicopter while following the desired trajectory. For the same scenario realised with the LQR controller, the system underperforms resulting in deviation from the path, oscillations, and high voltage transients. From the observations it is identified that, the developed RABPC controller outperforms the classical controller in all the characteristics. Further, the root mean square error has been calculated for the helicopter system tracking the square trajectory as shown in Table 4.3. Besides, a comparative assessment for the performance of pitch and yaw control by both the controllers on the simulated conditions of the helicopter system are shown in Table 4.4.

Table 4.3. Root mean square error for randomised algorithm based probabilistic control of helicopter in real-time

Controller	Root mean square error	
	Pitch (deg)	Yaw (deg)
LQR	7.3214 deg	5.3146 deg
RABPC	6.7695 deg	4.9175 deg

Table 4.4. Time response analysis for randomised algorithm based probabilistic control of helicopter in real-time

Controller	Pitch		Yaw	
	Settling time ( $t_s$ ) (sec)	Steady state error ( $e_{ss}$ ) (cm)	Settling time ( $t_s$ ) (sec)	Steady state error ( $e_{ss}$ ) (cm)
LQR	8.31 sec	4.91 cm	10.06 sec	4.31 cm
RABPC	6.89 sec	1.61 cm	5.81 sec	3.91 cm

From the observation of the time response characteristics the average settling time, and steady-state error, are minimum with 6.89 sec, and 1.61 cm for pitch control and 5.81 sec, and 3.91 cm for yaw control with RABPC acting on helicopter tracking the square trajectory. From the results, it has been observed that the response generated by the RABPC restricts the variations to the smaller range when compare to classical LQR controller, and stabilizes the model efficiently.

## 4.2 SIMULTANEOUS PERTURBATION AND STOCHASTIC APPROXIMATION-BASED CONTROL SYSTEM FOR BALL BALANCER SYSTEM

To improve the performance of a closed loop system, achieve online tuning, and deal with random uncertainties, a simultaneous perturbation stochastic approximation (SPSA) algorithm which recursively generates and estimates along random directions has been designed [276], [277], [353], [354]. Conventionally, the stochastic approximation methods were used for statistical computations and later emerged as a separate field of control theory [355]. In the initial stage, these methods were proven for minimization of stationary functionals [281]. Later, the drawbacks of the gradient and newton methods while dealing with time-varying functionals due to the known bounds resulted in the development of stochastic approximation algorithms [282]. But the issue of constant step size in stochastic approximation limited their applications to time-varying systems, tracking and position control problems [356], [357]. Further, these drawbacks are overcome by developing a distributed asynchronous stochastic approximation algorithm [358]. Apart from the development, the constant step size stochastic approximation method has been used in the presence of arbitrary noises and stochastic disturbances with multi-agent systems under dynamic state changes [359]. Considering the flexibility and advantages of SPSA, this chapter develops an adaptive control strategy for solving the problem of balancing control in a closed loop system with unknown but bounded disturbances. The proposed approach is aimed at achieving a finite bound of residual between estimates and time-varying unknown parameters when observations are made under an unknown but bounded noise. This provides an intuitive tuning method for the various controllers, and achieves balancing control for the closed loop system by updating the adaptive parameters in real-time.

### 4.2.1 *Simultaneous Perturbation Stochastic Approximation*

#### 4.2.1.1 Problem statement

Initially, a typical closed loop system is described for formulating a task that can be characteristically solved by SPSA. The closed loop system consists of a plant and a controller. A sequence of uncertainties expressed as  $w_n$  and  $v_n$  are considered, where  $w_n$  corresponds to indeterministic behavior of internal system and  $v_n$  is an external noise combined with the measured output of the system. This optimization problem can be further divided into online, offline and stochastic classes [360]. In the offline class, the approach is classical whereas for



the online class, a new function is measured for every iteration. This resulted in the need for calculating the average of all the generated cost functions which makes the system complex. Further, the stochastic approach involves one function and it is measurable with noise. Hence, in this research the stochastic approach is considered, where the function  $F(x, w)$  is optimized and measured with noise.

Consider  $F(x, w): \mathbb{R}^q \times \mathbb{R}^p \rightarrow \mathbb{R}^1$  is differentiable by an argument function  $(x_1, x_2, \dots)$  at every instant  $n = 1, 2, 3, \dots$ . The value of optimization function with additive noise  $v_n$  is given as:

$$y_n = F(x_n, w_n) + v_n \quad (4.21)$$

where  $w_n \in \mathbb{R}^p$  corresponds to a sequence of uncontrollable random values with equal but unknown distribution  $P_w(\cdot)$ .

The problem in hand is to construct a sequence of estimates  $\{\hat{\theta}_n\}$  of an unknown vector  $\theta$  using observations  $y_1, y_2, y_3 \dots y_n$ . for minimize an average cost type function  $f(x)$  given as

$$f(x) = \int_{\mathbb{R}^p} F(x, w) P_w(dw) \quad (4.22)$$

Usually, a simple model of observation is considered for the problem of minimization function  $f(\cdot)$  as given in (4.23) which is easily suitable for the proposed scheme.

$$y_n = f(x_n) + v_n \quad (4.23)$$

But to accommodate the sequence of uncertainties with the cost type function, the complicated model of minimizing is given as

$$y_n = w_n f(x_n) + v_n \quad (4.24)$$

for a general model with  $F(x, w) = w f(x)$  [361].

Typically, for any condition in this complicated model, if the distribution  $P_w(\cdot)$  is unknown, the problem lies outside the scope of a classical optimization theory. Further, this condition can be overcome by measuring the function  $F(x_n, w_n)$  with an additive random zero mean disturbance  $v_n \in \mathbb{R}$  which is independent and identically distributed. Hence, by adding an additional component  $v$  to the vector  $w$  which forms  $\bar{w} = \begin{pmatrix} w \\ v \end{pmatrix}$ , the measured function can be rewritten as:

$$\bar{F}(x, \bar{w}) = F(x, w) + v \quad (4.25)$$

This forms a new observation scheme with an additive disturbance which involves an unknown distribution  $P_{w,v}(\cdot)$  instead of  $P_w(\cdot)$ . Further, if the noise added with the measurement doesn't have any significant statistical properties, the complexity of the problem cannot be simplified. Hence from the above considerations the problem formulation can be simplified using the inputs  $x_1, x_2, x_3, \dots, x_n$  and observations  $y_1, y_2, y_3, \dots, y_n$ .

#### 4.2.1.2 SPSA Algorithm

The SPSA algorithm is used for efficient estimation of unknown vectors based on small measurements for application with signal identification and adaptive control. Generally, the SPSA works on selected coefficients [362] with two observations which generate estimations recursively in random directions at each iteration. The flow process of SPSA for optimal design parameter is shown in Figure 4.8.

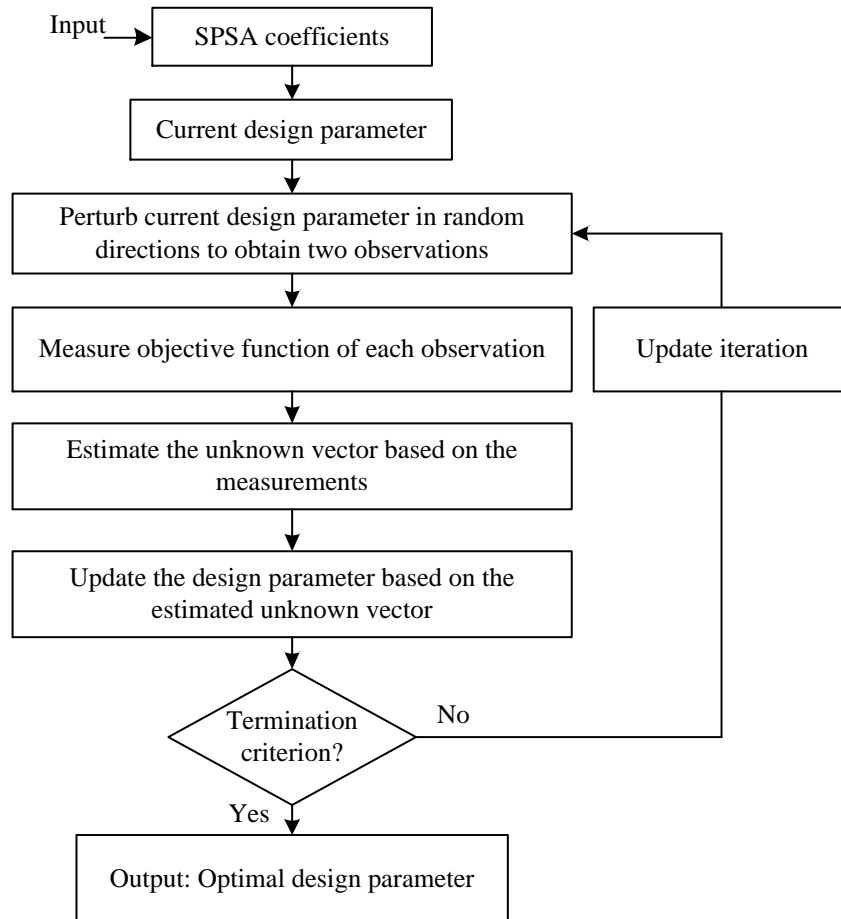


Figure 4.8. Flow process of parameter estimation with simultaneous perturbation and stochastic approximation [277]

Initially, the SPSA algorithms perturbs the current design parameter in random directions and measures the objective function of each observation. The measured observation is used to estimate the unknown vector which updates a new design parameter until the termination criterion is achieved. This provides an opportunity for achieving better convergence towards optimal solution for updating the design parameter. The motivation for using SPSA for adaptive control of a system with unknown but bounded disturbances is due to its easy to implement searching algorithm especially for the real-time control applications. Furthermore, the SPSA has advantages due to its iterative process which implies the idea of online learning

with adaptability of new data and memory saving. The algorithm remains operational while accommodating the growing dimension of the estimated parameters and is resistant to arbitrary external noise at the point of input data. Besides, it exhibits less computation time due to small number of measurements and is capable of solving high dimensional optimization problems. The simultaneous perturbations for building the estimates of unknown vector  $\theta$  are denoted as:

$$\Delta_n \in \mathbb{R}^q \quad (4.26)$$

Generally, the simultaneous perturbation vector is generated using the Monte Carlo approach which provides a two-dimensional random perturbation vector  $\Delta_n$ . Besides, a zero mean probability distribution is used to generate the components of  $\Delta_n$  independently. The common choice for all the components of  $\Delta_n$  is the use of  $\pm 1$  Bernoulli distribution with  $\frac{1}{2}$  probability for every  $\pm 1$  outcome.

In addition, two sets of sequence of positive numbers  $\{\alpha_n\}$  and  $\{\beta_n\}$ , where  $\{\alpha_n\}, \{\beta_n\} \rightarrow 0$  and a fixed initial vector  $\hat{\theta}_0 \in \mathbb{R}^q$  are defined. Three different scenarios are considered for estimating  $\{\hat{\theta}_n\}$  and constructing the sequence of points for measurements  $\{x_n\}$  as follows:

*Case 1:* Estimating using one observation.

$$\begin{cases} x_n = \hat{\theta}_{n-1} + \beta_n \Delta_n, y_n = F(x_n, w_n) + v_n, \\ \hat{\theta}_n = \hat{\theta}_{n-1} - \frac{\alpha_n}{\beta_n} \mathcal{K}_n(\Delta_n) y_n, \end{cases} \quad (4.27)$$

*Case 2&3:* Estimating using two observations on each iteration.

$$\begin{cases} x_{2n} = \hat{\theta}_{n-1} + \beta_n \Delta_n, x_{2n-1} = \hat{\theta}_{n-1} - \beta_n \Delta_n, \\ \hat{\theta}_n = \hat{\theta}_{n-1} - \frac{\alpha_n}{2\beta_n} \mathcal{K}_n(\Delta_n) (y_{2n} - y_{2n-1}), \end{cases} \quad (4.28)$$

$$\begin{cases} x_{2n} = \hat{\theta}_{n-1} + \beta_n \Delta_n, x_{2n-1} = \hat{\theta}_{n-1}, \\ \hat{\theta}_n = \hat{\theta}_{n-1} - \frac{\alpha_n}{\beta_n} \mathcal{K}_n(\Delta_n) (y_{2n} - y_{2n-1}), \end{cases} \quad (4.29)$$

From all the three cases, it can be observed that a kernel function  $\mathcal{K}_n(\cdot): \mathbb{R}^q \rightarrow \mathbb{R}^q$  is used to satisfy the simultaneous perturbation distribution  $P_n(\cdot)$ . This condition is denoted as:

$$\int \mathcal{K}_n(x) P_n(dx) = 0, \int \mathcal{K}_n(x) x^T P_n(dx) = I, \quad (4.30)$$

where,  $I \rightarrow I_{q \times q}$  is a unit matrix.

The general form of  $\mathcal{K}_n(\cdot)$  in the situation of uniform testing perturbation is formulated in a special case by Polyak *et al.* [363] corresponding to (4.27) and Spall *et al.* [354] corresponding to (4.28). This makes an assumption about the centralization and independency of the observed noise. This special case defines the rule for distribution of trail perturbation with kernel  $\mathcal{K}_n(\cdot)$  and finite inverse moments as follows:

$$\mathcal{K}_n(\Delta_n) = \begin{pmatrix} \frac{1}{\Delta_n^{(1)}} \\ \frac{1}{\Delta_n^{(2)}} \\ \vdots \\ \frac{1}{\Delta_n^{(q)}} \end{pmatrix} \quad (4.31)$$

Considering the same kernel  $\mathcal{K}_n(\cdot)$ , a new scenario is developed as per (4.29) with constraints on trail simultaneous perturbation distribution [364]. In this paper, the sequence of estimates is constructed from (4.27) by formulating kernel  $\mathcal{K}_n(\Delta_n) = \Delta_n$  with a projection which is given as

$$\begin{cases} x_n = \hat{\theta}_{n-1} + \beta_n \Delta_n, y_n = F(x_n, w_n) + v_n, \\ \hat{\theta}_n = \mathcal{P}_{\Theta_n}(\hat{\theta}_{n-1} - \frac{\alpha_n}{\beta_n} \mathcal{K}_n(\Delta_n) y_n), \end{cases} \quad (4.32)$$

where  $\mathcal{P}_{\Theta_n}$  corresponds to projecting operators on bounded closed convex subsets  $\Theta_n \subset \mathbb{R}^q$ , which contain  $\theta$  and starts from  $n \geq 1$ . The bounded subsets  $\Theta_n$  can extend up to infinity if the  $\theta$  is not known, and for any case if  $\Theta: \theta \in \Theta$  is known, then the bounded set  $\Theta_n = \Theta$ . For some specific cases, the bounded subsets  $\Theta_n$  can construct a decreasing sequence.

#### 4.2.1.3 Conditions for estimation

Further, the estimates are constructed using different scenarios as follows:

Let  $E\{\cdot\}$  be an expectation,  $\|\cdot\|$ ,  $\|\cdot\|_\rho$  and  $(\cdot, \cdot)$  be a norm in  $l_\rho$  space and scalar product in  $\mathbb{R}^q$  for Euclidean norm with  $\rho \in (1, 2]$  which is a general set, and  $\mathcal{F}_{n-1}$ , derived from a set of random values  $(\hat{\theta}_0, \hat{\theta}_1, \dots, \hat{\theta}_{n-1})$  for a  $\sigma$ -algebra of probabilistic events. The estimates are constructed using the predefined scenarios (4.28) and (4.29) as:

$$\bar{w}_n = \begin{pmatrix} w_{2n} \\ w_{2n-1} \end{pmatrix}, \bar{v}_n = \kappa(v_{2n} - v_{2n-1}), \quad (4.33)$$

$$\kappa = \begin{cases} \frac{1}{2} & \text{for (5)} \\ 1 & \text{for (6)} \end{cases} \quad (4.34)$$

$$F_w = \max_{x \in \mathbb{R}^q} E_{w'} \{ E_{w''} \{ \kappa^\rho |F(x, w') - F(x, w'')|^\rho \} \}, \quad (4.35)$$

Similarly, while constructing the estimates using (4.32)

$$\bar{v}_n = v_n, \bar{w}_n = w_n, F_w = E_w \{ |F(\theta, w)|^\rho \}. \quad (4.36)$$

Further, the assumptions required to find the optimal vector  $\theta$  are formulated [365] by considering a function as follows:

$$V(x) = \|x - \theta\|_\rho^\rho = \sum_{i=1}^q |x^{(i)} - \theta^{(i)}|^\rho, \quad (4.37)$$

*Assumption 1:* A unique minimum and (4.38) are considered for the function  $f(x)$  along with a constant  $\mu > 0$  [366].

$$(\nabla V(x), \nabla f(x)) \geq \mu V(x), \forall x \in \mathbb{R}^q \quad (4.38)$$

*Assumption 2:* For all the non-deterministic behaviors of the system, the gradients of the function  $F(\cdot, w)$  must satisfy the condition in (4.39) with a constant  $M > 0$ .

$$\|\nabla_x F(x, w) - \nabla_x F(y, w)\|_\rho \leq M \|x - y\|_\rho, \forall x, y \in \mathbb{R}^q \quad (4.39)$$

*Assumption 3:* A local condition is considered from the Lebesgue integration [367], where,  $\nabla_x F(x, \cdot): \forall x \exists (\exists \text{ is there exists}), \text{neighbourhood } U_x: \forall x' \in U_x \exists$ , and function  $\Phi_x(\cdot): \mathbb{R}^p \rightarrow \mathbb{R}, E_w\{\Phi_x(w)\} < \infty: |\nabla_x F(x', w)| \leq \Phi_x(w)$  for all of  $w$ .

*Assumption 4:* For kernel  $\mathcal{K}_n(\cdot)$  and simultaneous perturbation distribution  $P_n(\cdot), n = 1, 2, \dots$  satisfies the conditions as:

$$\bar{K} = F_w \sup_{n=1,2,\dots} \int \|\mathcal{K}_n(x)\|_\rho^\rho P_n(dx) < \infty, \quad (4.40)$$

$$\tilde{K} = \sup_{n=1,2,\dots} \int \|\mathcal{K}_n(x)\|_\rho \|x\|_\rho \|x\|_{\frac{\rho}{\rho-1}} P_n(dx) < \infty. \quad (4.41)$$

*Assumption 5:* For every value of  $n > 1$

$$\xi_n = \|E\{\mathcal{K}_n(\Delta_n) \bar{v}_n | \mathcal{F}_{n-1}\}\|_\rho^\rho \leq C_{\Delta v} \beta_n^2, E\{\|\mathcal{K}_n(\Delta_n) \bar{v}_n\|_\rho^\rho\} \leq \sigma_n^\rho. \quad (4.42)$$

Similarly, for  $\rho = 2$  the assumptions 1 and 2 can be formulated as:

*Assumption 1':* The function  $f(\cdot)$  is strictly convex.

$$\langle x - \theta, \nabla f(x) \rangle \geq \mu \|x - \theta\|^2, \forall x \in \mathbb{R}^q. \quad (4.43)$$

*Assumption 2':* For all the indeterministic behaviors of the system, the gradients of the function  $F(\cdot, w): \forall x, \theta \in \mathbb{R}^q$  are satisfied by Lipschitz condition [368]:

$$\|\nabla_x F(x, w) - \nabla_x F(y, w)\| \leq M \|x - \theta\|. \quad (4.44)$$

#### 4.2.1.4 Convergence of Estimates

Further, the convergence of sequence of estimates at a point  $\theta$  is given in the following sense:

*Note 1:* For a linear regression model, the problem of estimating parameters with observations in (4.24) and when  $\theta_n = \theta$  correspond to the minimization of average risk functional.

$$f(x) = \frac{1}{2} (x - \theta)^T (x - \theta). \quad (4.45)$$

*Note 2:* Even though the scenarios in (4.28) and (4.29) are similar, the usage of (4.29) for real-time systems during arbitrary noise in observations is better. This is because of the repetitive moment of a vector  $2n - 1$  in the system of scenario (4.28) restricts the independency of noise  $v_{2n}$  from trail perturbation  $\Delta_n$ . Where as in (4.29) the vector of trail perturbation  $\Delta_n$  and noise  $v_{2n}$  simultaneously enter the system allowing them to hope on their independency.

*Note 3:* For generalization of conditions for convergence of scenarios in (4.28), (4.29), and (4.32), sequence of positive numbers  $\{\alpha_n\}$  and  $\{\beta_n\}$  can be measured randomly, and  $\sigma$ -algebra  $\mathcal{F}_n$  can be measured relatively. This is necessary for instances where quality of the estimation is being assessed. Depending upon the quality of the estimation, the speed of sequence convergence is either lowered to zero or expanded to a bigger value.

*Note 4:* For generalization of the properties of noise, i.e., the existence of  $\rho \in (1,2]$ -values for  $w$ , indicates that the SPSA algorithms can be used for all sorts of adaptive control.

#### 4.2.1.5 Estimation Evaluation

Since the system is designed for unknown but bounded disturbance case, its behavior under relatively high noise level needs to be evaluated. Besides, the concept of estimation evaluation tests whether the estimated parameter is correct for the corresponding dynamics of the system. The following steps outline the estimation evaluation process:

*Step 1:* When a new estimation  $\hat{\theta}_{n_{new}}$  is obtained, discretize the estimated system and set its initial value at a certain time point  $k_0$ .

*Step 2:* Analyze the estimated system over an input from  $[k_0$  to  $(k_0 + LT_s)]$ , where  $L$  is a positive integer indicating the length of evaluation and  $T_s$  is the sampling time. The resulted output is denoted as  $y_e(k)$ , the evaluation output.

*Step 3:* Compare the simulation output with measurements  $x_n(k)$  along the same time sequence. This can be done by calculating their difference as

$$\Delta(\hat{\theta}_{n_{new}}) = \frac{1}{L-L'} \sqrt{\sum_{k=k_0+L'}^{k_0+LT_s} [y_e(k) - x_n(k)]^2} \quad (4.46)$$

Here it should be noted that the simulation starts from  $k_0 + L'$  instead of  $k_0$ , where  $L'$  is the positive integer number. This eliminates the influence of the noisy initial value. Further, the value of  $L'$  is to be selected long enough such that the impact of initial value can be neglected.

*Step 4:* For the condition where the previously or old estimated value is less than the newly estimated value, i.e.,  $\alpha\Delta(\hat{\theta}_{n_{new}}) > \Delta(\hat{\theta}_{n_{old}})$ , the  $\hat{\theta}_{n_{new}}$  is considered to be invalid or at least worse than the  $\hat{\theta}_{n_{old}}$ . Here, the new estimation is abandoned and the old estimation is used. For an otherwise condition where  $\hat{\theta}_{n_{old}}$  is greater than  $\hat{\theta}_{n_{new}}$ , the new estimation is valid and considered as a better estimation for updating the system. In this condition, the  $\alpha$  associated

with the new estimation is a constant positive real number which controls the standard of selection. If the value of  $\alpha \geq 1$ , it indicates strict selection and the only estimation leading to smaller  $\Delta(\hat{\theta}_{n_{new}})$  is accepted. Moreover, all the estimations are accepted if  $\alpha = 0$ . A perfect estimation should lead to  $\Delta(\hat{\theta}_{n_{new}}) = 0$ .

Further, to test the action of SPSA based adaptive control, a balancing problem is considered in closed loop with a proportional integral derivative control. The system description and implementation of developed approach are discussed in further sections.

#### 4.2.2 Stability Analysis

The stability analysis for the developed adaptive control loop is discussed here. Considering the linear matrix inequality-based stability analysis which adapts  $\sigma$  –modification for single input single output system, the expansion for multi-input multi output systems is analyzed. The arbitrary disturbance perturbation is not considered in this condition of closed loop stability analysis with adaptive control. Hence, a simplified state space representation of the 2DoF ball balancer system is considered as

$$\dot{x}_d(t) = A_p x(t) + B_p u(t). \quad (4.47)$$

Let  $A_p'$  and  $B_p'$  be represented as:

$$A' = E_p^{-1} A_p, \quad (4.48)$$

$$B' = E_p^{-1} B_p, \quad (4.49)$$

Where  $E_p^{-1} = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$  [369] and the controller gain  $K_{PID}$  is divided as an integration gain which is given by

$$K_{PID_r} \in \mathbb{R}_{2 \times 2}, \quad (4.50)$$

and state gain

$$K_{PID_x} \in \mathbb{R}_{2 \times 4}, \quad (4.51)$$

where  $K_{PID} = [K_{PID_r}, K_{PID_x}]$

The actual plant model is given as.

$$\begin{aligned} \dot{x}_{d_p} &= A' x_{d_p} + B'(u + W^T \phi(x)) \\ y_d &= c_p x_{d_p} \end{aligned}, \quad (4.52)$$

where,  $W(t) = [W_1(t) \quad W_2(t)] \in \mathbb{R}_{4 \times 2}$  is a matrix of uncertainties and  $\phi(x) \in \mathbb{R}_{4 \times 1}$  is a smooth basis function set.  $W^T(t)\phi(x)$  is a matched uncertainty. Input for the actual plant model is defined as:

$$\begin{aligned}
u &= u_{nom} - u_{ad} \\
u_{nom} &= -K_{PID_x}x + K_{PID_r} \int (r(t) - y)dt, \\
u_{ad} &= \widehat{W}(t)^T \phi(x)
\end{aligned} \tag{4.53}$$

where,  $u_{nom}$  is the nominal input for reference model,  $u_{ad}$  is the adaptive signal, and  $r(t)$  is step reference.

The  $u_{ad}$  acts as a function for cancelling matched uncertainty by estimating the uncertainty matrix  $W(t)$  with  $\widehat{W} = [\widehat{W}_1 \quad \widehat{W}_2] \in \mathbb{R}_{4 \times 2}$ . Further, a reference model corresponding to (4.52) which generates an ideal output given by:

$$\dot{x}_{d_m} = A_m x_m + B_m \int (r(t) - y)dt, \tag{4.54}$$

where,  $A_m = A' - B'K_{PID_x}$ , and  $B_m = B'K_{PID_r}$ .

Consider  $e = x_{d_m} - x_d$  as the tracking error, and  $\widetilde{W}(t) = \widehat{W}(t) - W(t)$  ( $\widetilde{W}_1(t) = \widehat{W}_1(t) - W_1(t)$ ,  $\widetilde{W}_2(t) = \widehat{W}_2(t) - W_2(t)$ ) is the estimation error. Finally, deviation of actual plant from reference model is obtained as:

$$\dot{e} = A_m e + B' \widetilde{W}(t)^T \phi(x) \tag{4.55}$$

Let  $B_1$  and  $B_2$  equal to  $B'$  as  $[B_1 \quad B_2]$ , and from  $B_1, B_2, W_1(t)$  and  $W_2(t)$ , (4.55) can be described as:

$$\dot{e} = A_m e + B_1 \widetilde{W}_1(t)^T \phi(x) + B_2 \widetilde{W}_2(t)^T \phi(x). \tag{4.56}$$

where,  $W_1(t)$  and  $W_2(t)$  can be updated using the adaptive control system with  $\sigma$ -modification.

$$\begin{aligned}
\dot{\widehat{W}}_1 &= -\gamma \phi(x) e^T P B_1 - \sigma \widehat{W}_1 \\
\dot{\widehat{W}}_2 &= -\gamma \phi(x) e^T P B_2 - \sigma \widehat{W}_2
\end{aligned} \tag{4.57}$$

where, the adaptive gain is denoted as  $\gamma$ , ( $\gamma > 0 \in \mathbb{R}$ ) and the  $\sigma$ -modification gain is given by  $\sigma$ . Further, the values of  $P > 0$  satisfies the Lyapunov (4.58) for the values of  $Q > 0$ .

$$P A_m + A_m^T P + Q = 0. \tag{4.58}$$

Further, the values of  $\gamma$  and  $\sigma$  which satisfy the linear matrix inequalities are used for assessing the stability of the system.

### 4.2.3 Simulation

In this section, numerical simulations are developed with the SPSA-PID controller for a closed loop operation of the modelled 2DoF ball balancer system. To begin with, the movement of ball on the plate and the variation in plate balancing angle with reference to  $X$  and  $Y$  axis are measured as plant responses. As the plate angle is adjusted to balance the ball without falling



off its surface, the position of the ball varies accordingly. Since the closed loop control of ball and plate system is being realized by a PID control system, the basic block representation of the closed loop system is depicted in Figure 4.9.

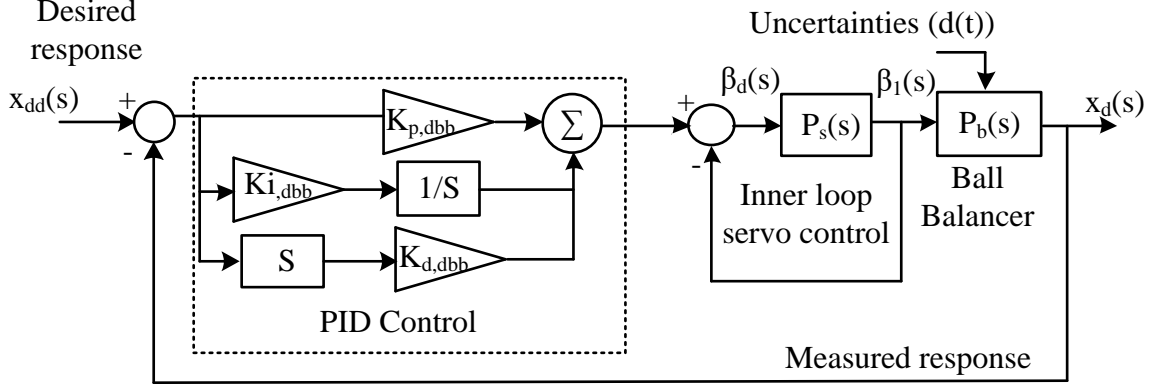


Figure 4.9. Closed loop control of ball balancer system

The closed loop representation deals with a reference  $r(t)$ , controlled input  $u(t)$ , uncertainties  $d(t)$ , and measurement  $y(t)$ . The controller  $K(s)$  for a plant  $G(s)$  is given by:

$$K_{PID}(s) = \begin{bmatrix} h_{11}(s) & h_{12}(s) & \dots & h_{1j}(s) \\ h_{21}(s) & h_{22}(s) & \dots & h_{2j}(s) \\ \vdots & \vdots & \ddots & \vdots \\ h_{i1}(s) & h_{i2}(s) & \dots & h_{ij}(s) \end{bmatrix} \quad (4.59)$$

For a PID controller, the element  $h_{ij}(s)$  is given by:

$$h_{ij}(s) = P_{ij} \left( 1 + \frac{1}{I_{ij}(s)} + \frac{D_{ij}(s)}{N_{ij}(s)} \right) \quad (4.60)$$

Here, the terms  $P_{ij}$ ,  $I_{ij}$  and  $D_{ij}$  correspond to the proportional gain, integral and derivative time of the PID controller and  $N_{ij}$  is the filter coefficient.

In order to assess the performance of the closed loop system depicted in Figure 4.9, the performance index is given by:

$$\begin{aligned} \hat{e}_i &:= \int_{t_0}^{t_f} |r_i(t) - y_i(t)|^2 dt \\ \hat{u}_i &:= \int_{t_0}^{t_f} |u_i(t)|^2 dt \end{aligned} \quad (4.61)$$

Where,  $r_i(t)$ ,  $y_i(t)$ , and  $u_i(t)$  indicate the  $i_{th}$  elements of vectors  $r(t)$ ,  $y(t)$ , and  $u(t)$  respectively,  $t_0 \in \{0\}$  defines a union of positive set of real numbers ( $\mathbb{R}_+$ ) and  $t_f \in \mathbb{R}_+$  which corresponds to the time interval or time period of performance evaluation. Hence the optimization problem of the closed loop control is defined as:

$$J(P, I, D, N) = \sum_{i=1}^p w_{1i} \hat{e}_i + \sum_{i=1}^q w_{2i} \hat{u}_i, \quad (4.62)$$

where  $P := [P_{11}, P_{12}, \dots, P_{ij}]^T$ ,  $I := [I_{11}, I_{12}, \dots, I_{ij}]^T$ ,  $D := [D_{11}, D_{12}, \dots, D_{ij}]^T$ , and  $N := [N_{11}, N_{12}, \dots, N_{ij}]^T$ . The weighting coefficients  $w_{1i}$  and  $w_{2i} \in \mathbb{R}$ , and  $p$  and  $q$  corresponds to the dimensions  $u(t)$  and  $y(t)$  respectively. Theoretically, the performance index of the controller is defined by the sum of ball position error and the controlled input energy. Hence, the problem formulation for the system considered in Figure 4.9 is to find a controller  $K_{PID}(s)$  by minimizing the optimization problem with respect to the values of  $P, I, D$ , and  $N$  by considering the controlled input and measurement data. The implementation of SPSA with the closed loop control of ball balancer system is shown in Figure 4.10.

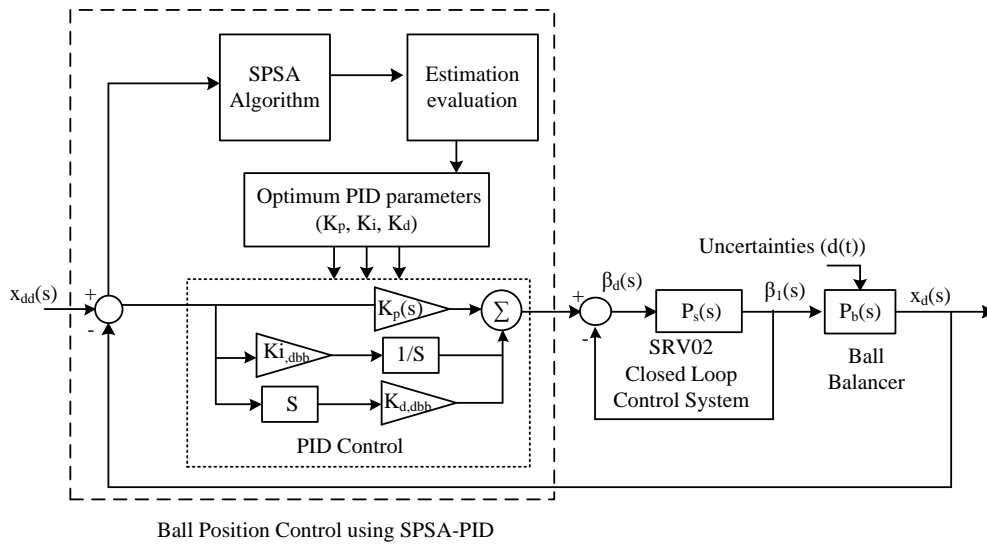


Figure 4.10. Overall structure of ball balancer system with SPSA algorithm and estimation evaluation for optimal PID controller

The steps for implementing the developed controller with the ball balancer system are discussed in algorithm 4.3 as follows:

---

**Algorithm 4.3:** Implementation of the developed SPSA-PID controller

---

- Step 1:* The error between the measured and desired position is obtained and provided as input to both PID controller and the SPSA algorithm.
  - Step 2:* The SPSA algorithm is initialized with the selected coefficients. Here the counter index  $n$  is initially set to 0, unknown parameter vector  $\hat{\theta}_0 \in \mathbb{R}^2$ , and the positive coefficient  $\alpha > 0$ .
  - Step 3:* The iterations are initiated:  $n \rightarrow n + 1$
  - Step 4:* The simultaneous perturbation vector is generated using Monte Carlo and Bernoulli  $\pm 1$  distribution.
  - Step 5:* Estimate the observations.
  - Step 6:* Evaluate the estimated observation by measuring the distance between old estimate and new estimate.
  - Step 7:* If the new estimation is valid, proceed to step 8, or continue with the old estimate and parallelly start estimating the new observation in the next iteration.
  - Step 8:* Update the design parameters to achieve the optimum PID values.
-

---

*Step 9:* Terminate the process.

---

Based on the estimated design parameter, the optimal gains for PID are obtained, such that the controller output varies the plate angle efficiently to balance the ball even in the presence of disturbances. The gain value calculation for the PID controller is obtained as follows:

$$K_{PID}(s) = \begin{bmatrix} h_{11}(s) & h_{12}(s) \\ h_{21}(s) & h_{22}(s) \end{bmatrix} \quad (4.63)$$

Further, to compensate the complexity and minimize the number of iterations for identifying the optimal values, it is considered that some of the gain values of PID has equal attributes as follows:  $I_{12} = I_{11}$ ,  $D_{12} = D_{11}$ ,  $N_{12} = N_{11}$ ,  $I_{21} = I_{22}$ ,  $D_{21} = D_{22}$ , and  $N_{21} = N_{22}$ .

The resultant parameters are calculated by minimizing the objective function  $J$  for values of  $p = 2$ ,  $q = 2$ ,  $w_{11} = 10$ ,  $w_{12} = 1000$ ,  $w_{2i} = 1$  ( $i = 1,2$ ),  $t_0 = 0$  and  $t_f = 20$ .

The response of the 2DoF ball balancer system has been assessed by implementing SPSA-PID method to control the ball position on plate. For analysis purpose a comparative result between conventional PID and SPSA-PID has been shown in Figure 4.11(a) and 4.11(b).

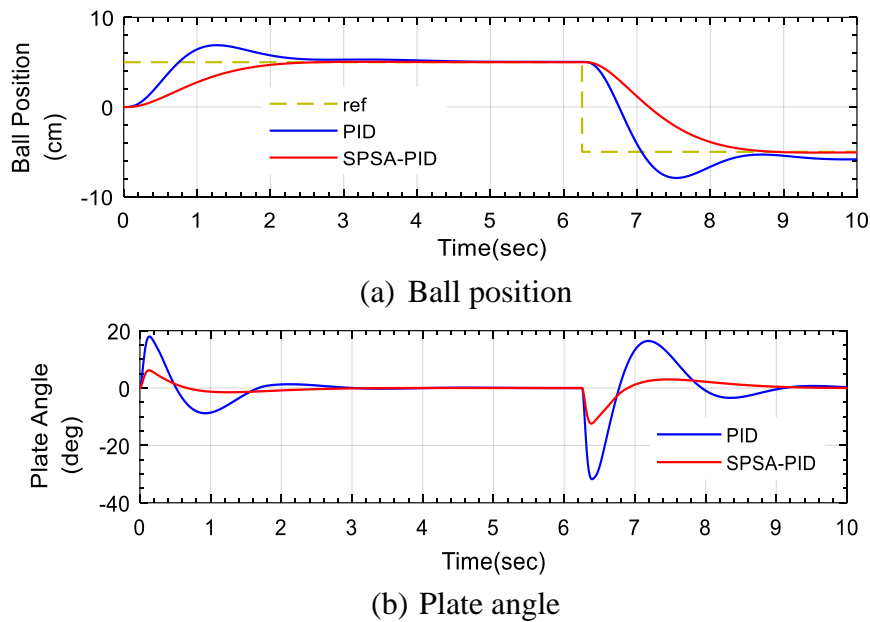


Figure 4.11. Ball balancer output with SPSA-PID control in Simulink

From the Figure 4.11(a), the ball position on plate settled at 2.4 sec without any oscillations and follows the reference square trajectory of amplitude 5. On the other side, the PID takes 4.12 sec to settle with large overshoot as compare to SPSA-PID and takes more time to follow the trajectory. It also has been analysed from the Figure 4.11(b), the plate angle between 5.8 degrees and  $-11.2$  degrees and resulted in less oscillation when controlled with SPSA-PID while PID varies between  $18.3^0$  and  $-31.7^0$ . So, the variation in angle movement is also less in SPSA-PID. Further, it is identified that the amplitude of voltage is also small in case of

SPSA-PID and even small voltage is enough to track and able to balance the ball on the plate. Besides, the time response characteristics of the plate angle output for both the controllers are calculated to estimate the superiority of the proposed approach. Table 4.5 shows the peak time, settling time, peak overshoot and steady-state error of SPSA-PID controller are 0.15 sec, 2.43 sec, 7.32e-06%, and 1.3e-06cm respectively which is less in comparison with PID. As the peak time reduces, the reaching time of the ball to its specified amplitude is minimized. As per the output waveform, it has been noticed that SPSA-PID control has a suitable response with the lowest error and overshoot which carries the perfect ball balance on the plate.

Table 4.5. Time response analysis for SPSA-PID control of ball balancer in Simulink

<b>Controller</b>	<b>Peak Time (<math>t_p</math>) (sec)</b>	<b>Settling Time (<math>t_s</math>) (sec)</b>	<b>Peak Overshoot (<math>M_p</math>) (%)</b>	<b>Steady-state error (<math>e_{ss}</math>) (cm)</b>
PID	0.18sec	3.62sec	1.76%	0.0049cm
SPSA-PID	0.15sec	2.43sec	7.32e - 06 %	1.3e - 06 cm

#### 4.2.4 Real-Time Results

The data exchange between the simulated models and the ball balancer setup is achieved by providing a whole suite of functions which use the features of supported data acquisition hardware from the C language. These functions configure the hardware and perform both synchronous and asynchronous I/O in various forms. The configuration functions provided by the Hardware-in-the-Loop - Application Programming Interface (HIL API) gives the ability to open a hardware-in-the-loop card and configure it. The simplest I/O functions of the HIL API allows single samples to be read or written immediately from the data acquisition card. This form of I/O is called "immediate I/O" and is supported by the Immediate I/O functions of the HIL API. This form of I/O is mainly useful for controlling the system with simulated outputs with various controllers.

As it was mentioned, the HIL API tasks are focused on the acquisition of ball position and D.C. motor positioning. The whole algorithm of ball positioning and control law may be easily implemented and changed in real-time in the running computer simulation, using graphical user interface (GUI). This user-friendly way of operation is ensured by a very simple, but effective data exchange protocol of HIL API- C programming function reference.

Further, the ball balancer system is controlled by applying SPSA-PID using MATLAB and for further interfacing with hardware, a Quarc HIL software has been used.

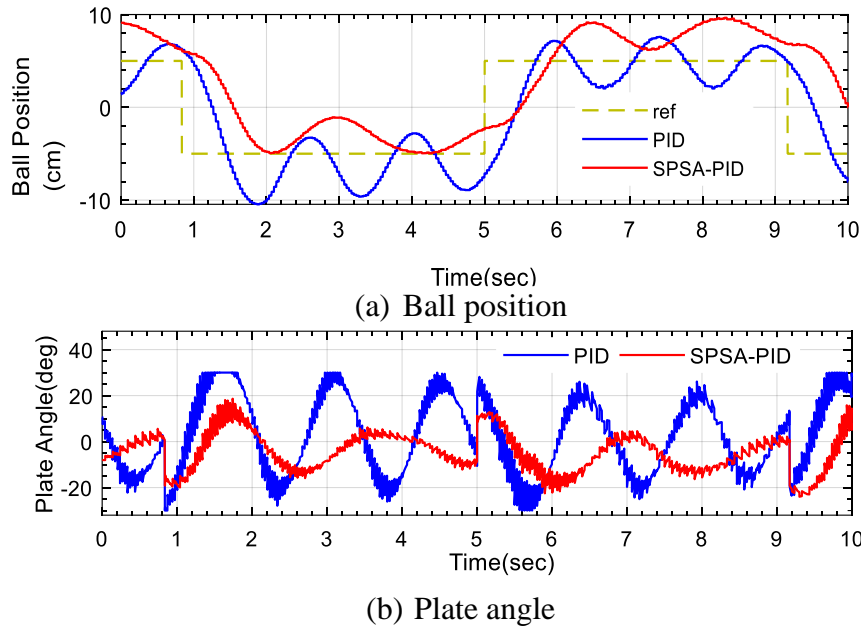


Figure 4.12. Ball balancer output with SPSA-PID control in real-time

Figure 4.12(a), and 4.12(b) show the ball position, and plate angle of the ball balancer system. The square input signal has been given to the setup with the frequency of 0.08Hz and amplitude of 5. From the Figure 4.12(a), the ball position follows the reference trajectory with fewer oscillations which make the system stable and ball moving speed will be slow down. While in PID number of oscillations are more and it settles for the trajectory with more variation in ball movement. Subsequently, angle variation in Figure 4.12(b) is between 18 degrees and  $-23$  degrees which made the movement of the ball stable and smooth for the SPSA-PID controller. From the result it is identified that, as the freedom of movement of angle increase, the ball movement will also increase and causes for instability. Further, the step response characteristics of the ball position are calculated to assess the superiority of the developed approach over the classical approach. As of the Table 4.6, the response of improved SPSA-PID is better based on the peak time, settling time and peak overshoot.

Table 4.6. Time response analysis for SPSA-PID control of ball balancer in real-time

Controllers	Peak time ( $t_p$ ) (sec)	Peak overshoot ( $M_p$ ) (%)	Steady state error ( $e_{ss}$ ) (cm)
PID	1.47 sec	22.9%	1.151 cm
SPSA-PID	0.9 sec	12.4%	0.487 cm

The results proved the effectiveness of adaptive controller using SPSA for position tracking and balancing control of balancer systems.

### 4.3 CONCLUSION

The analysis of parametric uncertainties in the helicopter and ball balancer systems is done by developing a set of probabilistic control approaches with randomized algorithms and stochastic approximations. Both the control actions provided strong control action when the plant is subjected to parametric uncertainty. The randomised algorithm measures the probability over a sample space of uncertain data and generated the independent and identically distributed random variables to avoid worst case behaviour with probabilistic performance verification. Further, it minimized the violation of probability to achieve optimal solution of the recasted problem. The developed algorithm is tested with helicopter motion control when subjected to parametric uncertainty. Similarly, the balancing problem of ball balancers is addressed with the help of SPSA-PID controller. The developed controller overcomes the problem of standard PID gain adjustment when applied for position control of a 2 DoF ball balancer system.

## Chapter 5. FAULT CLASSIFICATION-BASED

### RECONFIGURABLE CONTROL FOR 2DOF SYSTEMS

In this chapter a fault classification based reconfigurable control approach motivated at the operation of the benchmark systems by overcoming the failure modes in their components is developed. Initially, a fault classification-based controller capable of identifying the faults in the helicopter system is developed. Further, the classified fault data is combined with the proposed neural integrated fuzzy controller to achieve the efficient operation of the helicopter in attitude and trajectory tracking. Similarly, the variation in operating characteristics of the ball balancer system for different faults and operating conditions in time and frequency domain are identified with support vector machines. The controller operation here is achieved with the help of a wavelet fuzzy controller.

#### 5.1 FAULT CLASSIFICATION-BASED CONTROL FOR HELICOPTER SYSTEM

In the present-day scenario, helicopter systems are a viable mode of transportation for multiple applications due to their easy take off, and landing at uneven and small places. In most of these applications, the major control objective is to follow a predefined trajectory without deviating from the path [370]. Generally, the unmanned helicopter systems are nonlinear due to the coupling between pitch and yaw motion and were prone to various internal failures and external disturbances, making the control objective a difficult task. Hence, to overcome these conditions, and achieve optimal path tracking, several controllers have been designed in the literature. Besides, the various failure mechanisms that occur during the real-time operation of the helicopter affects the performance of the system making it unstable. So, to achieve operational safety, and provide consistency in operation, the localization of fault and development of fault-tolerant controllers (FTCs) became an essential and structural property [200].

Although, the literature discussed the issues of actuator faults [201], [202], [371], [372], sensor and measurement unit failures [203], effect of uncertainties on attitude tracking and control [373]–[375], and use of machine learning approaches for fault classification [193], [205], [206] individually, the simultaneous control of all these factor with an individual FTC scheme has not been explicitly addressed. In light of the above issues, this section develops a fault classification-based control approach capable of identifying the actuator faults in the system

and achieve optimal performance by combining the classified output with an intelligent controller. The major aspects of this section contribute towards:

- Development of an efficient fault classification approach using the pitch and yaw motor data measured with encoders in the presence of actuator, coupling, and external factor influence based failures.
- Design of an intelligent control approach to achieve attitude tracking of the unmanned helicopter with reference to position error and system operating state (classified state).
- Numerical simulation studies and real-time experiments are carried with the nonlinear system considering the effect of various disturbances under motor failures to validate the developed approach.

These objectives are achieved by adapting wavelet transforms (WTs) and support vector data descriptor (SVDD) for efficient classification of the operating state of the helicopter. Further, the classified state of the system is combined with the proposed neural integrated fuzzy (NiF) controller to achieve the attitude stabilization and trajectory tracking of unmanned helicopters.

#### 5.1.1.1 Feature extraction using wavelet transform

All the measurements corresponding to various operating states of the system can be represented in time or frequency domain and has a set of features which can differentiate between the states. Generally, this data can be analysed using various transform techniques in multiple spectral domain methods. In this paper the wavelet transforms which has the capability of representing the data both in time and frequency domain is adapted [376]. The scale  $a$  and location  $b$  of a wavelet function are defined as:

$$\Psi_{a,b}(t) = \frac{1}{\sqrt{a}} \Psi\left(\frac{t-a}{b}\right) \quad (5.1)$$

For discrete values  $a$  and  $b$ , the logarithmic discretization of the wavelet function can be written as:

$$\Psi_{m,n}(t) = \frac{1}{\sqrt{a_0^m}} \Psi\left(\frac{t-nb_0a_0^m}{a_0^m}\right) \quad (5.2)$$

Where,  $m, n \in \mathbb{Z}$ : are the discrete dilation and translation parameters respectively, and  $a_0$  and  $b_0$  are the fixed dilation and translation step parameters.

Setting  $a_0 = 2$  &  $b_0 = 1$  in (5.2), forms a dyadic grid arrangement for both the dilation and translation steps as follows:

$$\Psi_{m,n}(t) = \frac{1}{\sqrt{2^m}} \Psi\left(\frac{t-n2^m}{2^m}\right) \quad (5.3)$$

Where,  $\Psi_{0,0}(t) = \Psi(t)$  is known as the mother wavelet function.



This dyadic grid arrangement lends itself to the construction of an orthonormal wavelet basis in the  $L^2(\mathbb{R})$  space, that is, a set of vectors which are perpendicular to each other and can completely define a signal  $x(t)$ .

The discrete wavelet transforms (DWT) of a signal  $x(t)$  using dyadic arrangement in (5.3) is given by:

$$T_{m,n} = \sum_{t=0}^{N-1} x(t) \frac{1}{\sqrt{2^m}} \Psi\left(\frac{t-n2^m}{2^m}\right) := \langle x, \Psi_{m,n} \rangle \quad (5.4)$$

Where,  $N$  is the length of the discrete signal  $x(t)$ ,  $T_{m,n}$  is the wavelet (or detail) coefficients, that is, the discrete wavelet transforms values for a scale-location grid of index  $m, n$ .

Similarly, the reconstruction of the original signal  $x(t)$  through the wavelet coefficients  $T_{m,n}$ , using the inverse discrete wavelet transform is given by:

$$x(t) = \sum_{m=1}^M \sum_{n=0}^{2^M-1} T_{m,n} \Psi_{m,n}(t) \quad (5.5)$$

Where,  $M = \frac{\ln(N)}{\ln(2)}$ : is the number of iterations that can be computed. Further, (5.4) and (5.5)

can be summarized as follow:

*Decomposition process:*

$$x(t) \rightarrow \langle x, \Psi_{m,n} \rangle \rightarrow T_{m,n} \quad (5.6)$$

*Reconstruction process:*

$$x'(t) \leftarrow \sum_{m=1}^M \sum_{n=0}^{2^M-1} \langle x, \Psi_{m,n} \rangle \Psi_{m,n}(t) \leftarrow T_{m,n} \quad (5.7)$$

Analogous to the process of obtaining wavelet coefficients  $T_{m,n}$  (5.4), the signal  $x(t)$  can convolve to obtain the approximation coefficients  $S_{m,n}$  at all the scale-location indices  $m, n$  as

$$S_{m,n} = \sum_{t=0}^{2^M-1} x(t) \frac{1}{\sqrt{2^m}} \Phi\left(\frac{t-n2^m}{2^m}\right) := \langle x, \Phi_{m,n} \rangle \quad (5.8)$$

Where,  $S_{m,n}$  is the approximation coefficient for a scale-location grid of index  $m, n$ .

In general, during the decomposition process the energy feature of the signal is extracted. The energy of the wavelet coefficients at the detailed  $g_{dj}$  and the approximations  $g_{aj}$  of the  $j^{th}$  decomposition level can be calculated using (5.9) and (5.10):

$$g_{aj} = \sum_t \left| S_{m,n_j}(x(t)) \right|^2 \quad (5.9)$$

$$g_{dj} = \sum_t \left| T_{m,n_j}(x(t)) \right|^2 \quad (5.10)$$

Further, the reconstruction process is carried out as depicted in (5.7).

As the signal is reconstructed, various features corresponding to the denoised or reconstructed signal are extracted. The mathematical formulations for various features like mean, standard

deviation, skewness, kurtosis, peak to peak distance, total harmonic distortion, signal to noise ratio, power and entropy of a signal are derived as discussed in [377].

### 5.1.2 Support vector data descriptor classifier

Every failure condition of an unmanned helicopter system has a specific impact on its pitch and yaw characteristics. These characteristics when subjected to time and frequency domain analysis provide different features that are the major source of data for training a classifier. Generally, the characterization of such data is affected by the operation of the system at extreme conditions for a very short duration. Training a classifier with such uncertain data will result in misclassification especially during the testing stages. Considering these conditions, the SVDD [378], [379] algorithm which is a class of support vector machines (SVMs) is adapted. The SVDD obtains a boundary around the uncertain data (feature matrix) by computing a hypersphere. This deals with the targeted objects to achieve minimum outlier acceptance by minimizing the sphere volume.

Consider a signal  $i$  with a set of  $d$  real-valued measurements. This provides a feature vector  $X$  for the signal  $i$ , such that  $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,d})$ ,  $x_{i,j} \in \mathbb{R}$ . This indicates that, any given signal can be represented as a single point in the feature space  $\chi$ . Further, let  $f(X; w)$  be a model defined for a hypersphere, where  $w$  is the weight vector. Here all the training samples  $\chi^{tr}$  of the feature space are available in the sphere. This indicates that the empirical error for the training or sample data is 0. Hence, concerning the classical support vector classifiers (SVCs), the structural error is defined as:

$$\varepsilon_{struct}(R, a) = R^2 \quad (5.11)$$

Further, considering different constraints this structural error is minimized as

$$\|X_i - a\|^2 \leq R^2, \quad \forall i \quad (5.12)$$

Here, to achieve a robust classifier and allow the outliers possibility in the training set, the distance from the centre of the sphere ' $a$ ' to the objects  $X_i$  should not be strictly smaller than  $R^2$ . This indicates that the empirical error does not have to be 0. Further, in any case, if an object  $X_i$  lies at a distance  $\xi_i > 0$  outside the area of the sphere, that object is rejected by the description. Hence, the error is contributed both by the empirical and structural errors. To overcome this, the slack variables  $\xi, \xi_i \geq 0, \forall i$  are introduced and the minimization problem changes into:

$$\varepsilon(R, a, \xi) = R^2 + C \sum_i \xi_i \quad (5.13)$$

With a constraint that almost all the objects are within the sphere

$$\|X_i - a\|^2 \leq R^2 + \xi_i, \quad \xi_i \geq 0, \quad \forall i \quad (5.14)$$

Where the tradeoff between the volume of the description and the errors is provided by  $\mathcal{C}$ .

Further, a Lagrangian multiplier ( $L$ ) is used in mapping the constraints in (5.13) with (5.14) as [380]

$$L(R, a, \xi, \alpha, \gamma) = R^2 + \mathcal{C} \sum_i \xi_i - \sum_i \alpha_i \{R^2 + \xi_i - (X_i \cdot X_i - 2a \cdot X_i + a \cdot a)\} - \sum_i \gamma_i \xi_i. \quad (5.15)$$

Where,  $\alpha_i$  and  $\gamma_i$  correspond to the Lagrangian multipliers defined for each signal  $X_i$ . Further, the  $L$  is minimized about  $R$ ,  $a$ , and  $\xi$  and maximized about  $\alpha$  and  $\gamma$ . This is achieved by adding an extra constraint for  $\alpha$  and by setting the partial derivatives to 0. This further limit the signal influence on the final output and provides the final error as

$$L = \sum_i \alpha_i (X_i \cdot X_i) - \sum_{ij} \alpha_i \alpha_j (X_i \cdot X_j) \quad \text{with } 0 \leq \alpha_i \leq \mathcal{C}, \quad \forall i \quad (5.16)$$

Here the expression for the center of the hypersphere is calculated. Further, a new test is elaborated for an object  $z$  that lies within or exactly on the radius accepted by the description as

$$\|z - a\|^2 = (z \cdot z) - 2 \sum_i \alpha_i (z \cdot X_i) + \sum_{ij} \alpha_i \alpha_j (X_i \cdot X_j) \leq R^2 \quad (5.17)$$

$$R^2 = (X_k \cdot X_k) - 2 \sum_i \alpha_i (X_i \cdot X_k) + \sum_{ij} \alpha_i \alpha_j (X_i \cdot X_j) \quad (5.18)$$

Where,  $X_k \in 0 \leq \alpha_i \leq \mathcal{C}$  (set of support vectors with Lagrange multipliers).

Further, this classification process can be given by:

$$f_{SVDD}(z; \alpha, R) = I \left( \|z - a\|^2 \leq R^2 \right) \\ = I \left( (z \cdot z) - 2 \sum_i \alpha_i (z \cdot X_i) + \sum_{ij} \alpha_i \alpha_j (X_i \cdot X_j) \leq R^2 \right) \quad (5.19)$$

Where  $I$  is the identification factor given by

$$I(\mathcal{O}) = \begin{cases} 1 & \text{if } \mathcal{O} \text{ is true} \\ 0 & \text{otherwise} \end{cases} \quad (5.20)$$

Further, it is identified that the rigidness of the hypersphere in defining the data boundary, cannot fit the data as expected. But, to achieve a better fit, the data can be mapped in to a new dimension between the actual boundary of the data and the hypersphere model. Considering this scenario, a mapping  $\Phi$  is developed to improve the data fitting as

$$X^* = \Phi(X) \quad (5.21)$$

By applying this mapping to equations (5.16) and (5.19), the corresponding error is given by:

$$L = \sum_i \alpha_i \Phi(X_i) \cdot \Phi(X_i) - \sum_{ij} \alpha_i \alpha_j \Phi(X_i) \cdot \Phi(X_j) \quad (5.22)$$

and

$$f_{SVDD}(z; \alpha, R) = I \left( \Phi(z) \cdot \Phi(z) - 2 \sum_i \alpha_i \Phi(z) \cdot \Phi(X_i) + \sum_{ij} \alpha_i \alpha_j \Phi(X_i) \cdot \Phi(X_j) \leq R^2 \right) \quad (5.23)$$

From the equations, it can be observed that all the mapping  $\Phi(X)$  can only occur with inner products in other mappings. Considering this, a new function with two new input variables can be defined as follows:

$$K(X_i, X_j) = \Phi(X_i) \cdot \Phi(X_j) \quad (5.24)$$

Where,  $K(X_i, X_j)$  corresponds to a Mercer kernel function representing the inner product of two functions. This modifies  $L$  as

$$L = \sum_i \alpha_i K(X_i, X_j) - \sum_{ij} \alpha_i \alpha_j K(X_i, X_j) \quad (5.25)$$

and

$$f_{SVDD}(z; \alpha, R) = I(K(z, z) - 2 \sum_i \alpha_i K(z, X_i) + \sum_{ij} \alpha_i \alpha_j K(X_i, X_j) \leq R^2) \quad (5.26)$$

Further, to achieve efficient kernel operation, the authors in [381] proposed a kernel trick that replaces the mapping  $\Phi(X)$ . This trick is mainly used in the case of non-linearly separable data while classifying with the SVMs. Besides the kernel trick, several other kernels like Gaussian, linear, polynomial, and radial basis function are also used with the SVCs in the literature [382]. In this research, the Gaussian kernel [383] is used due to its advantages with independency, data positioning, and distance formulation. Mathematically, Gaussian kernel is given as

$$K(X_i, X_j) = \exp\left(-\frac{\|X_i - X_j\|^2}{s^2}\right) \quad (5.27)$$

where  $s$  is the width of the Gaussian kernel.

Hence, based on the developments observed for the SVDD, the classifier is used to train the operating and fault states of the unmanned helicopter in the further sections.

### 5.1.3 Classifier training

#### 5.1.3.1 Data Generation

The helicopter model discussed in Chapter 2 is simulated using MATLAB/Simulink software to identify the normal operation and different motor faults in an unmanned helicopter system. To generate the training data the helicopter model is simulated under normal operating conditions with a conventional PID controller as discussed in [384]. Further, the motor faults are injected in the system as discussed in Table 5.1.

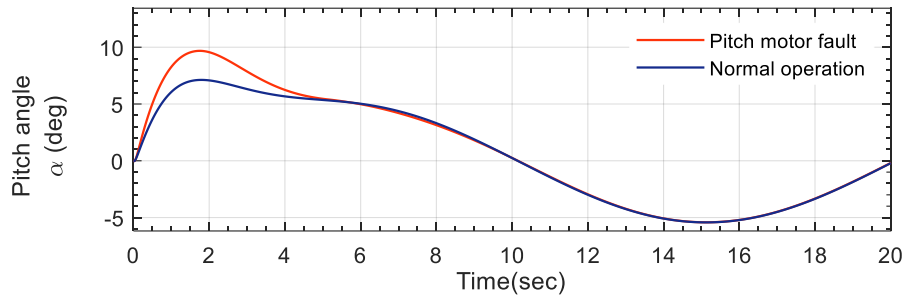
Table 5.1. Simulated conditions on helicopter system for generating data for developing fault classifier

Simulation condition	Component effected	Fault class
The helicopter is controlled by a conventional PID controller to follow the desired trajectory.	N/A	Normal operation

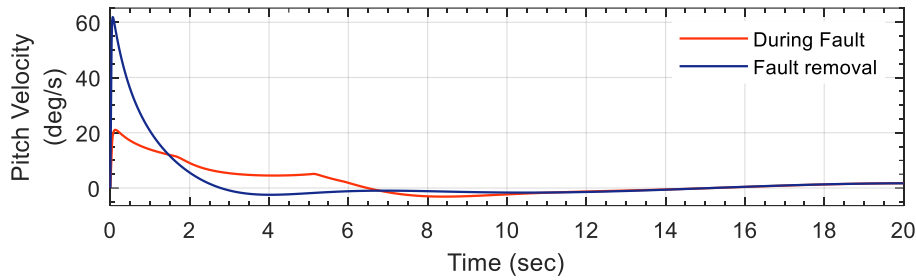
Pitch motor voltage drop resulting in thrust torque altering the position of the helicopter from the desired trajectory	Pitch actuator	Motor fault
Failure of pitch motor as the voltage exceeds the threshold resulting in the abrupt shutdown of the helicopter.		
Effect of transients in yaw motor voltage on the cross torque thrust of pitch motor failing the coupling.		
Yaw motor voltage drop resulting in thrust torque altering the speed of yaw motor and displacing the position of the helicopter from the desired trajectory	Yaw actuator	
Failure of yaw motor as the voltage exceeds the threshold failing the rotational control for the helicopter.		
Effect of transients in pitch motor voltage on the cross torque thrust of yaw motor failing the coupling and rotational control.		

To begin with, the system is operated to follow a sine trajectory by providing an input signal of amplitude 5 and frequency 0.05Hz. Here, the trajectory tracking control objective is achieved with a classical PID controller [384]. During this operation, the pitch velocity, pitch angle, yaw velocity, and yaw angle of the helicopter are measured for two cycles as a time series data. Further, the faults discussed in Table 5.1 are simulated by varying the pitch and yaw motor voltages. With the action of each fault condition, the variations in pitch velocity, pitch angle, yaw velocity, and yaw angle are measured for two cycles as a time series data. The sample of time series representation of pitch and yaw characteristics under normal and fault conditions are shown in Figure 5.1.

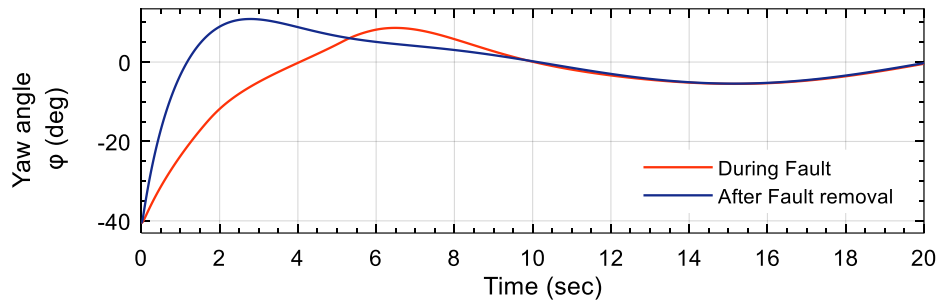
From Figure 5.1, it is identified that the variation in motor operating voltage to inject the fault affects the measured output characteristics of the helicopter. The pitch angle depicts a transient variation up to 9.7degrees during the high voltage fault. This resulted in a huge variation in the vertical movement of the helicopter. Similarly, the low voltage fault effect on the yaw motor fails the motor to achieve a significant yaw angle which resulted in huge misalignment at the pitch and yaw coupling. These deviations when compared with the characteristics of desired trajectories and normal operations showed a huge variation. Further, these variations are used to develop the classifier by extracting the relevant information.



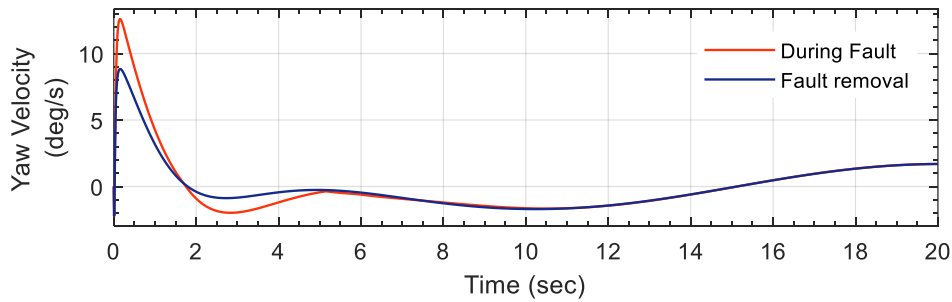
(a) Pitch angle



(b) Pitch velocity



(c) Yaw angle



(d) Yaw velocity

Figure 5.1. Output characteristics of helicopter system for normal and failure conditions

### 5.1.3.2 Feature extraction from helicopter fault data

As the fault threshold may vary considering the operation of the helicopter, the feature extraction aids in developing an efficient classifier using the simulated condition data. Conventionally, many feature extraction techniques for extracting features of 1-dimensional signals are discussed in the literature [385]. Out of these techniques, the WT proved its efficiency due to its advantages with time and frequency representation of signals and the adoption of mother wavelets while using DWT [386].

From the pattern of the signals observed during normal and motor faults, the ‘Daubechies-5 (db5)’ mother wavelet is chosen, with five-level decomposition and reconstruction [387]. During this process, the detailed and approximate coefficients of the signal are calculated and the corresponding energy feature ( $E$ ) is obtained as follows:

$$E = \sqrt{\left[ \sum_{i=1}^{i=N} \frac{abs^2(d_m(i))}{N} \right]} \quad (5.28)$$

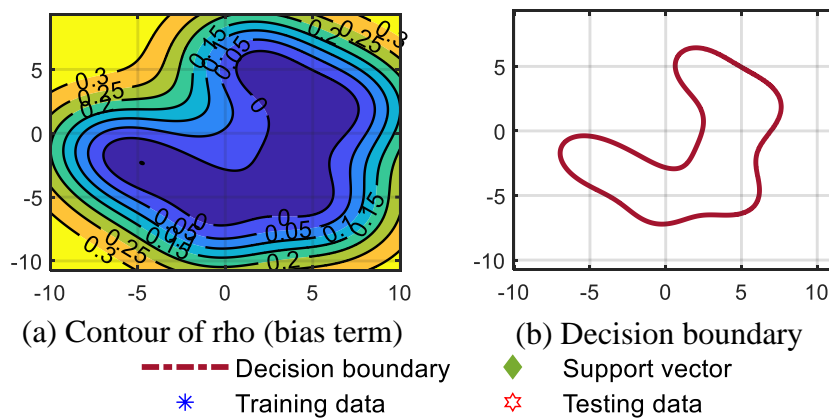
where  $N$  corresponds to the number of wavelet coefficients for each power cycle, and  $d_m(i)$  corresponds to a detailed coefficient at each level of decomposition with  $l \in 1,2,3,4,5$ .

In addition to the above feature, the entropy ( $H(R_s)$ ) is calculated using (5.29) to assess the robustness of the selected wavelet and support the energy feature during the training process.

$$H(R_s) = - \sum_{i=1}^N p(R_s) \log_{10} p(R_s) \quad (5.29)$$

Where  $R_s$  represents the reconstructed form of the original signal, and  $p(R_s)$  is given by the probability of the reconstructed signal.

The features form a matrix along with their corresponding classes for normal and motor fault operation. This accommodates a matrix of  $2000 \times 2$  samples for normal operation and a matrix of  $3000 \times 2$  (six sets of  $500 \times 2$  samples) for motor faults. The forms a matrix of feature vectors with 5000 samples for each feature. Further, the training process is carried out for the data by setting the width for Lagrangian multiplier  $L$  to 0.8 such that  $0 \leq \alpha_i \leq 0.8$ , and the kernel width is set to  $1/16$  by calculating the value of the kernel matrix using (5.27). The decision boundary range is calculated based on the total sample length and number of training samples. This sets the training data grid range, and estimates the bias term  $\rho$ . Here, based on the boundary conditions set, the classifiers trains and tests the corresponding data. The results for classifier training and testing are shown in Figure 5.2.



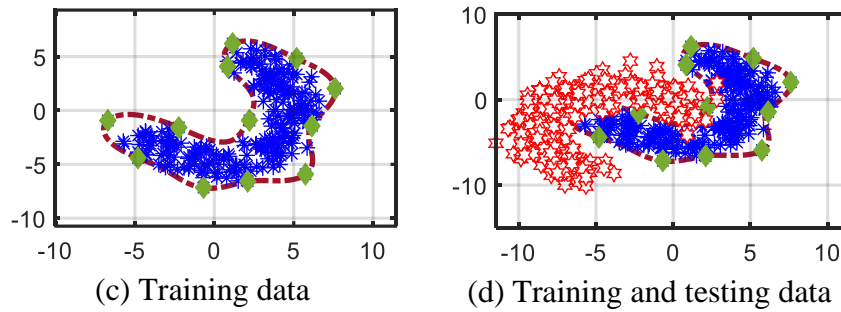


Figure 5.2. Parameter selection and decision boundary formation for training and testing process of the helicopter system fault classifier

Figure 5.2(a) depicts the contour plot while estimating the bias term for defining the region of the training data. From the contour, it can be defined that the data fits in the classifier without excluding any outliers or the delicate details of the data. To get a clear picture of the contour, the grid lines are used as depicted in Figure 5.2(b) to get an exact representation of the hypersphere. In Figure 5.2(c), the training data is scattered in the hyperplane and it is observed that the data is trained to lie within the support vectors. Further, in Figure 5.2(d), the behavior of the classifier with the testing data is depicted. The classifier results are shown in Figure 5.3.

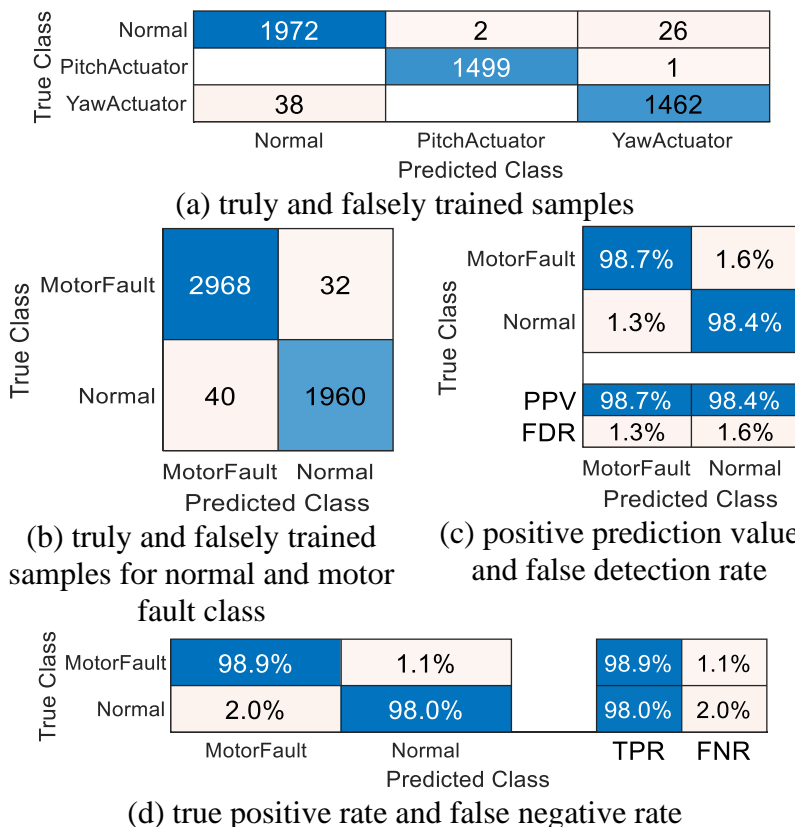


Figure 5.3. Confusion matrix of various faults and trained fault classes of helicopter system classifier

The truly and falsely classified samples of normal operating state, pitch actuator fault, and yaw actuator fault are shown in Figure 5.3(a). The normal operating condition has 1972 truly



classified samples and 28 falsely classified samples, while pitch actuator fault has 1499 truly classified and 1 falsely classified sample. Similarly, yaw actuator fault has 1492 truly classified and 38 falsely classified samples. Further, the truly and falsely classified of the fault sampled under normal and motor fault classes are shown in Figure 5.3(b). The positive prediction value and false detection rate along with true positive rate and false negative rate are shown in Figure 5.3(c) and 5.3(d) respectively. Collectively, the overall training accuracy is identified as 98.6%. Further, the same input data is trained with different conventional classification techniques to identify the superiority of the developed classifier. The corresponding results are discussed in Table 5.2.

Table 5.2. Comparison of classification parameters for helicopter system fault classifiers

Parameters	Model Type		
	SVDD	SVM	K-Nearest Neighbor
Preset			
Network Structure	2 predictors- 2 responses	2 predictors- 2 responses	2 predictors- 2 responses
Feature matrix	$5000 \times 2$	$5000 \times 2$	$5000 \times 2$
Kernel	Gaussian	Fine Gaussian	Coarse K-NN
Accuracy	98.6%	88.6%	81.4%

#### 5.1.4 Fault-tolerant control

The process of developing a fault classification algorithm for unmanned helicopters considering various faults and operating conditions was discussed earlier. Further, the system needs to be controlled using a fault-based control method where the complete system stability and desirable performance is maintained even in the case of external disturbance. Considering this, the classified fault information and the error data have been provided as inputs to the neural integrated fuzzy (NiF) controller.

The NiF controller can be expressed by a single structure that combines the learning capability of artificial neural networks and the inference ability of the fuzzy controller. In the proposed fault tolerant control (FTC), the NiF learns about the state of the system from the data provided by the classifier, and the rule base is trained using the least square estimates and gradient descent method.

##### 5.1.4.1 Neural integrated Fuzzy Architecture

The hybrid algorithm develops a new fuzzy inference that has four inputs and one output and trained with the Sugeno fuzzy model. This model is chosen to train the controller to adapt to the linear dependency of each rule on the input variables. A detailed overview of NiF

architecture along with the Sugeno fuzzy model learning algorithms are detailed in [388]. The architecture of four-input-single output NiF is shown in Figure 5.4.

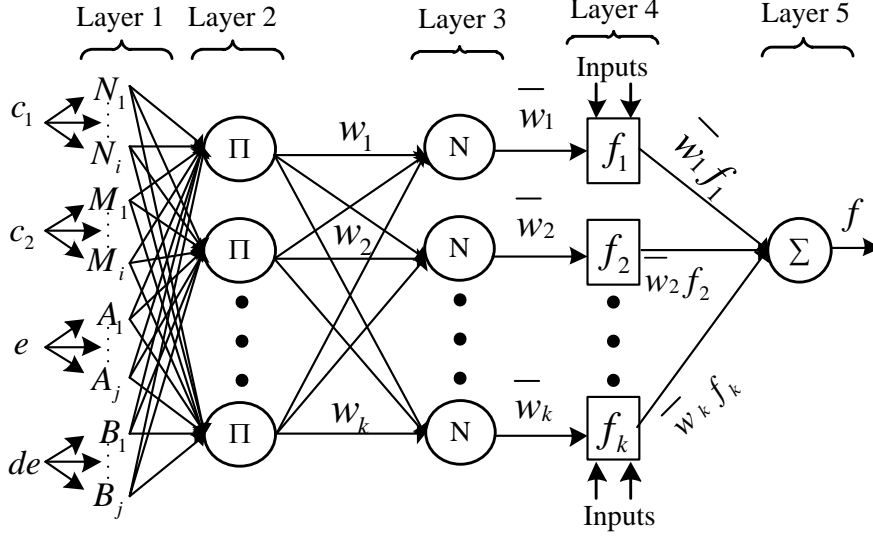


Figure 5.4. Schematic diagram of neural network for neural integrated fuzzy control of helicopter system

The inputs  $c_1$  and  $c_2$  correspond to the normal operating class and motor fault class, and inputs  $e$  and  $de$  corresponds to the error and change in the error of the measured signal. The terms  $N_i, M_i, A_j$ , and  $B_j$  ( $i, j = 1, 2, \dots, n$ ) in layer 1 corresponds to fuzzy membership functions of inputs  $c_1, c_2, e$ , and  $de$  respectively. The nodes in layer 2 which are labeled as  $\pi$  act as any  $T$ -norm operator providing a weighted output  $w_k$  ( $k = 1, 2, \dots, n$ ). Further, the output of the third layer labeled as  $N$  is calculated as  $\bar{w}_k = w_k / \sum_k w_k$ . Finally, the linear functions  $f_k$  ( $k = 1, 2, \dots, n$ ) at the fourth layer act as inputs for representing the ‘then’ part of the fuzzy rules ( $f_k = p_k x + q_k y + r_k$ ). The terms  $p_k, q_k$  and  $r_k$  are constant coefficients to be determined, and  $x$  and  $y$  are the inputs being processed. The final output is calculated from  $f = \sum_k \bar{w}_k f_k$ . Combining all the aspects, rules are formulated as follows:

*Rule 1 if  $c_1$  is  $N_1, c_2$  is  $M_1, e$  is  $A_1$  &  $de$  is  $B_1$ , then output  $f_1 = p_1 x + q_1 y + r_1$*

*Rule 2 if  $c_1$  is  $N_2, c_2$  is  $M_2, e$  is  $A_2$  &  $de$  is  $B_2$ , then output  $f_2 = p_2 x + q_2 y + r_2$*

*Rule  $k$  if  $c_1$  is  $N_k, c_2$  is  $M_k, e$  is  $A_k$  &  $de$  is  $B_k$ , then output  $f_k = p_k x + q_k y + r_k$*

The output  $f$  can be elaborated as:

$$f = \frac{w_1}{w_1 + w_2 + \dots + w_k} f_1 + \frac{w_2}{w_1 + w_2 + \dots + w_k} f_2 + \dots + \frac{w_k}{w_1 + w_2 + \dots + w_k} f_k \quad (5.30)$$

For the NiF implementation of the helicopter model, the design procedure completely depends on a data training method, calculating the input-output membership function, and rule base, to achieve optimum output. The block diagram of the NiF controller implemented with the plant

is shown in Figure 5.5. From the block diagram, it can be identified that the desired and faulty pitch and yaw position are compared with their reference values to generate an error which is further combined with the operating state classified by the classifier to apply as inputs for the controller. Where faulty pitch error  $e_p(t) = \psi - \psi_d$ , a derivative of faulty pitch  $de_p(t)$ , faulty yaw error  $e_y(t) = \varphi - \varphi_d$ , a derivative of faulty yaw is  $de_y(t)$ . Here,  $\psi$  and  $\varphi$  are the pitch and yaw angles of the helicopter, and  $\psi_d$  and  $\varphi_d$  are the desired pitch and yaw angles, respectively.

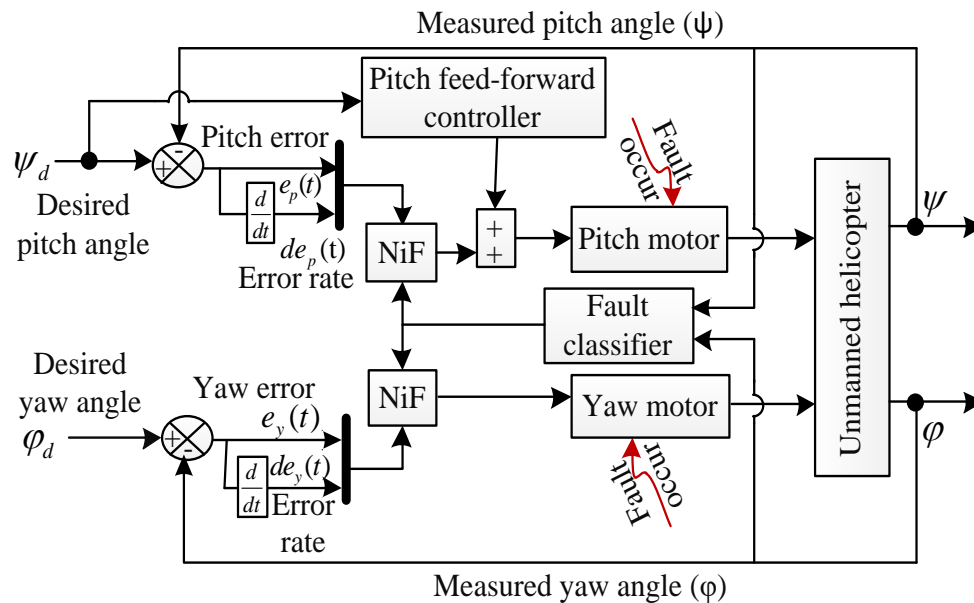


Figure 5.5. Implementation of neural integrated fuzzy control with the helicopter system

### 5.1.5 Experimental testing

To test the developed control approach, the helicopter test rig discussed in Chapter 2 is considered in the laboratory environment. Further, the controller developed is interconnected with the test rig by performing data exchange between the simulated models and the helicopter setup. This is achieved by building the C code through the MATLAB and Quanser software. The flow diagram for software-hardware integration with the proposed FTC scheme is shown in Figure 5.6.

The performance of the proposed controller is assessed by testing it with the helicopter test rig. A motor fault is created on the system and tested with the classifier to identify the operating state of the helicopter. The classified operating state is further combined with the developed NiF controller to operate with the precomputed rule base. The results of the testing and control process are discussed as follows.

The pitch and derivative of the pitch are the two inputs for controller design, and all possible combination of that lies in the range of pitch data is fed for the training of NiF. For the yaw controller design, the same approach is applied. The target output in this scenario is the control signal  $u(t)$ , which obtained from fuzzy control. Further, the universe of discourse is chosen by testing different thresholds in the simulation for maintaining the pitch and yaw characteristics to track the trajectory even in the presence of fault [389]. The universe of discourse is ranged between  $[0\ 1]$  for both the fault classes,  $[-0.2, 0.2]$  for yaw error & yaw derivative,  $[-0.35, 0.35]$  for pitch error, and  $[-1\ 1]$  for pitch derivative.

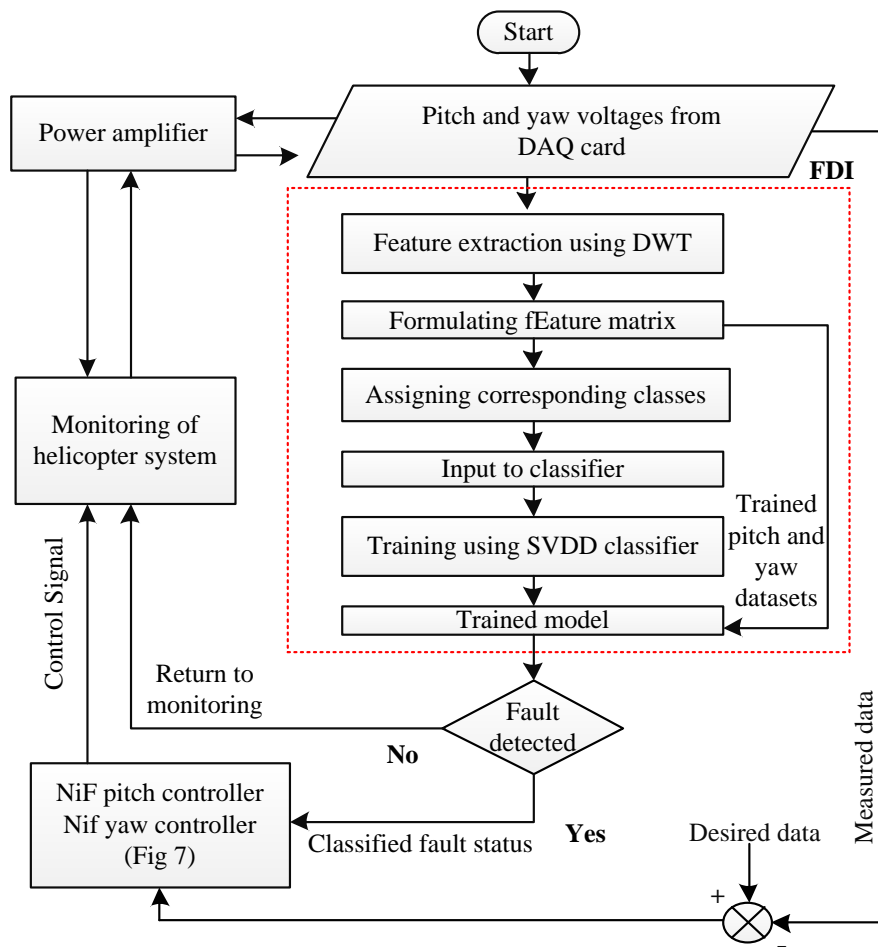


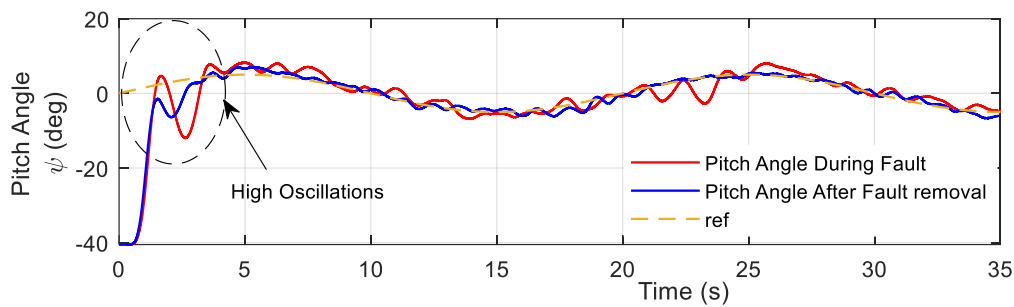
Figure 5.6. General scheme of fault-tolerant controller for helicopter system

The data of classes, yaw, and pitch are trained by a hybrid training algorithm and a Sugeno type fuzzy model with single output has been obtained after training. Each output membership function parameter is associated with the seven different linguistic variables expressed as NB, NM, NL, Z, PL, PM, & PB. These seven-input spaces are further portioned into the 49 fuzzy subspaces. The complete fuzzy subspace follows the if-then rules set. The various factors considered for developing the NiF controller are listed in Table 5.3.

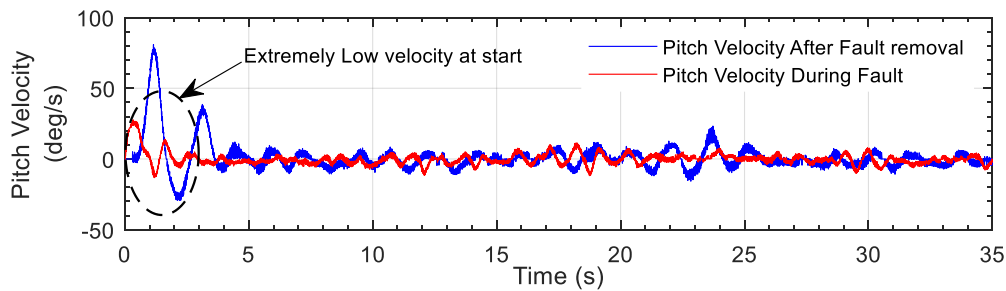
Table 5.3. Parameters of NiF controller for helicopter system

Parameter	Values
Yaw control scaling factor	$M_{e_y(t)} = N_{\dot{e}_y(t)} = 35$
Yaw output scaling factor	$O_{u_y(t)} = 0.7$
Pitch control scaling factor	$M_{e_p(t)} = 20, N_{\dot{e}_p(t)} = 7$
Pitch output scaling factor	$O_{u_p(t)} = 0.7$
Coefficient of learning for pitch	$\eta_p = 5.75352e^{-6}$
Root mean square error for pitch	$P_{rmse} = 0.000006$
Coefficient of learning for yaw	$\eta_y = 6.95252e^{-6}$
Root mean square error for yaw	$Y_{rmse} = 0.000006$

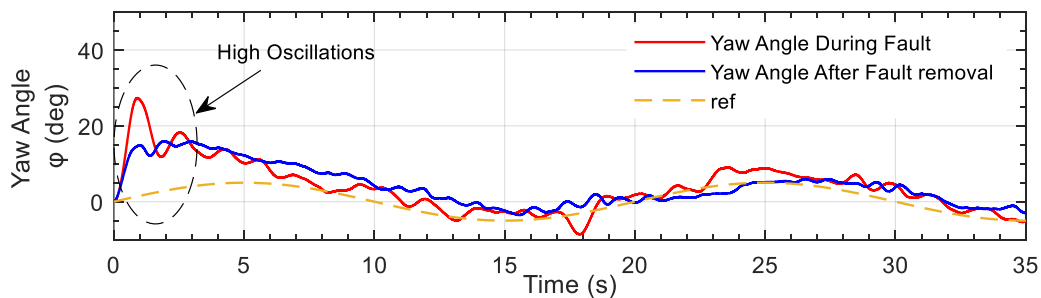
The main objective is to achieve path tracking of a fault effected helicopter by observing a predefined trajectory. A motor fault is injected which creates variation in pitch and yaw motion. This affects the pitch angle, yaw angle, pitch velocity, and yaw velocity. Initially, when the fault occurs, a huge number of oscillations is observed on the helicopter body during its run time. These high oscillations are due to the relatively strong force that lifts the helicopter body at the starting time. As the fault is removed, a more stable and oscillation free-response is achieved by the system.



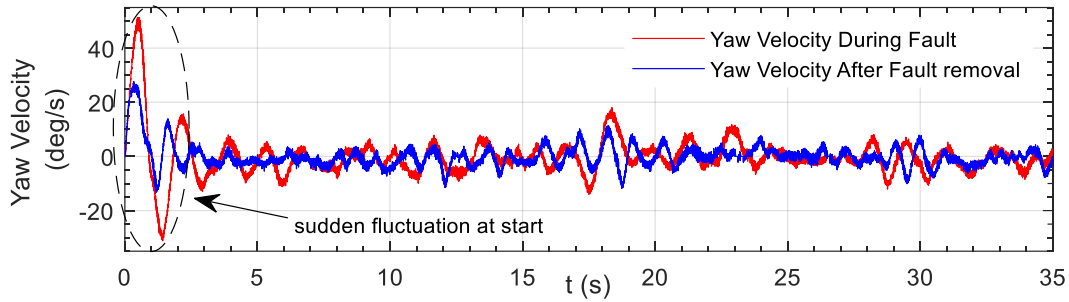
(a) Pitch angle



(b) Pitch velocity



(c) Yaw angle



(d) Yaw velocity

Figure 5.7. Pitch and yaw actuator response of helicopter during fault and after fault removal conditions in real-time

From Figure 5.7, it is observed that the pitch angle depicted huge initial oscillations ranging between 5 degrees to  $-12$  degrees during the fault. As the fault is removed, the initial oscillations are minimized and range between  $-1$  degree to  $-6$  degree for pitch angle. Similarly, the yaw angle depicted huge initial oscillations ranging from 28 degrees to 12 degrees during the fault, and after the fault is removed, these oscillations are ranged between 15 degrees to 13 degrees. Further, due to low pitch velocity, the necessary thrust is unable to generate by the pitch motor and fails to move in the vertical direction to follow a predefined trajectory. The coupling phenomena demands to minimize the yaw velocity, but motor fault raises it to the max 48 degree/sec. This is improved by implementing the NiF controller which removes the fault from the system and generates the necessary control action.

## 5.2 FAULT CLASSIFICATION-BASED CONTROL FOR BALL BALANCER SYSTEMS

In this section, a fault classification and control algorithm has been explored for reliable operation of ball balancer systems. The developed approach identifies the variation in operating characteristics of the ball balancer for different faults and operating conditions in time and frequency domain and achieves the control operation with the help of a wavelet fuzzy controller. Conventionally, fault tolerant control schemes such as adaptive linear parameter varying [296], network-based filter [297] and parametric linear quadratic regulator [298], etc have been implemented on ball balancer applications. The main drawback of these techniques is their inability to detect and adapt for multiple faults and operating conditions. To overcome this, the need to classify the operation of the system is very much required. Hence fault classification techniques come into the scenario for developing an efficient fault classification-based control.

### 5.2.1 Problem formulation in ball balancer system

Ball balancer systems are subjected to disturbances as they hold movement of freedom for their rigid body motion with two actuator control inputs. This presents some nonlinear dynamics to the ball balancer system, which includes translational and rotational movements in X, Y and Z-directions. The translational movements are expressed as surge, sway, and heave in X, Y, Z direction respectively and roll, pitch, and yaw respectively for rotational dynamics [390]. The schematic diagram for motion of ball balancer in X, Y, Z directions is shown in Figure 5.8.

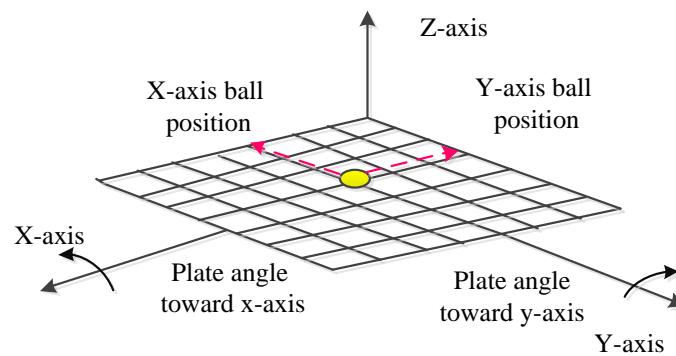


Figure 5.8. Three axial dynamics of ball balancer system representing the translational and rotational movements in XYZ plane

Generally, the main objective of a controller in the ball balancer system is to perform set point tracking under various disturbances such as variation in actuator operation, overloading, and other disturbances. Due to these conditions, there is huge possibility for faults in the ball balancer systems which further effect the dynamics of the system. In addition, the faults due to external and internal scenarios create a large amount of energy loss for the system. The main reason of internal faults are sensor failure [391], actuator failure [298] and for the external faults, the environmental conditions [392] are responsible. The sensor failure distorts the data used in ball position control by affecting the control decision which initiates after timely identification of environmental effects and neutralize them simultaneously. Apart from the above, while dealing with balancing control, robustness play important role under faulty conditions by keeping the dynamics of the system constant [297]. Hence, for achieving steady operation with the ball balancer system, the identification of faults is necessary. Once a fault is identified, the controller must be capable of operating the system by riding through the fault or indicating the action of preliminary measures to avoid further damage. To achieve this, two steps are adapted while developing the fault classifier prior to the reconfigurable control development. In step 1, the wavelet transform discussed in section 5.1.1 is used for extracting

features of different fault conditions in ball balancer systems. Further, in step 2, the support vector machine is adapted as discussed further to develop the fault classification approach.

### 5.2.2 Support Vector Machine Classifier

The main idea of support vector machine (SVM) classifier is to map the input vectors (features) into a high dimensional feature space through predefined non-linear mapping. In case of classification in the created high dimensional space, a linear decision surface is constructed with special properties that ensure a high generalization ability of SVM [393].

In general, for a linearly separable input data, the SVM algorithm searches for the optimal hyperplane in the unchanged feature space of the input data. However, in any case if the input data is linearly non-separable, the SVM maps the input data in to the higher dimensional feature space using a kernel function. The used kernel function must satisfy Mercer's theorem [394] and correspond to an inner product in the constructed high dimensional feature space. One of the advantages of SVM is its universality researcher's assumptions and problem domain. Another advantage of SVM algorithm is its effectiveness in the high dimensional feature spaces.

Generally, SVM is widely used in many problem domains and often outperforms other machine learning algorithms. The core of SVM lies within the different kernels, which are used for separating support vectors from the rest of the training data. The time complexity of SVM is  $O(\max(n, d) * \min(n, d)^2)$  according to [395] where  $n$  is number of samples and  $d$  is number of features.

#### 5.2.2.1 Data Preparation

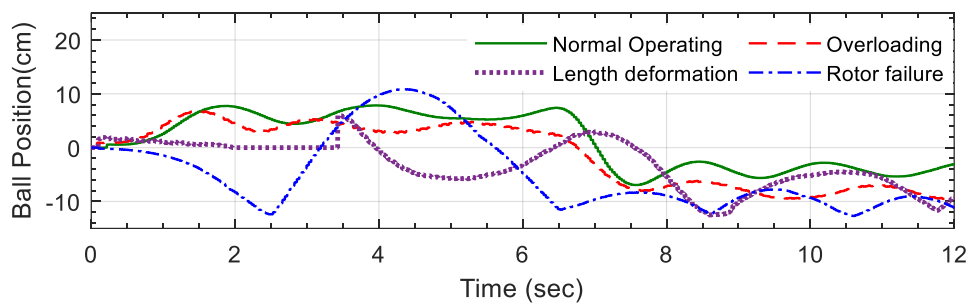
The ball balancer system discussed in chapter 2 has been considered for achieving fault classification and control analysis. As per the system specifications, the plate dimensions are  $27.5\text{cm}$  in length and  $27.5\text{cm}$  in width. These dimensions restrict the freedom of movement for the ball within a length of  $13.75\text{cm}$  considering the ball is to be positioned at the centre of the plate. But this length makes the ball to move to the corners of the plate, and the target of the research is to stabilize the ball on the centre. So, a range of  $0\text{-}6.875\text{ cm}$  has been taken to restrict the ball movement in middle of plate and reference of  $3\text{ cm}$  has been chosen to keep the ball closer to plate centre. To test the operation of the setup, a  $0.08\text{ Hz}$  square wave of amplitude  $3$  is considered as input for the system [213]. Further, from the internal and external factors that affect the ball balancer operation, the experimental analysis is carried out by



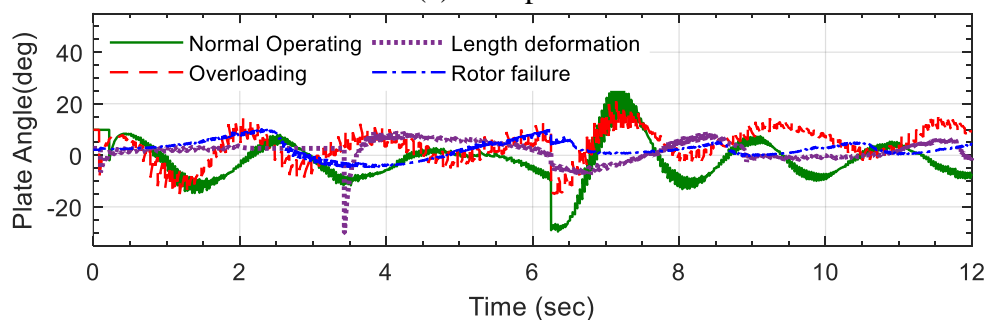
simulating 3 critical faults. The corresponding results are depicted in Figure 5.9 and their description and effects are detailed in Table 5.4.

Table 5.4. Sample data analysed in 2DoF ball balancer for developing the fault classifier

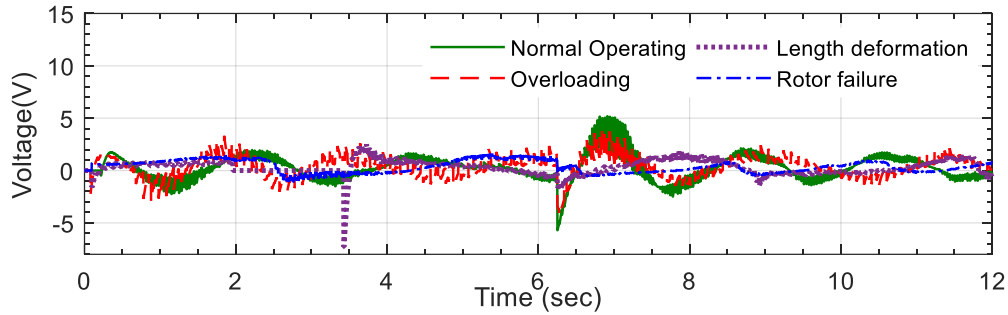
Component	Corresponding Ball balancer parameter	Fault Type	Description
Normal Operation (Ball Position-NO-D_X, Plate Angle- NO-T_X, Voltage- NO-V_X)	---	No Fault	The system is operating under normal test condition
Motor failure (Ball Position-RF-D_X, Plate Angle- RF-T_X, Voltage- RF-V_X)	Actuator failure for translation motion of ball on plate	Gains of servo units are increased	Sudden change in motor inertia
Plate Deformation (Ball Position-PL-D_X, Plate Angle- PL-T_X, Voltage- PL-V_X)	Ball balancer plate mechanical deformation	Plate distortion or twist	Reduced or deformations in plate length
Plate Overloading (Ball Position-OL-D_X, Plate Angle- OL-T_X, Voltage- OL-V_X)	Plate overloading	Loading with balls of different mass	Increased load and failure of motor operation



(a) Ball position



(b) Plate angle



(c) Operating voltage

Figure 5.9. Different faults and operating states of ball balancer system

From Figure 5.9, it can be observed that, the ball position follows the trajectory in normal operation and very unbalanced during remaining conditions (Figure 5.9. (a)). During overloading conditions, the plate angle will move slowly because of heavy ball, which takes more time to stabilize. Due to which ball movement will be unpredictable and it will show huge variation of 4 degree at 4sec and -29 degree at 13.5 sec respectively from Figure 5.9. (b) and cause in unstable operation. During length deformation fault condition, the plate will fluctuate due to external disturbance on plate and will subjected to rapid variation in ball position from Figure 5.9(a). During rotor failure the voltage will drop to extremely low value lying between 1.8volts to -1.2 volts from Figure 5.9(c) which results in to huge oscillations between 2 to 12 seconds from Figure 5.9(a).

#### 5.2.2.2 Feature Extraction and Classification:

In data analysis and data mining, the original data always requires some pre-processing before making further assumptions and applying machine learning algorithms. Initially, the data corresponding to various faults and operating conditions of the ball balancer system are extracted by performing multiple experiments on the laboratory setup of 2DoF ball balancer system. This data plays a major role in developing the fault classification algorithm.

In the next step, the extracted data is subjected to pre-processing for eliminating the noise, and extracting the features of the signal by adapting discrete wavelet transform as discussed in Section 5.2.2.1. The high pass and low pass filters of discrete wavelet transform perform decomposition and reconstruction of the data for eliminating the noise, and provide the energy feature of the signal. Apart from the energy feature, the reconstructed signals contain different simple and advanced statistical features which can help in developing the classification algorithm. In this research different features like entropy, power, peaks, harmonics, noise ratio, skewness and kurtosis of the signals are extracted as discussed in [377].

The extracted features are tabulated to form a feature matrix of  $11385 \times 8$  dimensionality, with 11385 features of all the samples of different classes (faults and operating conditions) and 8 is the types of features extracted from the plant operating and fault data. For developing a classifier with the SVM, the features of different classes are categorized as predictors and the classes are categorized as responses. Before subjecting the feature data for training with SVM, a 10-fold cross validation is specified to estimate the testing efficiency of the developed model with new data. The cross validation is performed by, 1) resampling or shuffling a limited data sample, 2) splitting the data into 10 folds, 3) testing each fold with the training set, and 4) summarize the testing or estimation efficiency using evaluated data of each fold.

### 5.2.2.3 Classifier Results

Initially the tabulated feature data is scattered in to a feature space where all the samples corresponding to each class can be observed between any two features as shown in Figure 5.10(a). Since the feature data obtained from the experiment is nonlinear in nature and SVM only deals with a linear data, a gaussian kernel is chosen to map the non-linear data in to a high dimensional feature space. The gaussian kernel has no prior knowledge of the data being handled and can handle multiple classes which is an advantage for developing the classifier with non-linear data. As the data is mapped into high dimensional feature space and trained with the SVM, the feature plane offers an insight of truly predicted samples and falsely predicted sampled between two features as shown in Figure 5.10(b). From the figure it can be identified that, the cross marked (x) samples are falsely classified and the remaining are truly classified samples.

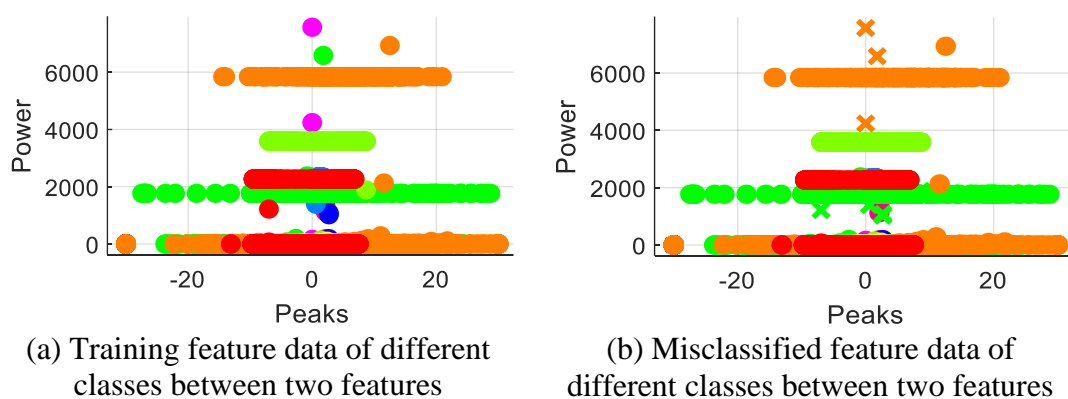


Figure 5.10. Feature space between peak and power features of ball balancer operating conditions

Considering the misclassified features while training the features of different classes, a confusion matrix is plotted to identify the truly and falsely classified data. This confusion matrix allots the samples under four different categories of true positive (TP), false positive

(FP), true negative (TN) and false negative (FN). Further, the confusion matrix also aids in estimating the accuracy of the developed model as follows:

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (5.31)$$

The confusion matrix for the trained classifier is shown in Figure 5.11. The truly and falsely classified samples are arranged against their respective classes in an axis of true class to the positive predicted class. The samples along the green diagonal depict the truly classified samples (in %) and the misclassified samples are arranged above and below the diagonal where the samples above the diagonal correspond to false positive and the samples below the diagonal are false negative.

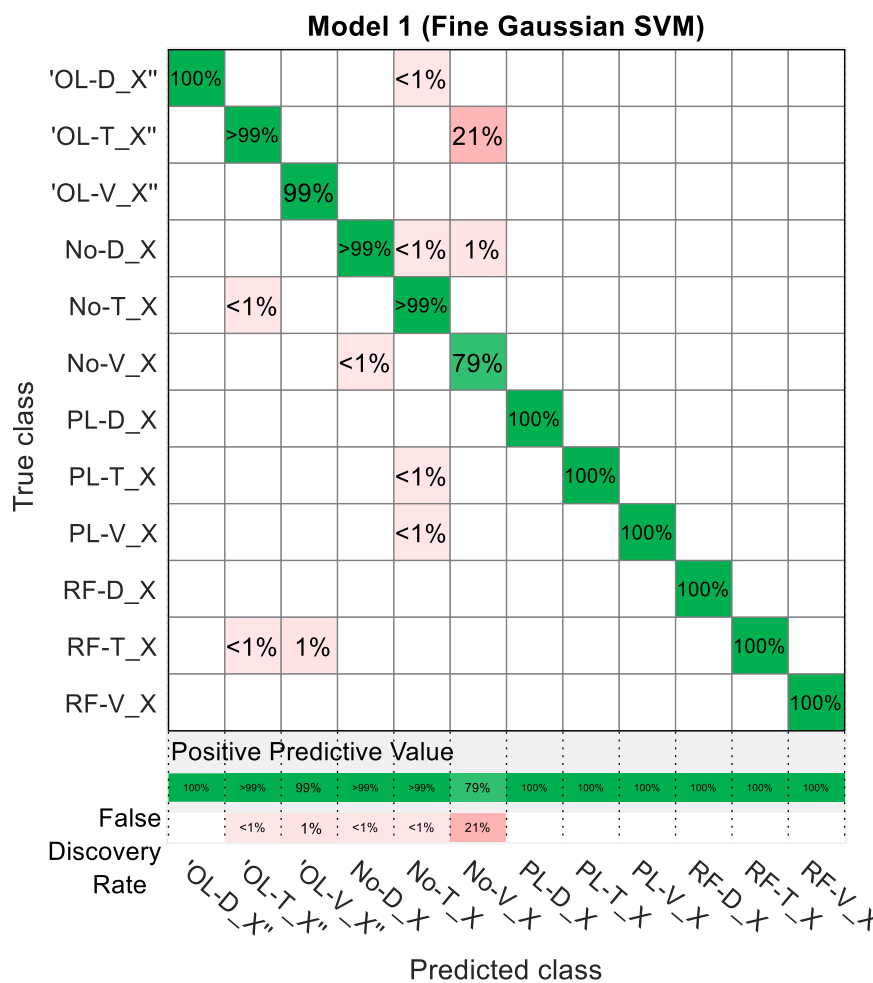


Figure 5.11. Confusion plot for prediction rate of the ball balancer fault classifier. From the feature matrix of size 11385 \* 8, 98.6% samples are truly classified for all the classified and the 1.4% samples are falsely classified. This brings the training accuracy of the classifier to 98.6%. Here, the classifier accuracy only reports the correct responses which is good in case of balanced datasets. But while dealing with real world signals i.e., imbalanced data, the classification accuracy needs to be supported with other metrics to justify the

classification process. Hence the area under the curve (AUC) receiver operating characteristics (ROC) are plotted for the trained model.

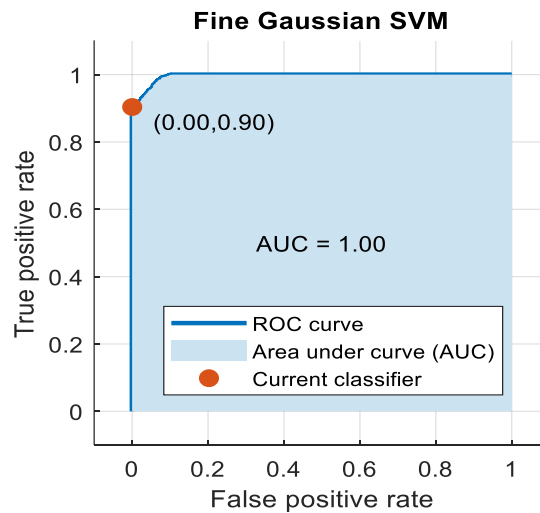


Figure 5.12. Receiver operating characteristics – Area under the curve for ball balancer fault classifier

Figure 5.12. depicts the ROC curve which is a graphical representation of relationship between recall and (1 – specificity), indicated as false positive rates on y-axes and true positive rates on x-axes, and illustrate the performance of the classifier. The maximum performance is achieved at point [0, 1]. ROC-AUC (Area Under the Curve) is a numerical estimation of classifier’s quality in case of using ROC curve. Here the ROC AUC is equal to 0.9 which depicts that the accuracy of the classifier in developing the classification algorithm is very high. Table 5.5 compares the efficiency of the developed classification algorithm with conventional classification algorithms.

Table 5.5. Comparison of classification parameters for ball balancer fault classifiers

Model Type	Parameters	
<i>Preset</i>	<i>Support Vector Machine</i>	<i>K-Nearest Neighbour</i>
Kernel Function	Fine Gaussian	Fine Gaussian
Kernel Scale	Automatic	Automatic
Accuracy	98.6%	81.4%
Prediction Speed	~18000 observations/sec	~8909 observations/sec
Training Time	9.909 sec	18.62 sec
Testing Time	20msecs	3secs

From the results, it is observed that the classifier efficiently trains the feature data to develop the fault classifier for the ball balancer system.

#### 5.2.2.4 Fault Identification:

For a ball balancer system which is operating under various factors and conditions, there are many potential faults that can impact its performance. In most of the cases, the possibility for

multiple faults to affect the operation of the system needs to be considered. For identifying or validating the presence of a classified fault in the system, a set of residual generators are computed. These residual generators act as a function of actuator and sensor data which is zero for an ideal or fault-free condition. Since, the performance of fault detection or classification is complicated by different measurement noises and model uncertainties. Any change in the output of residual generator needs to be evaluated initially either by thresholding or with a test quantity. In order to monitor different parts of the system and generate various fault subsets, multiple residual generators can be considered. In case of untrained or unknown faults, it is necessary to identify the likelihood that an unknown fault has occurred. By adapting this fault identification process, the ability of the classifier, for identifying the operating state of the system during unknown, untrained or multiple faults occurring simultaneously or in a sequence can be improved.

### 5.2.3 Fault based control process

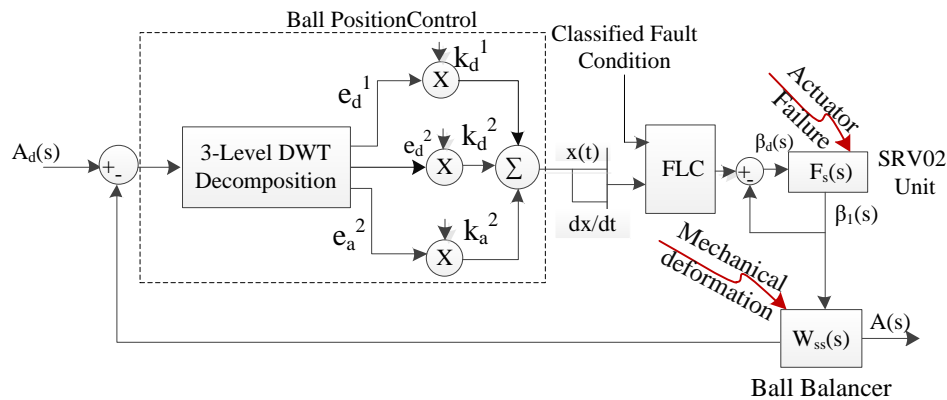


Figure 5.13. Discrete wavelet transform based reconfigurable fuzzy controller for ball balancer system

In order to accommodate the system control for better operation during degradation and fault conditions, a fault tolerant control method is developed to maintain system stability and achieve desired performance. To achieve this, a wavelet transform based fuzzy logic controller is developed for controlling the system by adhering to the condition of the system. The schematic of the wavelet fuzzy based position controller is shown in Figure 5.13.

The difference between the desired position and real-time position of the ball in the ball balancer model defines the input for the controller. The input of the system is a square wave with amplitude 3, and the input signal frequency is 0.08Hz. The monitoring for the inner loop is done by controlling the SRV02 position by providing a servo proportional gain ( $K_{PG,Servo-14}$ ), and for outer loop control, wavelet based fuzzy controller has been designed.

#### 5.2.4 Real-Time Results

To assess the performance of the proposed controller with the developed fault classification algorithm, three different sets of unclassified fault data related to motor fault, plate deformation and overloading are tested with the classifier individually. The classification efficiency of the developed classifier, when tested with different faults are depicted in Table 5.6.

Table 5.6. Testing efficiency of the classifier for different faults in ball balancer system

<b>Fault Type</b>	<b>Classifier Training Efficiency</b>	<b>Testing Efficiency</b>
Motor Fault	98.6%	99%
Plate Deformation	98.6%	99%
Over Loading	98.6%	82%

From the Table 5.6, it can be noted that, the classifier efficiently classifies the motor fault and plate deformation fault due to the variation of output signals from the normal operation. Whereas, while testing with the over loading fault, it can be observed that the classifier efficiency reduces as the output of the system during overloading has some similarities with the normal operation as the system tries to balance the ball.

Further, the action of motor fault is adapted for testing the operation of the control action. Initially, the system is operated to replicate the behaviour of motor fault by varying the actuator gains of the servo units. The plate angle, ball position, and system operating voltage are measured as shown in Figure 5.14 (During fault waveforms) and their corresponding features are extracted to test with the trained classifier. The output from the classifier identifies the current operating mode of the system. Once the fault mode is identified by the classifier, the output corresponding to the operating mode is sent to the controller to select the corresponding response of the controller from the pre-computed rule base. Further, the precomputed rules of the fuzzy controller act according to the classified fault to enhance the performance of the system by adjusting the weights of rules to reinstate the actuator gain values.

Results from the execution of these controllers to Simulink and real-time are checked dependent on the momentary investigation. Table 5.7 shows, the output of DWT-FLC, which is minimum in comparison with FLC on real-time results. Table 5.8 shows the root mean square value of DWT-FLC and FLC in terms of position and plate angle parameter using in real-time. Figure. 5.14(a), 5.14(b) and 5.14(c) show the plate angle, ball position and servo motor voltage. Angle variation is between 27 degrees and -14 degrees, made the movement of the ball stable and smooth for the DWT-FLC controller. Subsequently, by observing the real-time result of ball position from Figure 5.14(b), DWT-FLC is closer to reference with fewer oscillations

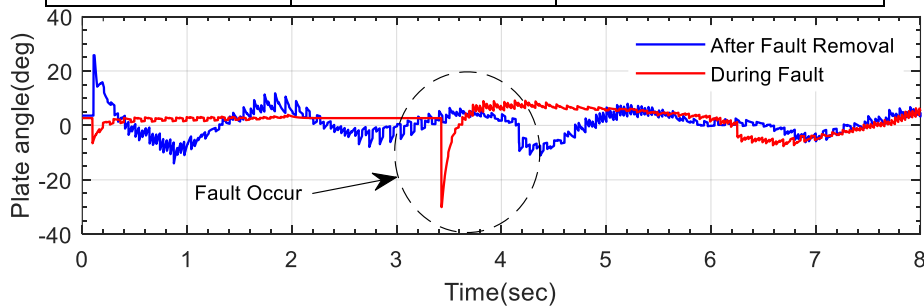
which make the system more stable than FLC as it settles for that trajectory which is away from the reference. However, DWT-FLC settled for the trajectory that almost equal to the desired one with excellent performance. Fuzzy logic controller provides a way of handling imprecision and nonlinearity while dealing with the position control error for the ball balancer system. Similarly, the DWT-FLC compares the position error with predefined wavelet amplitude and then reconstructs the signal to process it for the logical and analytical controlling.

Table 5.7. Time response analysis for reconfigurable control of ball balancer system

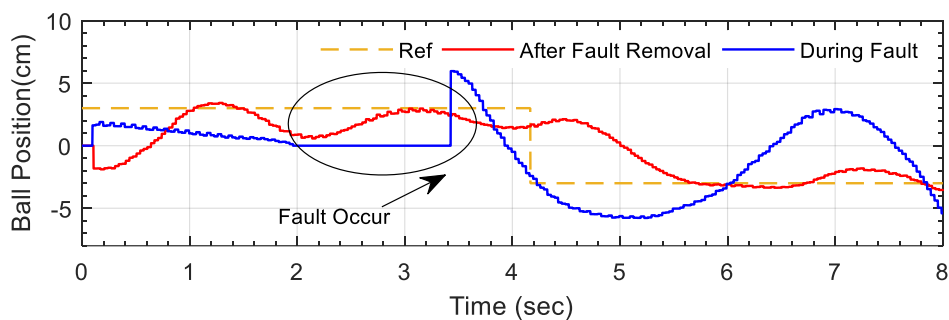
Controllers	Peak Time ( $t_p$ ) (sec)	Settling Time ( $t_s$ ) (sec)	Peak Overshoot ( $M_p$ ) (%)	Steady state error ( $e_{ss}$ ) (cm)
FLC	12.5sec	13.1sec	23.4%	4.1cm
DWT-FLC	3.07sec	4.07sec	20.9%	1.55cm

Table 5.8. Root mean square error for reconfigurable control of ball balancer system

Controller	Root mean square error	
	Position (cm)	Angle (deg)
FLC	5.1894 cm	3.4049 deg
DWT-FLC	3.5680 cm	2.7517 deg

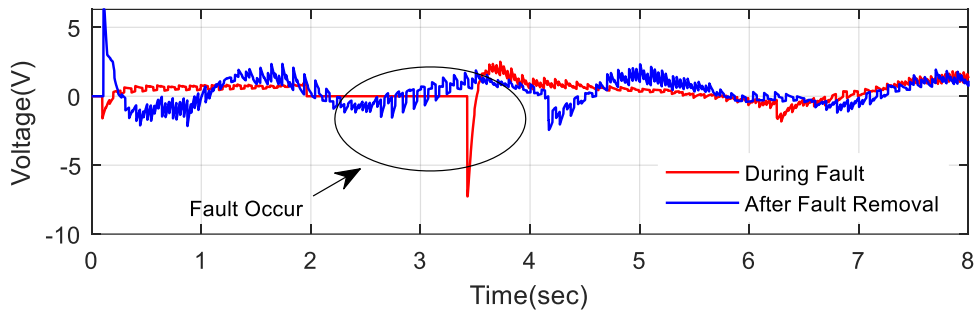


(a) Plate angle



(b) Ball position





(c) Operating voltage

Figure 5.14. Response of ball balancer during fault and after fault removal conditions in real-time

In order to evaluate the action of the controller during the fault, the position error of the ball movement on the plate with respect to the reference signal is measured. The position error of the system is calculated by finding the error between measured signal data and reference signal data, which is further subjected to piecewise function using curve fitting tool of the MATLAB. The steady state tracking error of the proposed control strategy is observed to be 1.55 cm. From the results, it is observed that the developed fault classification approach efficiently classifies the operating state of the system and controls it to achieve steady state operation. The ability of discrete wavelet transforms to identify the features for different faults and operating conditions and the supervised learning of the classifier proved to efficiently develop a fault classifier with better training and testing efficiencies. Further the fault identified is efficiently controlled with the help of wavelet fuzzy controller with less settling time, improve steady state error, position error and low angle variation.

### 5.3 CONCLUSION

This chapter develops a condition monitoring approach for helicopter and ball balancer system without the need for additional sensors. The developed approach localizes the faults to component level based on the classified fault and operating state of the plant. Further, the identified faults are validated based on residual generators. Besides, the developed intelligent fault tolerant control approach based on the identified fault and measured system characteristics, helps in improving the performance and safety of system. This achieves robust and stable operation even in the presence of faults.

## Chapter 6. REINFORCEMENT LEARNING WITH 2DOF SYSTEMS

The operation of robotic systems to perform complex tasks in dynamic environments has been a crucial area of control. Most of these control aspects are directly related to position control, path planning, trajectory tracking and balancing control of vehicles, etc. [305], [306]. Conventionally, many efforts have been made to achieve the control of autonomous robots, especially in the field of trajectory tracking, path planning [83], [93], [94], [105] and balancing control [307]–[309]. Recently, reinforcement learning (RL) based algorithm has been widely used to deal with these complicated control problems. In this chapter, a useful baseline is established for trajectory tracking and balancing control with most fundamental problems in optimal control theory. Further model free controllers are developed in linear and nonlinear environment using generalized learning algorithms, policy iteration (PI) and value iteration (VI). The PI and VI are implemented using various iterative and adaptive algorithms for solving the optimal control problem without a system model. To assess the performance of the developed model free controllers, real-time analysis is performed using benchmark control problems.

### 6.1 REINFORCEMENT LEARNING ALGORITHMS

The most general form of reinforcement learning (RL) is  $Q$ -learning, which uses action values ( $Q$  – values) to improve the behavior of learning iteratively [396]. These  $Q$  – values involve states and actions of a learning process  $Q(S, A)$  to estimate the performance of action  $A$  to state  $S$ . This estimation is computed iteratively by using a temporal difference update rule. Further, to learn a function according to the action, the RL agents have two types of policies, on policy and off policy. In on policy, the learning process is dependent on the current action derived from the current policy, whereas in off policy, the learning process is dependent on the action derived from a different policy. The  $Q$ -learning process refers to the off-policy technique by adapting a greedy approach to learn the  $Q$  – values. In addition to the  $Q$ -learning, the SARSA algorithm [397], which abbreviates for State Action Reward State Action, refers to on Policy technique to learn the  $Q$  – values.

## 6.2 LEARNING ALGORITHM-BASED LINEAR MODEL FREE CONTROL

The generalized expression for linear discrete time systems in time invariant state space is given by (6.1) as in [398] as:

$$\begin{cases} x_{k+1} = Ax_k + Bu_k \\ u_k = Kx_k + \epsilon_k \\ y_k = Cx_k + Du_k \end{cases} \quad (6.1)$$

where  $x_k \in \mathbb{R}^{n_x}$  corresponds to state at time  $k$ , with  $n_x$  corresponding to number of states,  $u_k \in \mathbb{R}^{n_u}$  corresponds to control input at time  $k$ , with  $n_u$  corresponding to number of inputs,  $y_k \in \mathbb{R}^{n_y}$  corresponds to controlled output at time  $k$ , with  $n_y$  corresponding to number of outputs,  $A \in \mathbb{R}^{n_x \times n_x}$  corresponds to state matrix,  $B \in \mathbb{R}^{n_x \times n_u}$  corresponds to input matrix,  $C \in \mathbb{R}^{n_y \times n_x}$  corresponds to output matrix,  $D \in \mathbb{R}^{n_y \times n_u}$  corresponds to feedthrough matrix, and  $\epsilon_k$  is the control noise.

Considering that the optimal gain of the system (6.1) is solved by a linear quadratic regulator (LQR), which minimizes the quadratic cost function. The expression for minimizing cost function is given by:

$$V_h(x_k) = \sum_{i_k=k}^{\infty} \gamma^{i_k-kr} (x_{i_k}, u_{i_k}), \quad (6.2)$$

where  $\gamma$  corresponds to the discounting factor,  $i$  indicates the number of step cost functions, and  $r(x_{i_k}, u_{i_k})$  corresponds to one step cost which is given as:

$$r(x_{i_k}, u_{i_k}) = x_{i_k}^T Q x_{i_k} + u_{i_k}^T R u_{i_k}, \quad (6.3)$$

where  $Q \in \mathbb{R}^{n \times n}$  and  $R \in \mathbb{R}^{m \times m}$  correspond to user defined state and control weighting matrix respectively.

The optimal gain ( $K^*$ ), minimizing the quadratic cost function with  $\gamma = 1$  is given in [398]–[400] as

$$K^* = (R + B^T X B)^{-1} B^T X A \quad (6.4)$$

where  $X$  corresponds to the solution of discrete time Algebra Riccati equation (ARE) which is given as:

$$X = A^T X A - A^T X B (R + B^T X B)^{-1} B^T X A + Q. \quad (6.5)$$

To solve (6.5), the knowledge regarding full dynamics of the system are necessary. Relentlessly, RL techniques proved to solve it using state, outputs and control actions without the need for system dynamics [399], [400].

### 6.2.1 Learning algorithm with state measurements

The model free solution of (6.6) is derived from [399] by expressing the quadratic cost function (6.2) as follows:

$$V_h(x_k) = r(x_k, h(x_k)) + \gamma V_h(x_{k+1}), V_h(0) = 0 \quad (6.6)$$

where  $V_h(x_k)$  is value function, and  $r(x_k, h(x_k))$  corresponds to one step cost as per (6.3) with  $i_k = k$ , and  $h(x_k) = u_k$ . This is known as Bellman equation. The optimal value and optimal policy are given as:

$$\begin{cases} V^*(x_k) = \min_{h(\cdot)} (r(x_k, h(x_k)) + \gamma V^*(x_{k+1})) \\ h^*(x_k) = \arg \min_{u_k} (r(x_k, h(x_k)) + \gamma V^*(x_{k+1})) \end{cases} \quad (6.7)$$

Using value function, the learning function as derived in [401] is given by:

$$Q^*(x_k, h(x_k)) = r(x_k, h(x_k)) + \gamma V^*(x_{k+1}). \quad (6.8)$$

From (6.7) and (6.8), (6.7) can be expressed as:

$$\begin{cases} V^*(x_k) = \min_{h(\cdot)} (Q^*(x_k, h(x_k))) \\ h^*(x_k) = \arg \min_{u_k} (Q^*(x_k, h(x_k))) \end{cases} \quad (6.9)$$

From the Bellman equation, the general learning equation ( $Q_s$ ) is derived as:

$$Q_s(x_k, h(x_k)) = V_h(x_k). \quad (6.10)$$

From (6.6) and (6.10):

$$Q_s(x_k, u_k) = r(x_k, u_k) + \gamma Q_s(x_{k+1}, h(x_{k+1})). \quad (6.11)$$

For an LQR controller, the learning equation is derived in [399], [400] as:

$$Q_s(x_k, u_k) = z_k^T S z_k = r(x_k, u_k) + \gamma z_{k+1}^T S z_{k+1}, \quad (6.12)$$

where  $z_k \in \mathbb{R}^{n_z}$ ,  $n_z = n_u + n_x$  is

$$z_k = \begin{bmatrix} x_k \\ h(x_k) \end{bmatrix}, \quad (6.13)$$

The symmetric positive definite quadratic kernel matrix  $S$  is given as:

$$S = \begin{bmatrix} A^T X A + Q & A^T X B \\ B^T X A & B^T X B + R \end{bmatrix} = \begin{bmatrix} S_{xx} & S_{xu} \\ S_{ux} & S_{uu} \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} & \cdots & S_{1l} \\ S_{21} & S_{22} & \cdots & S_{2l} \\ \vdots & \vdots & \ddots & \vdots \\ S_{l1} & S_{l2} & \cdots & S_{ll} \end{bmatrix} \quad (6.14)$$

where  $S \in \mathbb{R}^{n_z \times n_z}$ ,  $S_{xx} \in \mathbb{R}^{n_x \times n_x}$ ,  $S_{xu} \in S_{ux}^T \in \mathbb{R}^{n_x \times u}$ ,  $S_{uu} \in \mathbb{R}^{n_u \times n_u}$ , and  $s$  are the elements of  $S$ , which is learned without a system model and solution of ARE  $X$ . Further, the LQR learning equation is given as:

$$W^T \phi(z_k) = r(x_k, u_k) + \gamma W^T \phi(z_{k+1}), \quad (6.15)$$

where  $W \in \mathbb{R}^{(n_z(n_z+1)/2) \times 1}$  corresponds to the upper triangular terms of  $S$  matrix, and  $\phi(z_k) \in \mathbb{R}^{(n_z(n_z+1)/2) \times 1}$  corresponds to quadratic basis function. Here  $W \in \mathbb{R}^{(n_z(n_z+1)/2) \times 1}$  is a vector given by:

$$W = [s_{11}, 2s_{12}, \dots, 2s_{1n_z}, s_{22}, \dots, 2s_{2n_z}, s_{33}, \dots, 2s_{3n_z}, \dots, s_{n_z n_z}]^T, \quad (6.16)$$

and  $\phi(z_k) \in \mathbb{R}^{(n_z(n_z+1)/2) \times 1}$  in the case of LQR is a vector of quadratic terms of  $z_k$  given by:

$$\phi(z_k) = z_k \otimes z_k = [z_{k_1}^2, z_{k_1} z_{k_2}, \dots, z_{k_1} z_{k_{n_z}}, z_{k_2}^2, z_{k_2} z_{k_3}, \dots, z_{k_2} z_{k_{n_z}}, \dots, z_{k_{n_z}}^2]^T \quad (6.17)$$

where  $z_{k_{n_z}}$  is the  $n_z^{th}$  element of  $z_k$ .

If the optimal policy is minimized without any constraints, it yields:

$$\frac{\partial Q_s(x_k, u_k)}{\partial u_k} = 0 \quad (6.18)$$

Hence the complete solution for the above equation is presented by combining (6.11) to (6.13) as follows:

$$u_k = h(x_k) = -S_{uu}^{-1} S_{ux} x_k. \quad (6.19)$$

This defines the solution for optimal LQR gain without system model by using measured states and control inputs.

### 6.2.2 Learning algorithm with feedback measurements

In the case of partially observable linear systems, the learning algorithm is derived in [400]. Initially, the state  $x_k$  is given by:

$$x_k = [M_u \quad M_y] \bar{x}_k \quad (6.20)$$

Where  $M_u$  and  $M_y$  are the matrices with measured control and output values which are defined in the further equations, and  $\bar{x}_k$  is a vector dependent on previous controls and outputs which is given by:

$$\bar{x}_k = \begin{bmatrix} \bar{u}_k \\ \bar{y}_k \end{bmatrix}, \quad (6.21)$$

with  $\bar{u}_k$  corresponding to control vector and  $\bar{y}_k$  is the output vector given in (6.22) and (6.23) respectively:

$$\bar{u}_k = [u_{k-1} \quad u_{k-2} \quad \dots \quad u_{k-n}]^T \quad (6.22)$$

$$\bar{y}_k = [y_{k-1} \quad y_{k-2} \quad \dots \quad y_{k-n}]^T \quad (6.23)$$

Further, the observability index  $n$  is selected as  $n \leq n_x$ , with  $n_x$  corresponding to number of states as the upper bound. With observability index  $n$ , the matrices  $M_u$  and  $M_y$  are derived from [400] as

$$M_u = U_n - M_y T_n, \quad (6.24)$$

$$M_y = A^n (V_n^T V_n)^{-1} V_n^T \quad (6.25)$$

where  $U_n$  is the controllability matrix,  $T_n$  is the Toeplitz matrix, and  $V_n$  is the observability matrix which are given by:

$$U_n = [B \quad AB \quad \dots \quad A^{n-1}B] \quad (6.26)$$

$$T_n = \begin{bmatrix} 0 & CB & CAB & \dots & CA^{n-2}B \\ 0 & 0 & CB & \dots & CA^{n-3}B \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & 0 & CB \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (6.27)$$

$$V_n = [(CA^{n-1})^T \quad \dots \quad (CA)^T \quad C^T]^T \quad (6.28)$$

In addition to the upper bound of the observability index, the lower bound is given by  $n_k \leq n$ , in such a way that the  $\text{rank}(V_n) < n_x$  for  $n < n_k$ , and  $\text{rank}(V_n) = n$  for  $n \geq n_k$ . Therefore, the upper and lower bounds of the observability index are given by  $n_k \leq n \leq n_x$ .

The optimal policy for learning algorithm based on output feedback is obtained by denoting  $x_k$  in (6.9) with respect to (6.20) as follows:

$$u_k = K^* x_k = K^* [M_u \quad M_y] \bar{x}_k \quad (6.29)$$

Further, the state weighting parameter of LQR is given by:

$$Q = C^T Q_y C, \quad (6.30)$$

where  $Q_y$  corresponds to weighting parameter of output. Here, the learning algorithm uses the knowledge of state  $\bar{x}_k$  and output  $y_k$  instead of full state  $x_k$ , which gives two forms of learning algorithm as follows:

$$Q_s(\bar{x}_k, u_k) = \bar{z}_k^T T \bar{z}_k = r(y_k, u_k) + \gamma \bar{z}_{k+1}^T \bar{z}_{k+1}, \quad (6.31)$$

$$W^T \phi(\bar{z}_k) = r(y_k, u_k) + \gamma W^T \phi(\bar{z}_{k+1}) \quad (6.32)$$

where  $\bar{z}_k \in \mathbb{R}^{n_z}$ ,  $n_z = n(n_u + n_y) + n_u$ , is derived with a new state  $\bar{x}_k$  from (6.31) as follows:

$$\bar{z}_k = \begin{bmatrix} \bar{x}_k \\ u_k \end{bmatrix} \quad (6.33)$$

and  $r(y_k, u_k)$  is the new one step cost which is given by:

$$r(y_k, u_k) = y_k^T Q_y y_k + u_k^T R u_k \quad (6.34)$$

and the symmetrical matrix  $T$  is given by:

$$T = \begin{bmatrix} T_{\bar{u}\bar{u}} & T_{\bar{u}\bar{y}} & T_{\bar{u}u} \\ T_{\bar{y}\bar{u}} & T_{\bar{y}\bar{y}} & T_{\bar{y}u} \\ T_{u\bar{u}} & T_{u\bar{y}} & T_{uu} \end{bmatrix} \quad (6.35)$$

where  $T_{\bar{u}\bar{u}} \in \mathbb{R}^{n \times n}$ ,  $T_{\bar{u}\bar{y}} = T_{\bar{y}\bar{u}}^T \in \mathbb{R}^{n \times n}$ ,  $T_{\bar{u}u} = T_{u\bar{u}}^T \in \mathbb{R}^{n \times n_u}$ ,  $T_{\bar{y}\bar{y}} \in \mathbb{R}^{n \times n}$ ,  $T_{\bar{y}u} = T_{u\bar{y}}^T \in \mathbb{R}^{n \times n_y}$ , and  $T_{uu} \in \mathbb{R}^{n_u \times n_u}$  correspond to elements of matrix  $T$ .

Hence, the new policy is given by:

$$u_k = h(\bar{x}_k) = -(T_{uu})^{-1} [T_{u\bar{u}} \quad T_{u\bar{y}}] \bar{x}_k. \quad (6.36)$$

which results in an optimal control solution without considering the full state measurements.

### 6.2.3 LQR learning algorithm with temporal difference

In this section, the learning algorithm learns the optimal function and control policy, based on the temporal difference error of value and policy iterations. Generally, the policy iterations are used while stabilizing the control policy and the value iterations are used without stabilizing the control policy. The temporal difference error is expressed as:

$$e = Q_s(x_k, u_k) - r(x_k, u_k) - Q_s(x_{k+1}, u_{k+1}). \quad (6.37)$$

The learning function  $Q_s(x_k, u_k)$  and control policy  $u_k$  are iterated by policy and value iterations, such that,  $e$  converges close to *zero*. The main advantage with policy and value iteration is that, they work forward in time and are mostly suitable for real-time control. As the value iteration doesn't contribute for stable control at each iteration, it is generally adapted for safe offline operation of unstable systems. Further, the updating steps of policy and value iterations are given in algorithm 6.1 and 6.2. The first step of these algorithms deals with policy evaluation for updating the learning function. For linear conditions, this process is carried out with different techniques like least square, recursive least squares, and stochastic gradient descent. All these methods can be on policy method as in algorithm 6.1 or off policy methods as in algorithm 6.2.

---

**Algorithm 6.1:** On policy recursive algorithm [402]

---

**Step: 1** Initialize:  $\widehat{W}_0$  and  $\widehat{K}_0$

**Step: 2**  $k = 0, j = 0$

**Step: 3** Collecting data: Measure  $x_k, u_k$ , and  $x_{k+1}$  and calculate  $u_{k+1}$  using current policy.

**Step: 4** Value Update: update the kernel matrix  $\widehat{S}_{j+1}$  or weight  $\widehat{W}_{j+1}$

**Step: 5** if convergence happens?

Policy Update: update the current policy using the converged kernel matrix  $\widehat{S}_{j+1}$  and go to step 6.

else

update  $k = k + 1$

**Step: 6** if convergence happens?

The optimal control is found

else

update  $j = j + 1$  and  $k = k + 1$  and go to step 3.

---

Further, while performing the on-policy batch algorithm, a few modifications are made to the step 3 as follows:

**Step: 3** Collecting a batch of data:  $k_0 = k$ . Measure  $x_k, u_k$ , and  $x_{k+1}$  and calculate  $u_{k+1}$  using the current policy. Repeat for each  $k = k + 1$  until  $k = k_0 + M$ , where  $M$  is batch size.

---

**Algorithm 6.2:** Off policy algorithm [403], [404]

---

**Step: 1** Initialize:  $\widehat{W}_0$  and  $\widehat{K}_0$ . Collect data using a behavior policy.

**Step: 2**  $j = 0$

**Step: 3** Collecting a batch of data: Measure  $x_k, u_k$ , and  $x_{k+1}$  and calculate  $u_{k+1}$  using the current policy.

**Step: 4** Value Update: update the kernel matrix  $\widehat{S}_{j+1}$  or weight  $\widehat{W}_{j+1}$

**Step: 5** Policy Update: update the current policy using the converged kernel matrix  $\widehat{S}_{j+1}$ .

**Step: 6** if convergence happens?

The optimal control is found

else

update  $j = j + 1$  and go to step 3.

---

This algorithm is known as interleaved learning algorithm as the value update is performed only once unlike algorithm 6.1.

### 6.2.3.1 Policy Iterations:

The policy iteration equation for linear systems can be obtained either by following algorithm 6.1 or 6.2. To derive the PI, a stabilizing gain is needed as in [405], which can be obtained with some initial knowledge of the system. The algorithm is initialized at  $j = 0$  and the initial kernel matrix  $\widehat{S}_0$  is chosen randomly. The PI value update is given as:

$$Q_{j+1}(x_k, u_k) = r(x_k, u_k) + \gamma Q_{j+1}(x_{k+1}, h_j(x_{k+1})) \quad (6.38)$$



This results in temporal difference error and policy update which are given as (6.39) and (6.40) respectively:

$$e_j = Q_{j+1}(x_k, u_k) - r(x_k, u_k) - Q_{j+1}(x_{k+1}, u_{k+1}) \quad (6.39)$$

$$h_{j+1}(x_k) = \arg \min_{u_k} (Q_{j+1}(x_k, u_k)) \quad (6.40)$$

Further, by combining (6.39) with (6.12) or (6.15), the PI algorithm for LQR is derived. The value update using kernel matrix  $\hat{S}_{j+1}$  is given by:

$$z_k^T \hat{S}_{j+1} z_k - \gamma z_{k+1}^T \hat{S}_{j+1} z_{k+1} = r(x_k, u_k) \quad (6.41)$$

Similarly, the valued update using weighting matrix  $\hat{W}_{j+1}$  is given by:

$$\hat{W}_{j+1}^T \varphi_k = \mu_k, \quad (6.42)$$

where the data vector  $\mu_k$  is given by:

$$\mu_k = r(x_k, u_k) \quad (6.43)$$

and the regression vector  $\varphi_k$  is given by:

$$\varphi_k = \phi(z_k) - \gamma \phi(z_{k+1}) \quad (6.44)$$

Considering (6.41) or (6.42), the kernel or the weighing matrices are approximated while performing the value update. Further, the learning function is updated either by least square algorithm the recursive least square algorithm until the matrix converges. If the weighting matrix is used for policy update, it needs to be unpacked into kernel matrix from (6.14) and (6.16). The updated policy results in a new value update and the process is repeated until the convergence of weights is achieved i.e.,  $\|\hat{W}_{j+1} - \hat{W}_j\| \leq \varepsilon_j$ , where  $\varepsilon_j$  is a small constant.

For any case, only if the measured outputs are known, (6.41) or (6.42) are replaced with equations derived from (6.31) or (6.32) as follows:

$$\begin{cases} \bar{z}_k^T \hat{T}_{j+1} \bar{z}_k - \gamma \bar{z}_{k+1}^T \hat{T}_{j+1} \bar{z}_{k+1} = r(x_k, u_k) \\ \hat{W}_{j+1}^T \varphi_k = \mu_k, \\ \mu_k = r(x_k, u_k) \\ \varphi_k = \phi(\bar{z}_k) - \gamma \phi(\bar{z}_{k+1}) \end{cases} \quad (6.45)$$

Further, an exploration noise  $\epsilon_k$  is introduced with the control input to safeguard persistence of excitation condition, to get linearly independent data, and to achieve convergence of kernel matrix. This exploration can either be a sum of sine waves at different frequencies or a simple Gaussian noise. In addition, the bias from the solution can be removed by adjusting the

discounting factor  $\gamma$  around  $0 < \gamma < 1$ . For a value of  $\gamma < 1$ , results in a finite quadratic cost function, but doesn't guarantee the stability of closed loop system.

### 6.2.3.2 Value Iteration:

The value iteration algorithm follows the similar procedure of the policy iteration algorithms, except for the case that no stabilizing policy is required and the value update step is changed. The updated value in VI algorithm is given by:

$$Q_{j+1}(x_k, u_k) = r(x_k, u_k) + \gamma Q_j(x_{k+1}, h_j(x_{k+1})) \quad (6.46)$$

This results in temporal difference error:

$$e_j = Q_{j+1}(x_k, u_k) - r(x_k, u_k) - Q_j(x_{k+1}, u_{k+1}) \quad (6.47)$$

Therefore (6.31) is updated and the value update using kernel matrix is given by:

$$z_k^T \hat{S}_{j+1} z_k = r(x_k, u_k) + \gamma z_{k+1}^T \hat{S}_j z_{k+1} \quad (6.48)$$

and the data and regression vectors (6.43) and (6.44) in (6.42) are updated as:

$$\varphi_k = \phi(z_k) \quad (6.49)$$

$$\mu_k = r(x_k, u_k) + \gamma \hat{W}_j^T \phi(z_{k+1}), \quad (6.50)$$

where the kernel matrix  $\hat{S}_j$  and weighting matrix  $\hat{W}_j$  are from previous iteration  $j$ .

For the case, where the full states are not known, (6.45) is updated as in [406]:

$$\begin{cases} \bar{z}_k^T \hat{T}_{j+1} \bar{z}_k = r(x_k, u_k) + \gamma \bar{z}_{k+1}^T \hat{T}_{j+1} \bar{z}_{k+1} \\ \hat{W}_{j+1}^T \varphi_k = \mu_k, \\ \mu_k = r(x_k, u_k) + \gamma \hat{W}_j^T \phi(\bar{z}_{k+1}) \\ \varphi_k = \phi(\bar{z}_k) \end{cases} \quad (6.51)$$

Further, to ensure signal exploration, an exploration noise  $\epsilon_k$  is introduced with the control input.

#### 6.2.3.2.1 Batch Least Square Weight value update

The batch least square algorithm is used for calculating the value update of policy and value iteration by updating the weight matrix as per (6.42). This fits the weight matrix in such a way that the temporal difference error becomes small. When VI is used, the weight matrix is calculated based on previous iterations. Further, the updated weight for a batch algorithm is modified as  $\hat{W}_{j+1}^T \Phi = Y$ , with  $\Phi$  and  $Y$  corresponding to regression and data matrix

respectively which are formed from regression and data vector  $\varphi_k$  and  $\mu_k$  respectively as [407], [408]:

$$\Phi \in \mathbb{R}^{(n_z(n_z+1)/2) \times M} = [\varphi_k, \varphi_{k+1}, \dots, \varphi_{k+M}] \quad (6.52)$$

$$Y \in \mathbb{R}^{(n_z(n_z+1)/2) \times 1} = [\mu_k, \mu_{k+1}, \dots, \mu_{k+M}]^T \quad (6.53)$$

where  $M \geq n_z(n_z + 1)/2$  corresponds to the batch size.

These matrices are generated by choosing the data and regression vectors from (6.43, 6.44), (6.49-6.50), and (6.45) or (6.51) depending on known number of full states, and algorithm used. When the full states are known, these vectors use  $x_k, u_k, x_{k+1}$  and  $u_{k+1}$  which are calculated from (6.19). For the case if only measured outputs are known, each data point of the vector  $\bar{x}_k$  and  $\bar{x}_{k+1}$  are obtained from (6.21-6.23) and  $u_{k+1}$  is obtained from (6.26).

The updated least squares solve the one step weight update as follows:

$$\widehat{W}_{j+1} = (\Phi\Phi^T)^{-1}\Phi Y. \quad (6.54)$$

The inverse operation is performed only if the exploration noise is added to the control input so that the  $\text{rank}(\Phi) = n_z(n_z + 1)/2$ . Finally, the elements of weighting matrix are unpacked in to an updated kernel matrix.

### 6.2.3.2.2 Batch least squares kernel matrix value update

The batch least square algorithm is used for calculating the value update of policy and value iteration by updating the kernel matrix without a basis function as per (6.41) or (6.48). When VI is used, the previously updated kernel matrix ( $\hat{S}_j$ ) is calculated based on previous iterations. Since this is a batch value update, the temporal difference error at step  $j$  in (6.39) and (6.47) are updated for PI (6.55, 6.56) and VI (6.57, 6.58) as:

$$e_{LS,j} = Z_k^T \hat{S}_{j+1} Z_k - \Psi_k - \gamma Z_{k+1}^T \hat{S}_{j+1} Z_{k+1} \quad (6.55)$$

$$\Psi_k = X_k^T Q X_k + U_k^T R U_k. \quad (6.56)$$

$$e_{LS,j} = Z_k^T \hat{S}_{j+1} Z_k - \Psi_k. \quad (6.57)$$

$$\Psi_k = X_k^T Q X_k + U_k^T R U_k + \gamma Z_{k+1}^T \hat{S}_j Z_{k+1} \quad (6.58)$$

At each time  $k$  in batch size  $M \geq n_z(n_z + 1)/2$ , the states and control are measured,  $z_k$  and  $z_{k+1}$  are obtained from (6.13) and  $u_{k+1}$  is obtained from (6.19) and the data is collected to matrices  $X_k \in \mathbb{R}^{n_x \times M}$ ,  $Z_k \in \mathbb{R}^{n_z \times M}$ ,  $Z_{k+1} \in \mathbb{R}^{n_z \times M}$ , and  $U_k \in \mathbb{R}^{n_u \times M}$  as follows:

$$X_k = [x_k, x_{k+1}, \dots, x_{k+M}] \quad (6.59)$$

$$U_k = [u_k, u_{k+1}, \dots, u_{k+M}] \quad (6.60)$$

$$Z_k = [z_k, z_{k+1}, \dots, z_{k+M}], Z_{k+1} = [z_{k+1}, z_{k+2}, \dots, z_{k+M+1}] \quad (6.61)$$

For the case where only the output measurements are available, the vectors  $\bar{x}_k$  and  $\bar{x}_{k+1}$  are calculated from (6.21),  $u_{k+1}$  is obtained from (6.36) and  $\bar{z}_k$  and  $\bar{z}_{k+1}$  are obtained from (6.33). Further, the equivalent elements in batch matrices (6.59-6.61) are replaced with vectors  $\bar{x}_k$  and  $\bar{z}_k$ , which fits the kernel matrix with least squares. The batch equation for VI of (6.47) is given by:

$$Z_k^T \hat{S}_{j+1} Z_k = \Psi_k, \quad (6.62)$$

where  $\Psi_k$  is obtained from (6.58). Further, the above equation is derived as:

$$\Phi \hat{S}_{j+1} = Y \quad (6.63)$$

$$\begin{cases} Y = \Psi_k Z_k^T (Z_k Z_k^T)^{-1} \\ \Phi = Z_k^T \end{cases} \quad (6.64)$$

For  $Y$  to exist, the matrix  $Z_k Z_k^T$  should be invertible. This is possible only for  $\text{rank}(Z_k) = n_z$ . This condition is achieved by adding the exploration noise to the control. As the kernel matrix is symmetric, the batch policy iteration of (6.41) is given as follows:

$$Z_k^T \hat{S}_{j+1} Z_k - \sqrt{\gamma} Z_{k+1}^T \hat{S}_{j+1} \sqrt{\gamma} Z_{k+1} = (Z_k^T - \sqrt{\gamma} Z_{k+1}^T) \hat{S}_{j+1} (Z_k^T + \sqrt{\gamma} Z_{k+1}^T) = \Psi_k \quad (6.65)$$

The least square matrices  $Y$  and  $\Phi$  are updated as:

$$\begin{cases} Y = \Psi_k (Z_k^T + \sqrt{\gamma} Z_{k+1}^T)^T \left( (Z_k^T + \sqrt{\gamma} Z_{k+1}^T) (Z_k^T + \sqrt{\gamma} Z_{k+1}^T)^T \right)^{-1} \\ \Phi = (Z_k^T - \sqrt{\gamma} Z_{k+1}^T) \end{cases} \quad (6.66)$$

Similar to condition of (6.64), for  $Y$  to exist, the matrix  $(Z_k^T + \sqrt{\gamma} Z_{k+1}^T) (Z_k^T + \sqrt{\gamma} Z_{k+1}^T)^T$  should be invertible. This is possible only for  $\text{rank}(Z_k^T + \sqrt{\gamma} Z_{k+1}^T) = n_z$ .

Further, the updated kernel matrix is obtained with least squares as follows:

$$\hat{S}_{j+1} = (\Phi^T \Phi)^{-1} \Phi^T Y, \quad (6.67)$$

where  $(\Phi^T \Phi)$  is invertible, if  $\text{rank}(\Phi) = n_z$ .

From the above conditions, it is observed that, while using VI, the  $\text{rank}(\Phi) = \text{rank}(Z_k^T) = \text{rank}(Z_k)$ . This satisfies all the rank conditions simultaneously. But in the case of PI, the rank conditions are not equal and should be separately satisfied.

In case of nonlinear least squares, the value update is calculated by forming an objective to find a kernel matrix that minimizes the mean square error.

$$\min_{\hat{S}_{j+1}} \left( \frac{1}{2} e_{LS,j}^T e_{LS,j} \right) \quad (6.68)$$

Equation (6.55) or (6.57) are used to solve unknown kernel matrix, where an initial guess  $\hat{S}_0$  is iterated until minimum mean square error is achieved.

### 6.2.3.2.3 Recursive Least square value update

The recursive least square algorithm is used for calculating the value update of policy and value iteration by iterating the new weighting matrix  $\hat{W}_{j+1,i}$  at each time step until it converges and the values of  $\hat{W}_{j+1,\infty}$  is updated. To perform this, the value of covariance matrix  $P_0$  is initially chosen as:

$$P_0 = \delta I \quad (6.69)$$

where  $\delta$  corresponds to a large scalar value as given in [398]. For initialization, index  $i = 0$  and  $\hat{W}_{j+1,0} = \hat{W}_j$ .

If the full states are known, the vectors  $x_k, u_k, x_{k+1}$  are measured and  $u_{k+1}$  is calculated from (6.29). Otherwise, the vectors  $\bar{x}_k$  and  $\bar{x}_{k+1}$  are obtained from (6.21) and  $u_{k+1}$  is obtained from (6.36). Similarly, the regression and data vectors are obtained from (6.43, 6.44), (6.49, 6.50), and (6.45) or (6.51) depending on the algorithm chosen.

For a step update of the weight  $\hat{W}_{j+1,i}$ , the update matrix  $L_{i+1}$  is obtained as follows:

$$L_{i+1} = \lambda^{-1} P_i \varphi_k (a^{-1} + \lambda^{-1} \varphi_k^T P_i \varphi_k)^{-1}, \quad (6.70)$$

where  $\lambda$  corresponds to discounting factor of recursive least squares and  $P_i$  corresponds to covariance matrix at  $i^{th}$  iteration. For a regular least square, the value of discounting factor  $\lambda = 1$  and  $a = 1$ , but in the case of exponentially weighted recursive least squares,  $0 < \lambda < 1$  and  $a = 1 - \lambda$ . The updated weight and covariance matrices are obtained from (6.71) and (6.72) respectively as follows:

$$\hat{W}_{j+1,i+1} = \hat{W}_{j+1,i} + L_{i+1} (\mu_k - \varphi_k^T \hat{W}_{j+1,i}) \quad (6.71)$$

$$P_{i+1} = \lambda^{-1} (I - L_{i+1} \varphi_k^T) P_i \quad (6.72)$$

For increment in time  $k + 1$  and iteration  $i + 1$ , the new measurements are obtained with the current policy. Further, the one step updates of (6.70-6.72) are repeated for new data, until the convergence is achieved, such that  $\|\hat{W}_{j+1,i+1} - \hat{W}_{j+1,i}\| \leq \varepsilon_i$ .

### 6.2.3.2.4 Stochastic Gradient Descent based value update

The stochastic gradient descent algorithm is used for calculating the value update of policy and value iteration by fitting the linear models with small batches of data. As time  $k$  increases, the weight  $\widehat{W}_{j+1,i}$  is trained till the convergence is achieved. The value update for one step is initialized with  $i = 0$  and  $\widehat{W}_0 = \widehat{W}_j$ .

If the full states are known, the vectors  $x_k, u_k, x_{k+1}$  are measured and  $u_{k+1}$  is calculated from (6.19). Otherwise, the vectors  $\bar{x}_k$  and  $\bar{x}_{k+1}$  are obtained from (6.21) and  $u_{k+1}$  is obtained from (6.36). Similarly, the regression and data vectors are obtained from (6.43, 6.44), (6.49, 6.50), and (6.45) or (6.51). The temporal difference error is obtained from (6.39) and (6.47) as follows:

$$e_{sgd,i} = \widehat{W}_{j+1,i}^T \Phi_k - \mu_k \quad (6.73)$$

The minimized mean square error is obtained from [409] as:

$$E_{sgd,i} = \frac{1}{2} e_{sgd,i}^T e_{sgd,i} \quad (6.74)$$

and its gradient  $\frac{\partial E_{sgd}}{\partial \widehat{W}_{j+1,i}}$  is obtained using a chain rule. For PI, this is given as:

$$\frac{\partial E_{sgd}}{\partial \widehat{W}_{j+1,i}} = \frac{\partial E_{sgd}}{\partial e_{sgd,i}} \frac{\partial e_{sgd,i}}{\partial \widehat{W}_{j+1,i}} = (\phi(z_k) - \gamma \phi(z_{k+1})) e_{sgd,i}^T \quad (6.75)$$

and for VI, it is given as:

$$\frac{\partial E_{sgd}}{\partial \widehat{W}_{j+1,i}} = \phi(z_k) e_{sgd,i}^T \quad (6.76)$$

Further, the one step weight updated in obtained as follows:

$$\widehat{W}_{j+1,i+1} = \widehat{W}_{j+1,i} - \alpha_{sgd} \frac{\partial E_{sgd}}{\partial \widehat{W}_{j+1,i}} \quad (6.77)$$

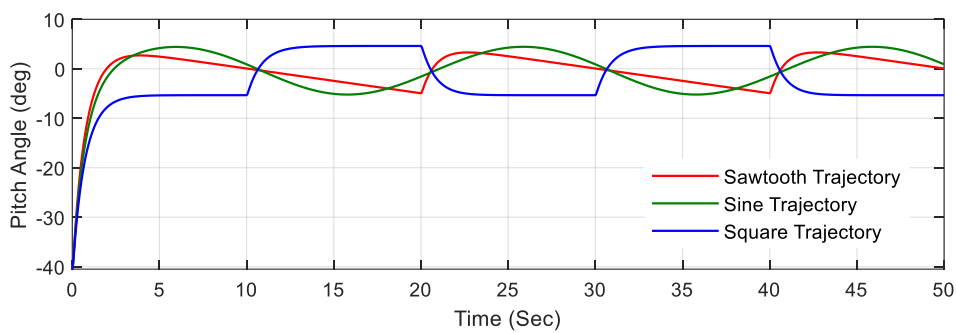
where  $\alpha_{sgd}$  corresponds to learning rate, which is constant.

For increment in time  $k + 1$  and iteration  $i + 1$ , the new measurements are obtained with the current policy. Further, the one step updates of (6.77) are repeated for new data, until the convergence is achieved, such that  $\|\widehat{W}_{j+1,i+1} - \widehat{W}_{j+1,i}\| \leq \varepsilon_i$ .

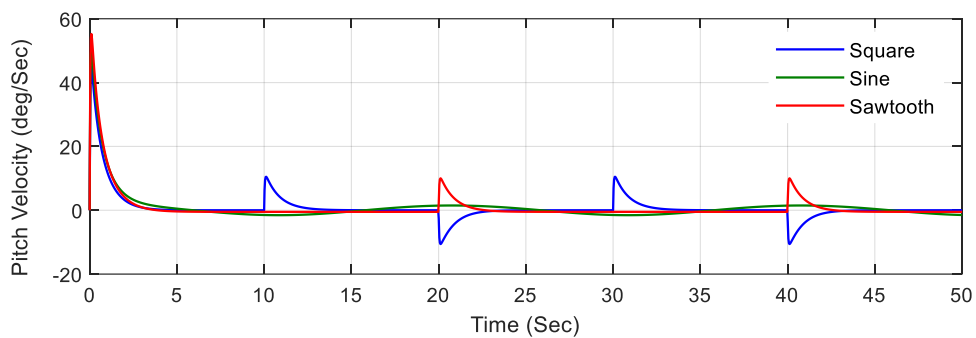
#### 6.2.4 Linear model free control for trajectory tracking of helicopter system

The problem of trajectory tracking in unmanned vehicles is realized by modelling the working phenomenon of the helicopter considering the pitch and yaw motion as shown in Chapter 2. Considering the pitch and yaw motor differential equations, the state, input, and output vectors

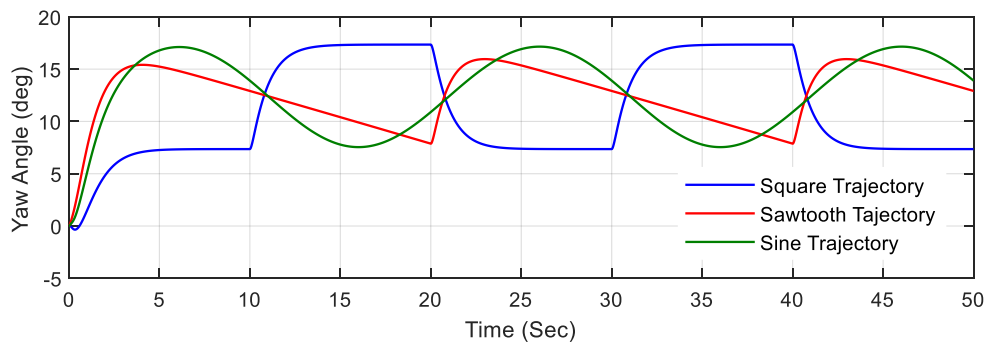
can represent the two-degree motion of helicopter as a continuous time linear system. Further, to achieve control development for the unmanned helicopter, the continuous time linear system is expressed as a linear discrete time system in time invariant state space. The developed model-free control is implemented for achieving trajectory tracking in unmanned helicopters using output feedback version of on-policy approach. The selected algorithm is implemented with the theoretical system model of unmanned helicopter for generating data required for learning the algorithm. The data in this experiment corresponds to various trajectories that define the states of the helicopter are obtained from the simulated model. The response of the helicopter pitch and yaw motors for various trajectories are shown in Figure 6.1.



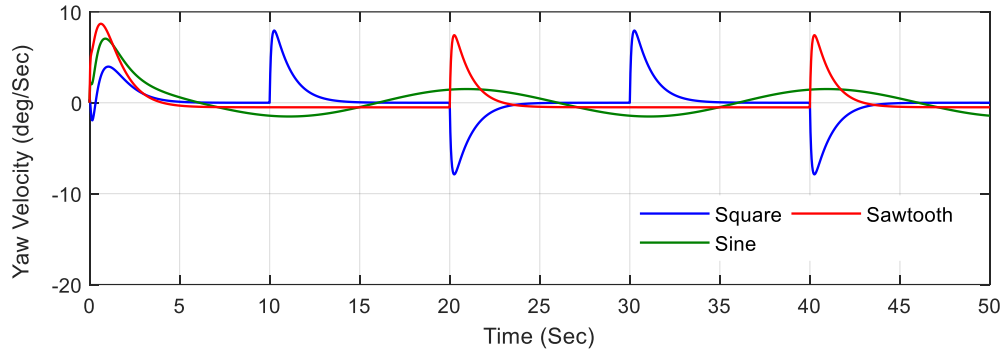
(a) Pitch angle



(b) Pitch velocity

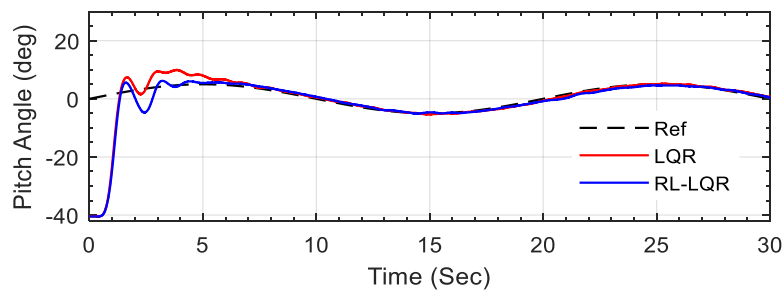


(c) Yaw angle



(d) Yaw velocity

Figure 6.1. Response of helicopter pitch and yaw motor operation for different trajectories. Initially, a reference point is developed to implement output feedback version of the algorithm. Further, the developed controller is modelled for a real-time system using Quanser two degree of freedom helicopter [57]. The learning process is associated with a linear control implemented in the system. The trajectory of unmanned helicopter has been defined by pitch angle and yaw angle movements. The main rotor lift is responsible for change in pitch, and pulling forces of tail rotor are responsible yaw angle movements. The objective is to decouple the two typical output channels such that the output responses follows a desired trajectory. From the dynamic model of the unmanned helicopter system it has been identified that, pitch and yaw axes can be controlled by appropriately selecting the voltages of the pitch and yaw motors respectively. Further the of error of both the pitch and yaw angles are computed by comparing them with the reference developed. The simulations are performed using both full state and output measurements of the model, but the real-time implementation of the developed algorithms is carried out only using output measurements as all the state measurements are not available for real-time system. The recursive least square based value update for PI and VI algorithms is implemented as on-policy method. The response of pitch and yaw motion of the unmanned helicopter with RL based LQR controller and LQR is analyzed. A sine reference trajectory with amplitude 5 is considered for assessing the pitch and yaw path tracking. The corresponding results are shown in Figure 6.2.



(a) Pitch angle



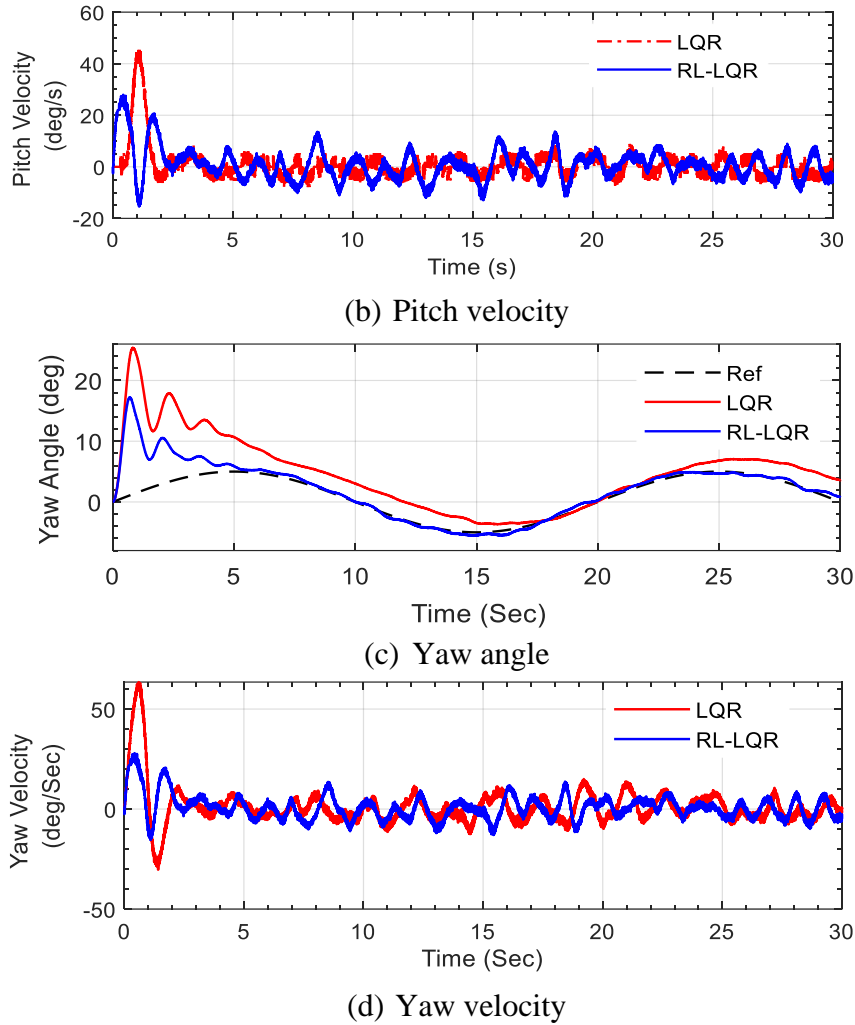


Figure 6.2. Pitch and yaw output with reinforcement learning-linear quadratic regulator control of helicopter in real-time for sine trajectory

The results depict the tracking response of helicopter with reinforcement learning based state feedback controller on simulation and real-time system with reference to the desired trajectory for pitch angle movement (Figure 6.2(a)), pitch velocity (Figure 6.2(b)), yaw angle movement (Figure 6.2(c)), and yaw velocity (Figure 6.2(d)). The control objective of the work is to track desired trajectory with minimum settling time. The pitch angle trajectory reaches 4.8 degrees at 1.2 secs and settles with the desired trajectory around 4 secs, and the yaw angle reaches 13.2 degrees at 0.8 secs and settles with the desired trajectory within 6 secs. The initial pitch velocity is reduced to 27.2 deg/sec at 0.5 secs when subjected to real-time environments which is sufficient enough for pitch rise, and the yaw velocity operates at 29 deg/ sec around 0.5 secs due to cross coupling effect.

Table 6.1. Root mean square error for reinforcement learning-linear quadratic regulator control of helicopter in real-time

Controller	Root mean square error	
	Pitch angle (deg)	yaw angle (deg)
LQR	7.4012 deg	5.4324 deg
RL-LQR	6.8636 deg	4.9981 deg

Table 6.2. Time response analysis for reinforcement learning-linear quadratic regulator control of helicopter in real-time

Controller	Pitch Control Response		Yaw Control Response	
	Settling time ( $t_s$ ) (sec)	Steady state error ( $e_{ss}$ ) (cm)	Settling time ( $t_s$ ) (sec)	Steady state error ( $e_{ss}$ ) (cm)
LQR	5.07 sec	3.19 cm	17.86 sec	5.15 cm
RL-LQR	4.67 sec	1.7 cm	5.02 sec	3.22 cm

The root mean square error for the controller action on real-time system is shown in Table 6.1. The settling time ( $t_s$ ), and steady-state error ( $e_{ss}$ ) are minimum in case of RL-LQR which are 4.67sec, and 1.7cm for pitch control, and 5.02sec, and 3.22cm respectively for yaw control as shown in Table 6.2. As the behavior policy of the real-time controller is calculated with the discrete linear model of the system, the learning algorithm converges quickly to settle the helicopter motion with respect to the defined trajectory.

### 6.3 LEARNING ALGORITHM-BASED NONLINEAR MODEL FREE CONTROL

The generalized expression for nonlinear affine discrete time system is given by:

$$\begin{cases} x_{k+1} = f(x_k) + g(x_k)u_k \\ y_k = c(x_k) \end{cases} \quad (6.78)$$

where  $f(x_k)$  corresponds to inner dynamics,  $g(x_k)$  corresponds to input dynamics, and  $c(x_k)$  corresponds to output dynamics of the system.

As discusses in 6.1.1, the optimal control for the system in (6.78) is achieved by minimizing the performance index of the system as given by:

$$V_h(x_k) = \sum_{k=0}^{\infty} r(x_k, u_k) \quad (6.79)$$

where  $r(x_k, u_k)$  is given in (6.3), with  $i_k = k$ . Further, with discounting factor  $\gamma = 1$ , the optimal value function in (6.7) can be solved analytically for (6.78) as:

$$u_k^* = -\frac{1}{2}R^{-1}g(x_k)^T \frac{\partial V^*(x_{k+1})}{\partial x_{k+1}} \quad (6.80)$$

From (6.8) and (6.9), the learning algorithm for nonlinear system in (6.7) and (6.80) is given as:

$$\begin{cases} Q^*(x_k, u_k) = r(x_k, u_k) + \gamma Q^*(x_{k+1}, u_{k+1}^*) \\ u_k^* = -\frac{\gamma}{2}R^{-1}g(x_k)^T \frac{\partial Q^*(x_{k+1}, u_{k+1}^*)}{\partial x_{k+1}} \end{cases} \quad (6.81)$$

To approximate the optimal value function, the neural network approach is adapted with the generalized actor critic architecture as shown in Figure 6.3 [410] for learning the optimal policy (actor) and optimal learning function (critic). Further, to achieve model free operation of the neural network, an additional network is added which provides the information of the system inner dynamics to the neural network. This additional network is commonly used with value function approximation and is known as identification network [411].

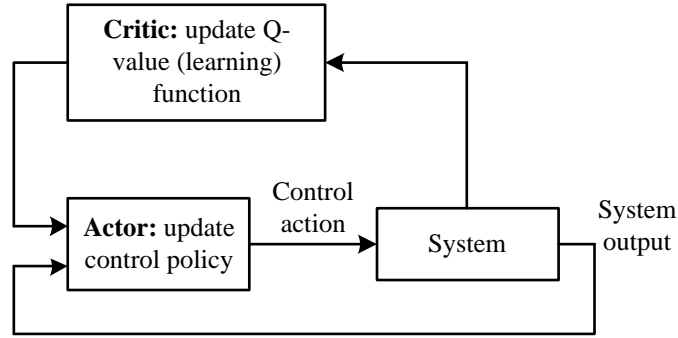


Figure 6.3. Actor-Critic architecture

### 6.3.1 Interleaved Learning with state measurements

In the interleaved learning process, a batch of neural network is fitted with the learning method to achieve optimal control for nonlinear affine discrete time system [404]. The equations in (6.81) are solved approximately by simultaneously updating three different networks. This process aims at minimizing the learning for the networks by minimizing the mean square error (MSE)  $E_{nw}(k)$  as follows:

$$E_{nw,j}(k) = \frac{1}{2}e_{nw,j}^T(k)e_{nw,j}(k) \quad (6.82)$$

where  $e_{nw}(k)$  corresponds to network estimation error, and  $e_{nw}^T(k)$  is the transpose of network estimation error at time instant  $k$  which is minimized using gradient descent method to update the weighted matrix  $\widehat{W}_{nw,j+1}$  of the network. The updated weight in an iteration  $j$  is given by:

$$\widehat{W}_{nw,j+1} = \widehat{W}_{nw,j} - \alpha_{nw} \frac{\partial E_{nw,j}(k)}{\partial \widehat{W}_{nw,j}(k)} \quad (6.83)$$

where  $\alpha_{nw}$  corresponds to the learning rate of the network and  $\frac{\partial E_{nw,j}(k)}{\partial \hat{W}_{nw,j}(k)}$  corresponds to gradient of mean square error (MSE).

The learned weights along with an activation function  $\sigma(z)$  are used to estimate the output of each network. In this research, the tanh function is chosen as the activation function for each network as follows:

$$[\sigma(z)]_{i_{nw}} = \tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (6.84)$$

where  $i_{nw}$  is one neuron, and its derivative is given by:

$$[\dot{\sigma}(z)]_{i_{nw}} = 1 - \tanh^2(z) \quad (6.85)$$

The interleaved learning process uses a training data set to learn the optimal policy. This training data set is collected by performing several experiments within the control region using stabilizing behavior policy with additional exploration noise. The experiments are repeated starting at the same initial time for  $n_{sets}$  times until  $N$  time steps. The learning process is divided into  $N$  sample sets such that each learning step forms its own batch size of  $n_{sets}$  at time step  $k$ . The algorithm for interleaved learning process is given as follows:

---

**Algorithm 6.3:** Interleaved learning procedure [404]

---

**Step: 1** Collect data: repeat the same learning step  $n_{set}$  times for  $N$  time steps. Divide the data into  $N$  sample sets so that  $k = 1, k = 2, \dots, k = N$  are formed from their own sample set.

**Step: 2** Set  $k = 0$  and initialize the network weights  $\hat{W}_m$ ,  $\hat{W}_a$ , and  $\hat{W}_c$

**Step: 3** Update the model network weight  $\hat{W}_m$  for each  $k$  until convergence.

**Step: 4**  $k = 0, j = 0$ .

**Step: 5** Update the actor and critic weights  $\hat{W}_a$ , and  $\hat{W}_c$

**Step: 6** if convergence happens?

update  $k = k + 1, j = 0$  and go to *step 5*

else

update  $j = j + 1$  and go to *step 5*.

**Step: 7** If  $k = N$ , stop.

---

As the interleaved learning process is a value iteration-based method, the initial weights need not be stabilizing. Hence, the weights of three networks are randomly initialized. From the algorithm it is observed that, the weights of identification networks are learned using all the data, hence, the time index is initialized only at  $k = 0$  without involving the iteration  $j$ . The output of identification network  $\hat{x}_{k+1}$  is given by [412]:

$$\hat{x}_{k+1} = \hat{W}_m^T \sigma \left( v_m^T \begin{bmatrix} x_k \\ u_k \end{bmatrix} \right) \quad (6.86)$$

Where  $v_m$  is the weight matrix of hidden layer of constant identification network. The identification network estimates the input dynamics  $\hat{g}(x_k)$  using (6.78) with gradient  $\hat{x}(k+1)$  in terms of  $u_k$  such that:

$$\frac{\partial \hat{x}_{k+1}}{\partial u_k} = \hat{g}(x_k) \quad (6.87)$$

The estimation error of identification network is given by:

$$e_m(k) = \hat{x}_{k+1} - x_{k+1} \quad (6.88)$$

The MSE of identification network is calculated from (6.82) with  $nw = m$ , and its gradient

$\frac{\partial E_m(k)}{\partial \hat{W}_m(k)}$  is given as:

$$\frac{\partial E_m(k)}{\partial \hat{W}_m(k)} = \sigma \left( v_x^T \begin{bmatrix} x_k \\ u_k \end{bmatrix} \right) e_m^T(k) \quad (6.89)$$

The weight of updated identification network with gradient descent update is given as:

$$\hat{W}_m(k+1) = \hat{W}_m(k) - \alpha_m \frac{\partial E_m(k)}{\partial \hat{W}_m(k)} \quad (6.90)$$

The equations (6.86-6.90) are repeated for each  $k$  till the estimation error  $\|e_x(k)\| \leq \varepsilon_m$ .

Once the identification network converges, the actor critic networks are updated as shown in algorithm 6.3. Initially, the time index is initialized at  $k = 0$ , and for each time  $k$ , the learning function is initialized at  $j = 0$  as  $Q_0(\cdot) = 0$ . The initial target policy  $u_0(x_k)$  is obtained as follows:

$$u_0(x_k) = \arg \min_{u_k} \left( x_k^T Q x_k + u_k^T R u_k + Q_0(\cdot) \right). \quad (6.91)$$

The learning function is iterated with increasing index  $j$  till the convergence is achieved for each time  $k$ . The temporal difference error is given as:

$$e_{c,j+1}(k) = \hat{Q}_{j+1}(x_k, \hat{u}_j(x_k)) - Q_{j+1}(x_k, \hat{u}_j(x_k)) = \hat{Q}_{j+1}(x_k, \hat{u}_j(x_k)) - r(x_k, \hat{u}_j(x_k)) - \hat{Q}_j(x_{(k+1),j}, \hat{u}_j(x_k)), \quad (6.92)$$

where the output of critic network is given by:

$$\hat{Q}_{j+1}(x_k, \hat{u}_j(x_k)) = \hat{W}_{c,j+1}^T(k) \sigma \left( v_c^T \begin{bmatrix} x_k \\ \hat{u}_j(x_k) \end{bmatrix} \right) \quad (6.93)$$

$$\hat{Q}_j(x_{(k+1),j}, \hat{u}_j(x_{(k+1),j})) = \hat{W}_{c,j}^T(k) \sigma \left( v_c^T \begin{bmatrix} x_{(k+1),j} \\ \hat{u}_j(x_{(k+1),j}) \end{bmatrix} \right) \quad (6.94)$$

$$x_{(k+1),j} = x_{k+1} - \hat{g}(x_k) (u_k - \hat{u}_j(x_k)) \quad (6.95)$$

Where  $v_c$  is the weight matrix of hidden layer of constant critic network, and  $\hat{u}_j(x_k)$  and  $\hat{u}_j(x_{(k+1),j})$  are the outputs of actor network. The MSE of critic network is calculated from (6.82) with  $nw = c$ , and its gradient  $\frac{\partial E_{c,j}(k)}{\partial \hat{W}_{c,j}(k)}$  is given as:

$$\frac{\partial E_{c,j}(k)}{\partial \hat{W}_{c,j}(k)} = \sigma \left( v_c^T \begin{bmatrix} x_k \\ \hat{u}_j(x_k) \end{bmatrix} \right) e_{c,j}^T(k) \quad (6.96)$$

Further, the outputs of the actor network that estimate the control policy are given as:

$$\hat{u}_j(x_k) = \hat{W}_{a,j}^T \sigma(v_a^T x_k) \quad (6.97)$$

$$\hat{u}_j(x_{(k+1),j}) = \hat{W}_{a,j}^T \sigma(v_a^T x_{(k+1),j}) \quad (6.98)$$

where  $v_a$  is the weight matrix of hidden layer of constant actor network. The estimation error for actor network is given by:

$$e_{a,j}(k) = \hat{u}_j(x_k) - u_j(x_k) \quad (6.99)$$

where the target policy  $u_j(x_k)$  is given by:

$$u_j(x_k) = -\frac{\gamma}{2} R^{-1} \hat{g}(x_k)^T \frac{\partial \hat{Q}_j(x_{(k+1),j}, \hat{u}_j(x_{(k+1),j}))}{\partial (x_{(k+1),j})} \quad (6.100)$$

The MSE of critic network is calculated from (6.82) with  $nw = a$ , and its gradient  $\frac{\partial E_{a,j}(k)}{\partial \hat{W}_{a,j}(k)}$  is given as:

$$\frac{\partial E_{a,j}(k)}{\partial \hat{W}_{a,j}(k)} = \sigma(v_a^T x_k) e_{a,j}^T(k) \quad (6.101)$$

Further, the process in (6.91-6.101) is repeated as discussed in algorithm 6.3 by increasing index  $j$  until  $\left\| \hat{Q}_j(x_k, \hat{u}_{j-1}(k)) - \hat{Q}_{j+1}(x_k, \hat{u}_j(k)) \right\| \leq \varepsilon_j$ . Once the convergence is achieved, the optimal control policy at time  $k$  is saved and the interleaved iterations are repeated from  $j = 0$  for each  $k$  until all the networks are converged.

### 6.3.2 Nonlinear model free control for ball balancer system

The problem of balancing control in robotic systems is realized by modelling a ball balancer system. To achieve nonlinear control for the ball balancer system, the neural network approach is adapted as shown in Figure 6.4.

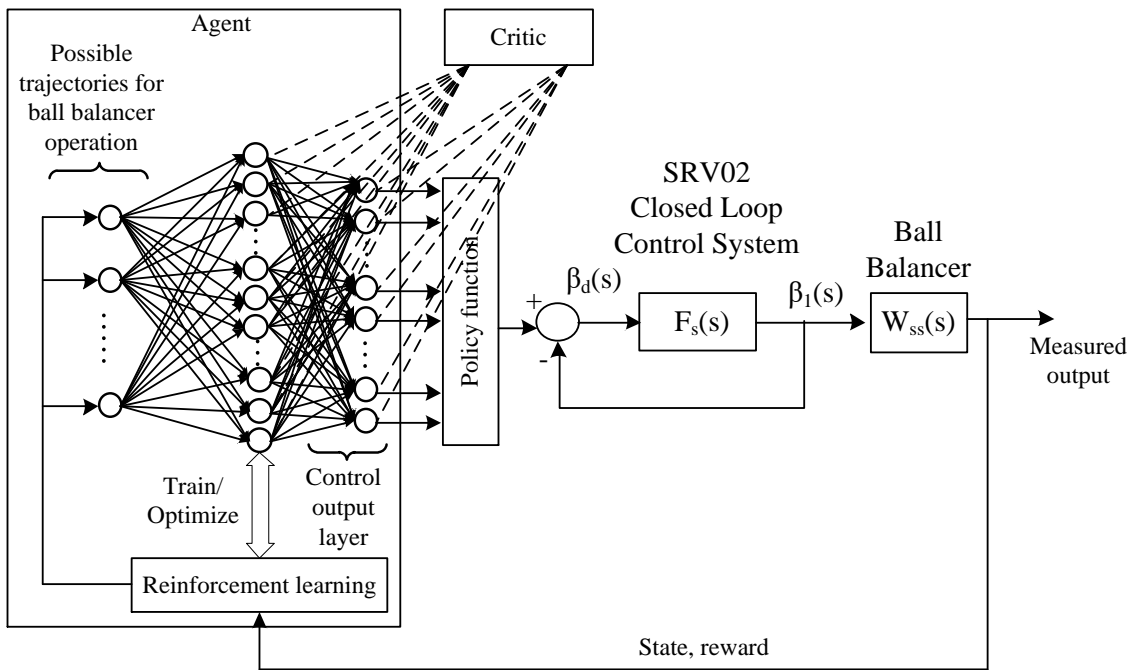
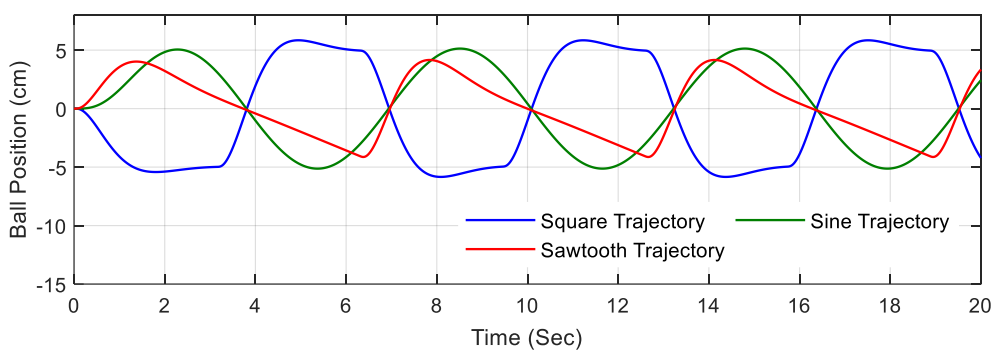
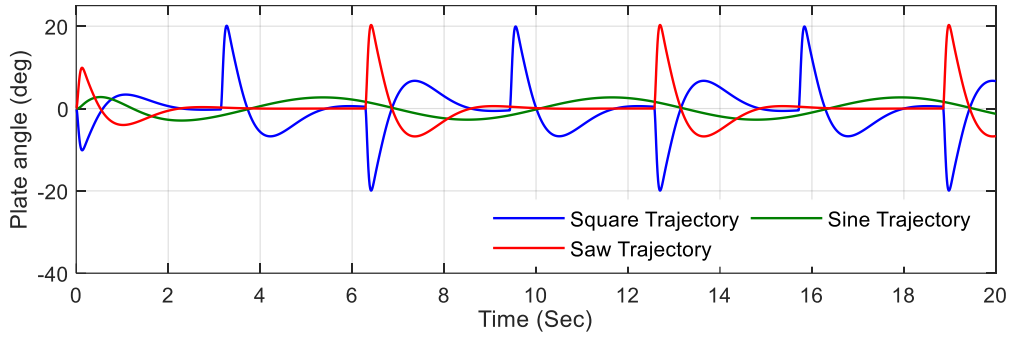


Figure 6.4. Reinforcement learning-neural network architecture for ball position and plate angle control of ball balancer system

In this experiment, three possible trajectories for ball balancer system are learned by the network. The neural network formed a structure of 3-10-6, where 3 is the input operation for the corresponding desired trajectory, 10 is the number hidden layers selected by the action of the critic function, and 6 corresponds to the control variables obtained for each hidden layer and critic function evolution feedback. Further, the interleaved learning function is adapted using full state data as discussed in Section 6.2.1. Initially, the nonlinear system measurements are obtained by simulating the model discussed above using stabilizing behavior policy. The system is learned with different balancing situations by varying the position of the ball with respect to different trajectories as shown in Figure 6.5. To ensure that the system is being operated within the affine control region without unbalancing, an exploratory noise i.e., random perturbations in the motor output are added.



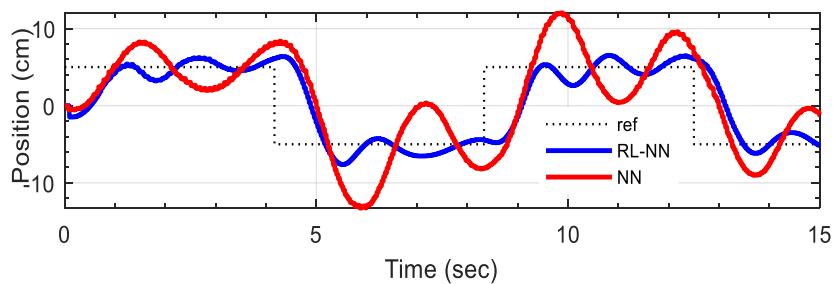
(a) Ball position



(b) Plate angle

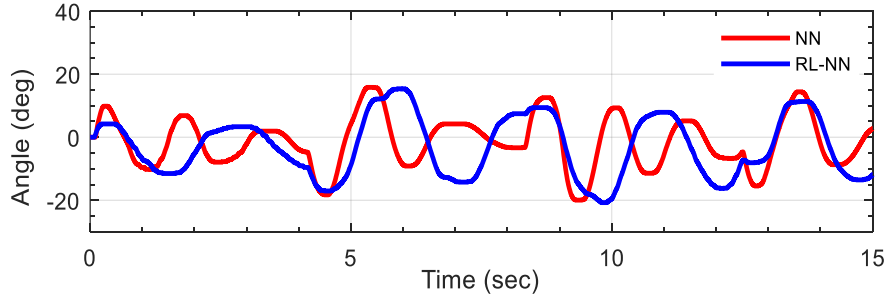
Figure 6.5. Response of ball balancer for ball movement in different trajectories

The simulation results in the Figure 6.5 display the data corresponding to ball position and plate angle for various trajectories. The x-axis SRV02 unit is responsible for x axis plate movement and y-axis SRV02 unit responsible for y axis plate movement. Due this movement of plate, the ball placed on plate will move accordingly and try to balance without falling off. Further, the real-time implementation of developed interleaved learning function is achieved with the help of Qunaser ball balancer system. The data exchange between the simulated models and the ball balancer setup is achieved by providing a whole suite of functions which use the features of supported data acquisition hardware from the C language. These functions configure the hardware and perform both synchronous and asynchronous I/O in various forms. The configuration functions provided by the Hardware-in-the-Loop - Application Programming Interface (HIL API) gives the ability to open a hardware-in-the-loop card and configure it. By using larger exploration noise, the performance of the ball balancer system for achieving balancing control is observed under square trajectory using RL-neural network (NN) controller and compared with the conventional NN controller. The ball position and plate angle for real-time operation of ball balancer system with non-linear model free control are shown in Figure 6.6.



(a) Ball position





(b) Plate angle

Figure 6.6. Ball balancer output with reinforcement learning-neural network controller in real-time

Figure 6.6(a), and 6.6(b) show the ball position, and plate angle of real-time ball balancer setup operating with developed nonlinear model free controller under square trajectory. The square input signal has been given to the setup with the frequency of 0.08Hz and amplitude of 5 cm. From the Figure 6.6(a), the ball position follows the reference trajectory with fewer oscillations which make the system stable and ball movement will be slow down. Subsequently, the plate angle variation in Figure 6.6(b) is initially between +5 degrees and -12.5 degrees and after the falling edge of the reference trajectory the plate angle varies between  $-18^{\circ}$  to  $+18^{\circ}$  due to the movement of the ball to stabilize around the trajectory. In both the cases, the response of RL-NN is better compared with the NN controller. Further, the step response characteristics of the ball position are calculated to assess the superiority of the developed approach over the classical approach. As of the Table 6.3, the response of improved RL-NN is better based on the peak time, settling time and peak overshoot.

Table 6.3. Time response analysis for reinforcement learning-neural network control of ball balancer in real-time

<b>Controllers</b>	<b>Peak time (<math>t_p</math>) (sec)</b>	<b>Peak overshoot (<math>M_p</math>) (%)</b>	<b>Steady state error (<math>e_{ss}</math>) (cm)</b>
NN	0.37 sec	22.9%	1.151 cm
RL-NN	0.1 sec	12.4%	0.487 cm

The results proved the effectiveness of model free controller using reinforcement learning for trajectory tracking and balancing control in linear and nonlinear systems.

## 6.4 CONCLUSION

In this chapter, a least square temporal difference learning algorithm, and interleaved learning algorithm are adapted for developing a model free controller for robotic applications. The Q-function of reinforcement learning was utilized with the learning algorithm and a model free state feedback controller is developed by establishing LQR and NN controllers as a baseline controller. Further, the classical least square policy iteration technique is used to establish the boundary conditions for complexities incurred by the learning algorithm. The developed model free controller is tested for trajectory tracking and balancing control of robotic systems. The controller generates the required action by learning the various states of the system through simulation.

## Chapter 7. CONCLUSION AND FUTURE SCOPE OF WORK

### 7.1 INTRODUCTION

In this chapter, the concluding remarks for various control approaches developed in the thesis are provided based on the observed simulation and real-time results. Initially, the problems associated with the mechanical systems were identified by modelling the benchmark systems of an aerial vehicle and robotic balancer. Further, the identified problems are solved with various approaches of the nonlinear control theory. The development of controllers is mainly focussed at achieving path tracking, and position control for the benchmark systems by overcoming the disadvantages due to uncertainty and failure modes. A detailed conclusion of the work discussed in Chapter 1 to 6 are given in further sections.

### 7.2 MAIN CONTRIBUTIONS OF THE WORK

The summary of this thesis is as follows:

In *chapter 1*, an introduction for the control of nonlinear systems, along with the basics nonlinear control theory is provided. The motivation for improvements in nonlinear control theory is identified and the required objectives are formulated. Further, the major contributions of the thesis are highlighted and the remaining chapters are outlined.

In *chapter 2*, some background on 2DoF helicopter and ball balancer benchmark systems and basic notions in their mechanics are given. Further, a detailed literature review bearing the linearization controller, sliding mode, model predictive, stochastic, intelligent, randomized, and learning algorithm-based control approaches is provided. From the review it is identified that, the linear controllers offer a simple way of designing closed loop control for these systems. But the complicated nonlinear dynamics of benchmark systems effect the capability of providing a plausible solution and limits the generalized applications of control laws. In addition, linearization of the nonlinear systems is also carried which heavily effected the speed of system response. This motivated for the development of several nonlinear control techniques to deal with the problems in underactuated systems. Lots of nonlinear controllers like Lagrangians, lambda method, and back stepping controller, have been evolved in the last few years. These controllers faced some drawbacks during the early stage of their development with

the limitations of lambda method in dealing with external load and back stepping controllers lagging due to additional feedback.

These issues in the conventional nonlinear techniques were overcome by carrying out feedback linearization, and partial feedback linearization for the mechanical systems. These methods transformed the nonlinear system into an equivalent linear system by cancelling the nonlinearities through a feedback control. However, the problem with lack of robustness is considered as a major drawback while adapting the procedure of feedback linearization approach and the partial feedback linearization. In order to achieve robustness while controlling, sliding mode approach is considered as a reasonable solution. But the switching surface behaviour of sliding mode controller makes it insensitive to external disturbances and parameter variations. Another approach in the line of nonlinear controllers for the benchmark systems is the passivity-based control approach. This aims at passivating the system with a storage function, which has a minimum at the desired balance point. This has a disadvantage with differential feedback as it cannot amplify the measurement noises. Considering all the drawbacks and disadvantages, the basic control schemes of fuzzy and neural integrated fuzzy are developed for the helicopter and ball balancer systems. The results identified the action of the control techniques and highlighted their drawbacks due to effect of external disturbances, parametric uncertainties, and faults on the plant operation. In light of these drawbacks, various intelligent, probabilistic, and learning algorithms are developed in Chapter 3 to 6.

In *chapter 3*, the control of benchmark systems is achieved through intelligent control approaches. Initially, a three-level discrete wavelet transform based fuzzy controller for achieving trajectory tracking with helicopter, and self-balancing control with a ball balancer system is developed. The dynamics of the systems were studied, and the control aspects are identified. As per the drawbacks of conventional control techniques observed from the literature, the novel wavelet optimized fuzzy technique is developed. Since the acquired signals in both the systems are prone to noise due to various external disturbances, the discrete wavelet transform is used as a filter for denoising the signal by adapting the minimum description length criterion. Further, the denoised signal and the modified wavelet coefficients of the signal are processed for a fuzzy controller, where the weights of the fuzzy are tuned by the wavelet transform. The pitch and yaw angles of the helicopter, and the plate angle and ball position control of the ball balancer system for DWT based fuzzy and fuzzy control techniques are analysed depending on Simulink and real-time responses. The Simulink and hardware output

performances are assessed through RMSE, and time response analysis. The DWT based fuzzy controller improves the system response by reducing settling time, peak time, steady-state-error, root mean square error, and peak overshoot.

Further, the eT2QFNN control scheme is developed for achieving attitude tracking control, and position control in unmanned helicopters, and ball balancer system respectively. The control action is developed by achieving a parallel structure controller with the conventional PD controller and the eT2QFNN scheme are developed to achieve the attitude tracking. Both pitch motor and yaw motor of the helicopter are individually controlled by operating the controller with a feedback error. Similarly, the X and Y axis actuators are controlled by the developed controller for the ball balancer system for the measured ball position. The controller generated the required rules through rule growing mechanism and parameter adjustment learning scenario. Further, a sliding surface based adaptive law is equipped to compensate the nonlinearity of the benchmark systems operating with eT2QFNN and PD controller. The stability of the developed approach is assessed using the Lyapunov stability method. Besides, Simulation and experimental analysis are conducted with the developed approach and the corresponding results are compared with the conventional interval type-2 fuzzy logic control technique with Gaussian membership function which depicted the robustness of the proposed approach in achieving attitude tracking.

In *chapter 4*, the control of benchmark systems is achieved through probabilistic control approaches. Here, a randomized algorithm based probabilistic approach is developed for parametric uncertainties in unmanned helicopter systems. The approach is developed considering the stochastic characterization of bounded uncertainty with white Gaussian noise and parametric disturbance. Further, it is assumed that the plant dynamics are exactly known for developing the controller. This provided a new paradigm for gain synthesizing the controller which solved the problem of trajectory tracking for unmanned helicopters. Further, the operation of the developed controller is tested with the 2DoF helicopter benchmark system by operating it to follow a predefined trajectory. To assess the robustness of the controller operating under uncertainties, the reachability analysis is developed considering the reach tubes and reach sets of the ellipsoidal method. The results identified the efficiency of the proposed approach with time domain characteristics for both simulation and real-time experiments. Moreover, the developed controller has better response and provide less error with the adequate time response when compared to conventional control techniques.

Similarly, a simultaneous perturbation stochastic approximation approach for unknown but bounded disturbances in a typical closed loop system is developed for position and balancing control of the ball balancer system. The developed approach achieved adaptive control by constructing a sequence of estimates and formulating an optimization problem. To enhance the performance of the developed controller, additional generalizations and convergence conditions are mentioned. Stability analysis is carried out considering the linear matrix inequalities and Lyapunov stability. Further, the developed approach is assessed with a closed loop balancing control problem with 2DoF ball balancer system and PID controller. The developed method minimizes the optimization problem to achieve the gain values of the PID controller. The simulation and, real-time experiments have been conducted for balancing control using the developed adaptive controller. The results depicted improved tracking response of the SPSA-PID over conventional PID. The PID gains optimized using SPSA provide better control when compared to conventional control approach in terms of tracking response under random uncertainties.

In *chapter 5*, the control of benchmark systems is achieved through fault classification based reconfigurable control approach. This chapter developed a support vector data description-based classification approach for identifying the motor faults during the operation of unmanned helicopters. Further, the classified faults are trained with a neural integrated fuzzy controller to enhance the trajectory tracking capabilities of the helicopter. Initially, the numerical simulation of the helicopter was modeled to operate in normal and with pitch and yaw motor failures. These failures showed an effect on the output characteristics of the helicopter. To analyze these effects, the energy and entropy features of all the characteristics are extracted using wavelet transform-based feature extraction process. The extracted features form a non-linear feature vector which are mapped into a high dimensional feature space using the Gaussian kernel-based support vector data descriptor. The developed classifier showed 98.6% training and 98.96% testing accuracy and had better performance when compared to the conventional approaches applied on the same data set. Further, to control the system during a fault, a neural integrated fuzzy controller is developed by combining the learning capability of artificial neural networks and the inference ability of the fuzzy controller. The developed controller depicted efficient path tracking with less settling time.

Similar to the fault classification-based control approach developed for the 2DoF helicopter, an SVM-wavelet fuzzy based controller is developed with the 2DoF ball balancer system for

achieving their steady state operation. The fault classification algorithm is developed by simulating all the possible faults and operating scenarios, extracting their features using discrete wavelet transform, and by training the features with support vector machine classifier. Further, to achieve steady state operation of the vehicle in the case of faults, a wavelet based fuzzy controller is developed. This accommodates the operation of the system as per the fault or condition of the system. The developed controller when implemented with the laboratory setup for defined set of faults depicted high precision and accuracy in solving the problem of position tracking and balancing of the ball with less settling time, improved steady state error, position error and low angle variation.

In *chapter 6*, a least square temporal difference learning algorithm is adapted for developing a model free controller for the benchmark systems. The Q-function of reinforcement learning was utilized with the learning algorithm and a model free state feedback controller is developed by establishing linear quadratic regulator as a baseline controller. The classical least square policy iteration technique is used to establish the boundary conditions for complexities incurred by the learning algorithm. The developed model free controller is tested for trajectory tracking and balancing control of 2DoF helicopter, and ball balancer benchmark systems respectively. In both the cases, the controller generates the required action by learning the various states of the system through simulation. The performance of the developed controllers is assessed with the real-time experimental analysis. The step response characteristics adhere to the superiority of the developed approach.

### 7.3 FUTURE SCOPE OF WORK

This section provides potential future directions of research in continuation of this work. For the trajectory tracking approach with both intelligent and probabilistic controller, the research can be extended to improve the tracking efficiency with steady state performance and multiple degrees of freedom. Besides, the operation of underactuated systems in strong and weak fields that involve zero dynamics can be a possible extension for the research. Furthermore, migration strategy-based algorithms like 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 and teaching-learning-based optimization (TLBO) can be implement with the nonlinear systems.

In learning algorithm-based control approaches, the work can be extended for increased number of faults by developing an advanced fault-tolerant control process. These control approaches can be further developed with the effects of bounded control inputs, and inevitable parametric uncertainties.

Apart from the above, the reliability analysis of components in the mechanical systems and the effect of various control approaches on their end of life and remaining useful life can be suggested as a future analysis for possible extension to some of the learning approaches.



## LIST OF PUBLICATIONS

The work conducted in this research has resulted in the following major publications:

Journals:

- Singh, R., and B. Bhushan, "Evolving Intelligent System for trajectory tracking of Unmanned Aerial Vehicles," in IEEE Transactions on Automation Science and Engineering, (In-press).
- 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, (doi: 10.1109/TIM.2020.3025656)
- Singh, R., and B. Bhushan, "Condition Monitoring based Control using Wavelets and Machine Learning for Unmanned Surface Vehicles," in IEEE Transactions on Industrial Electronics, (doi: 10.1109/TIE.2020.3001855).
- R. Singh, B. Bhushan, Randomized algorithms for probabilistic analysis of parametric uncertainties with unmanned helicopters, Mech. Syst. Signal Process. 152 (2021) 107459. <https://doi.org/10.1016/j.ymssp.2020.107459>.
- Singh, R., and B. Bhushan, "Adaptive Control Using Stochastic Approach for Unknown but Bounded Disturbances and its Application in Balancing Control", Asian Journal of Control, (Wiley). (In-press)
- Singh, R., and Bhushan, B. (2020). Improving Self Balancing and Position Tracking Control for Ball Balancer Application with Discrete Wavelet Transform Based Fuzzy Logic Controller. International Journal of Fuzzy System (Springer). (Accepted)
- R. Singh, B. Bhushan, Improved ant colony optimization for achieving self-balancing and position control for balancer systems, J. Ambient Intell. Humaniz. Comput. (2020). <https://doi.org/10.1007/s12652-020-02566-y>.
- Singh, R., and Bhushan, B. (2020). Randomized Probabilistic Approach for Parametric Uncertainties in Unmanned Helicopters. Optimal Control, Applications and Methods (Wiley). (10.1002/oca.2678)
- Singh, R., and Bhushan, B. (2020). A Novel Fault Classification Based Fault Tolerant Control for 2DoF Helicopter Systems. International Journal on Adaptive Control Signal Processing (Wiley), Volume 34, Issue 8. (<https://doi.org/10.1002/acs.3121>).
- Singh, R., Bhushan, B. (2018). Real-time control of ball balancer using neural integrated fuzzy controller. Artificial Intelligence Review (Springer), Volume 53 (January 2020), pp 351–368. (<https://doi.org/10.1007/s10462-018-9658-7>).

#### Book series:

- Singh, R., Bhushan, B. (2021) Application of Stochastic Approximation for Self-tuning of PID in Unmanned Surface Vehicles. In: Gupta D., Khanna A., Bhattacharyya S., Hassanien A., Anand S., Jaiswal A. (eds) International Conference on Innovative Computing and Communications. Advances in Intelligent Systems and Computing, vol 1165. Springer, Singapore. pp. 965–973, [https://doi.org/10.1007/978-981-15-5113-0\\_81](https://doi.org/10.1007/978-981-15-5113-0_81).
- Singh, R., Bhushan, B. (2021) Fault Classification Using Support Vectors for Unmanned Helicopters. In: Singh V., Asari V., Kumar S., Patel R. (eds) Computational Methods and Data Engineering. Advances in Intelligent Systems and Computing, vol 1227. Springer, Singapore. pp 369-384 [https://doi.org/10.1007/978-981-15-6876-3\\_28](https://doi.org/10.1007/978-981-15-6876-3_28).
- Singh, R., Bhushan, B. (2019) Adaptive Neuro-Fuzzy-PID and Fuzzy-PID-Based Controller Design for Helicopter System. In: Mishra S., Sood Y., Tomar A. (eds) Applications of Computing, Automation and Wireless Systems in Electrical Engineering. Lecture Notes in Electrical Engineering, vol 553. Springer, Singapore. pp 281-293, [https://doi.org/10.1007/978-981-13-6772-4\\_25](https://doi.org/10.1007/978-981-13-6772-4_25).

#### Conference proceedings:

- Singh, R., Bhushan, B. (2020) Control of Unmanned Surface Vehicles with Randomised Probabilistic Approach. 3rd International Conference on Intelligent Circuits and Systems 2020 (ICICS-2020), Lovely Professional University, Phagwara, India, June 26-27, 2020. (Presented)
- Singh, R., Bhushan, B. and Varshney, A. (2019) Real-Time Simulation and Control of Helicopter Systems. International Conference on Power Electronics, Control and Automation (ICPECA-2019), Jamia Millia Islamia (A Central University), New Delhi, India, November 16-17, 2019. DOI: 10.1109/ICPECA47973.2019.8975393.
- Singh, R. and Bhushan, B. (2019) Fault Classification for Unmanned Surface Vehicles using Supervised Learning Methods. International Conference on Power Electronics, Control and Automation (ICPECA-2019), Jamia Millia Islamia (Central University), New Delhi, India, November 16-17, 2019. DOI: 10.1109/ICPECA47973.2019.8975587.

## REFERENCES

- [1] R. M. Murray, K. J. Astrom, S. P. Boyd, R. W. Brockett, and G. Stein, “Future directions in control in an information-rich world,” *IEEE Control Syst. Mag.*, vol. 23, no. 2, pp. 20–33, 2003, doi: 10.1109/MCS.2003.1188769.
- [2] O. Nelles, *Nonlinear System Identification*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001.
- [3] M. A. Dahleh and I. J. Diaz-Bobillo, *Control of uncertain systems: a linear programming approach*. Prentice Hall, 1995.
- [4] 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.
- [5] 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.
- [6] F. Albertini and E. D. Sontag, “Discrete-Time Transitivity and Accessibility: Analytic Systems,” *SIAM J. Control Optim.*, vol. 31, no. 6, pp. 1599–1622, Nov. 1993, doi: 10.1137/0331075.
- [7] N. Shimkin, “Nonlinear Control Systems,” in *Encyclopedia of Neuroscience*, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 2886–2889.
- [8] 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.
- [9] M. N. Bismarck-Nasr, “Nonlinear Systems,” *Struct. Dyn. Aeronaut. Eng.*, pp. 119–138, 1999, doi: 10.2514/5.9781600862458.0119.0138.
- [10] M. Krstic, P. V Kokotovic, and I. Kanellakopoulos, *Nonlinear and Adaptive Control Design*, 1st ed. USA: John Wiley & Sons, Inc., 1995.
- [11] 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.
- [12] A. R. Teel, “A nonlinear small gain theorem for the analysis of control systems with saturation,” *IEEE Trans. Automat. Contr.*, vol. 41, no. 9, pp. 1256–1270, 1996, doi: 10.1109/9.536496.
- [13] F. Mazenc and L. Praly, “Adding integrations, saturated controls, and stabilization for feedforward systems,” *IEEE Trans. Automat. Contr.*, vol. 41, no. 11, pp. 1559–1578,

- 1996, doi: 10.1109/9.543995.
- [14] H. J. Sussmann, “Limitations on the stabilizability of globally-minimum-phase systems,” *IEEE Trans. Automat. Contr.*, vol. 35, no. 1, pp. 117–119, 1990, doi: 10.1109/9.45159.
- [15] B. Zhou, G. Duan, and L. Zhang, “Nonlinear control for global stabilization of multiple-integrator system by bounded controls,” *J. Control Theory Appl.*, vol. 6, no. 3, pp. 293–299, Aug. 2008, doi: 10.1007/s11768-008-6210-3.
- [16] R. Sepulchre, M. Janković, and P. V. Kokotović, *Constructive Nonlinear Control*. London: Springer London, 1997.
- [17] R. Sepulchre, “Slow peaking and low-gain designs for global stabilization of nonlinear systems,” *IEEE Trans. Automat. Contr.*, vol. 45, no. 3, pp. 453–461, Mar. 2000, doi: 10.1109/9.847724.
- [18] M. W. Spong and L. Praly, “Control of underactuated mechanical systems using switching and saturation,” in *Control Using Logic-Based Switching*, London: Springer-Verlag, pp. 162–172.
- [19] 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.
- [20] M. W. Spong, “The swing up control problem for the Acrobot,” *IEEE Control Syst.*, vol. 15, no. 1, pp. 49–55, Feb. 1995, doi: 10.1109/37.341864.
- [21] M. W. Spong and D. J. Block, “The Pendubot: a mechatronic system for control research and education,” in *Proceedings of 1995 34th IEEE Conference on Decision and Control*, vol. 1, pp. 555–556, doi: 10.1109/CDC.1995.478951.
- [22] C. C. Chung and J. Hauser, “Nonlinear control of a swinging pendulum,” *Automatica*, vol. 31, no. 6, pp. 851–862, Jun. 1995, doi: 10.1016/0005-1098(94)00148-C.
- [23] K. J. Åström and K. Furuta, “Swinging up a pendulum by energy control,” *Automatica*, vol. 36, no. 2, pp. 287–295, Feb. 2000, doi: 10.1016/S0005-1098(99)00140-5.
- [24] 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.
- [25] 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.
- [26] 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.
- [27] M. Jankovic, D. Fontaine, and P. V. Kokotovic, “TORA example: cascade- and passivity-based control designs,” *IEEE Trans. Control Syst. Technol.*, vol. 4, no. 3, pp. 292–297, May 1996, doi: 10.1109/87.491203.
- [28] 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.
- [29] 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.
- [30] O. Egeland, M. Dalsmo, and O. J. Sørдалen, “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.
- [31] 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. 967–972, doi: 10.1109/CDC.1996.574602.
- [32] 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.
- [33] 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.
- [34] R. W. Brockett, “Asymptotic stability and feedback stabilization,” *Differ. Geom. Control Theory*, pp. 181–191, 1983.
- [35] 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.
- [36] 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.
- [37] 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.
- [38] 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.
- [39] 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.
- [40] 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.
- [41] 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.
- [42] 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.
- [43] 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.
- [44] 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.
- [45] L. Keviczky and C. Bányász, *Two-Degree-of-Freedom Control Systems*. Elsevier, 2015.
- [46] 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.
- [47] 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, [Online]. Available: <http://www.researchpub.org/journal/jac/jac.html>.
- [48] 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.
- [49] J. Yang, J. Xu, and Z. Li, “Two-Degree-of-Freedom Based Cross-Coupled Control for

- 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.
- [50] 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.
- [51] 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.
- [52] 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.
- [53] 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.
- [54] 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.
- [55] 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.
- [56] 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.
- [57] Quanser Inc., “2-DOF Helicopter,” 2011.
- [58] Faulhaber, “Model 006C Motor Series 2842.” 2003.
- [59] Quanser Inc, “‘VOLTPAQ-X2’ Reliable linear voltage amplifier for superior real-time Hardware-in-the-Loop (HIL) performance.” p. 2020, 2020.
- [60] Pittman, “LO-COG DC Servo Motors 8000, 9000, 14000.” 2010.
- [61] P. Das, R. K. Mehta, and O. P. Roy, “Optimized methods for the pre-eminent performance of LQR control applied in a MIMO system,” *Int. J. Dyn. Control*, vol. 7, no. 4, pp. 1501–1520, Dec. 2019, doi: 10.1007/s40435-019-00587-w.
- [62] S. K. Choudhary, “Optimal feedback control of a twin rotor MIMO system,” *Int. J. Model. Simul.*, vol. 37, no. 1, pp. 46–53, 2017, doi: 10.1080/02286203.2016.1233008.
- [63] B. Pratap and S. Purwar, “State observer based robust feedback linearization controller for twin rotor MIMO system,” in *2012 IEEE International Conference on Control*

- Applications*, Oct. 2012, pp. 1074–1079, doi: 10.1109/CCA.2012.6402397.
- [64] S. S. Butt and H. Aschemann, “Multi-Variable Integral Sliding Mode Control of a Two Degrees of Freedom Helicopter,” *IFAC-PapersOnLine*, vol. 48, no. 1, pp. 802–807, 2015, doi: 10.1016/j.ifacol.2015.05.129.
- [65] Feedback Instruments, “Twin Rotor MIMO Systems Experiments,” United Kingdom, 2006. [Online]. Available: [www.fbk.com](http://www.fbk.com).
- [66] Quanser educational solutions, “2D Ball Balancer User Manual,” Markham, Ontario., 2013.
- [67] Y. Arkun and J. Downs, “A general method to calculate input-output gains and the relative gain array for integrating processes,” *Comput. Chem. Eng.*, vol. 14, no. 10, pp. 1101–1110, Oct. 1990, doi: 10.1016/0098-1354(90)85006-V.
- [68] S. Spedicato and G. Notarstefano, “Minimum-Time Trajectory Generation for Quadrotors in Constrained Environments,” *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 4, pp. 1335–1344, Jul. 2018, doi: 10.1109/TCST.2017.2709268.
- [69] 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.
- [70] Shaun Sawyer, “Gain-Scheduled Control of a Quadcopter UAV,” University of Waterloo, 2015.
- [71] 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. 114–119, doi: 10.1109/WAC.2014.6935709.
- [72] 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.
- [73] 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.
- [74] 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.
- [75] 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.



- [76] 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.
- [77] 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.
- [78] 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.
- [79] 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.
- [80] 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.
- [81] 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.
- [82] 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.
- [83] 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.
- [84] 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.
- [85] 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.
- [86] 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.
- [87] 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.
- [88] 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, 2017, doi: 10.1016/j.isatra.2017.04.013.
- [89] 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.
- [90] 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.
- [91] 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.
- [92] 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.
- [93] 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.
- [94] 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.
- [95] 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.
- [96] 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.
- [97] 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.
- [98] 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.
- [99] 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.
- [100] 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.
- [101] 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.
- [102] 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, p. 34, Nov. 2008, doi: 10.5772/10583.
- [103] T. Oktay and C. Sultan, “Model Predictive Control of Maneuvering Helicopters,” Aug. 2012, doi: 10.2514/6.2012-4530.
- [104] 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.
- [105] 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.
- [106] 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.
- [107] P. K. Paul and J. Jacob, “H<sub>2</sub> Vs H<sub>∞</sub> 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.
- [108] 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.

- [109] 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>.
- [110] 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.
- [111] 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.
- [112] 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.
- [113] 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.
- [114] 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.
- [115] 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.
- [116] S. S. Ge and K.-P. Tee, “Adaptive Neural Network Control of Helicopters,” 2006, pp. 82–87.
- [117] 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.
- [118] 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.
- [119] 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.
- [120] 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.
- [121] 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.
- [122] A. Bounemour, 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.
- [123] 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.
- [124] 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.
- [125] 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.
- [126] H. A. Hagra, “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.
- [127] 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.
- [128] 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.
- [129] 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.
- [130] 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.
- [131] 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.
- [132] 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.
- [133] 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.
- [134] 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.
- [135] 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.
- [136] 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.
- [137] 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.
- [138] A. Girard and S. Martin, “Synthesis for Constrained Nonlinear Systems Using Hybridization and Robust Controllers on Simplices,” *IEEE Trans. Automat. Contr.*, vol. 57, no. 4, pp. 1046–1051, Apr. 2012, doi: 10.1109/TAC.2011.2168874.
- [139] A. Girard and S. Martin, “Motion planning for nonlinear systems using hybridizations and robust controllers on simplices,” in *2008 47th IEEE Conference on Decision and Control*, 2008, pp. 239–244, doi: 10.1109/CDC.2008.4738758.
- [140] 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.
- [141] 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.
- [142] 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.
- [143] 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.
- [144] 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.
- [145] 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&#x2013;Multi-Output System,” *IEEE Trans. Fuzzy Syst.*, vol. 18, no. 5, pp. 893–905, Oct. 2010, doi: 10.1109/TFUZZ.2010.2051447.
- [146] 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.
- [147] 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.
- [148] 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.
- [149] 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.
- [150] 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.
- [151] 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.
- [152] 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.
- [153] P. Dorato and C. T. Abdallah, *Randomized Algorithms for Analysis and Control of Uncertain Systems*, vol. 52, no. 1. 2007.
- [154] R. Tempo, G. Calafiore, and F. Dabbene, *Randomized Algorithms for Analysis and Control of Uncertain Systems*. London: Springer London, 2013.
- [155] G. C. Calafiore and M. C. Campi, “The scenario approach to robust control design,” *IEEE Trans. Automat. Contr.*, vol. 51, no. 5, pp. 742–753, 2006, doi: 10.1109/TAC.2006.875041.
- [156] H. Ishii and T. Bas, “Randomized Algorithms for Synthesis of Switching Rules for Multimodal Systems,” vol. 50, no. 6, pp. 754–767, 2005.
- [157] J. Vishnupriyan, P. S. Manoharan, and A. P. S. Ramalakshmi, “Uncertainty modeling of nonlinear 2-DOF helicopter model,” *2014 Int. Conf. Comput. Commun. Informatics Ushering Technol. Tomorrow, Today, ICCCI 2014*, pp. 1–6, 2014, doi: 10.1109/ICCCI.2014.6921840.
- [158] D. Rotondo, F. Nejjari, and V. Puig, “Quasi-LPV modeling, identification and control of a twin rotor MIMO system,” *Control Eng. Pract.*, vol. 21, no. 6, pp. 829–846, 2013, doi: 10.1016/j.conengprac.2013.02.004.
- [159] X. Yang, J. Cui, D. Lao, D. Li, and J. Chen, “Input Shaping enhanced Active Disturbance Rejection Control for a twin rotor multi-input multi-output system (TRMS),” *ISA Trans.*, vol. 62, pp. 287–298, May 2016, doi: 10.1016/j.isatra.2016.02.001.
- [160] B. Yu, S. Kim, and J. Suk, “Robust Control based on ADRC and DOBC for Small-Scale Helicopter,” *IFAC-PapersOnLine*, vol. 52, no. 12, pp. 140–145, 2019, doi: 10.1016/j.ifacol.2019.11.183.
- [161] W. Liyan, T. Gang, Y. Hao, and C. Fuyang, “An adaptive multivariable disturbance rejection algorithm for helicopter turbulence compensation control,” in *2015 34th Chinese Control Conference (CCC)*, Jul. 2015, pp. 3121–3126, doi: 10.1109/ChiCC.2015.7260121.
- [162] M.-B. Radac, R.-C. Roman, and R.-E. Precup, “Multi-input–multi-output system experimental validation of model-free control and virtual reference feedback tuning techniques,” *IET Control Theory Appl.*, vol. 10, no. 12, pp. 1395–1403, 2016, doi: 10.1049/iet-cta.2016.0028.
- [163] Z. T. Dydek, A. M. Annaswamy, and E. Lavretsky, “Adaptive Control of Quadrotor



- UAVs: A Design Trade Study With Flight Evaluations,” *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 4, pp. 1400–1406, Jul. 2013, doi: 10.1109/TCST.2012.2200104.
- [164] M. Farhood and E. Feron, “Obstacle-Sensitive Trajectory Regulation via Gain Scheduling and Semidefinite Programming,” *IEEE Trans. Control Syst. Technol.*, vol. 20, no. 4, pp. 1107–1115, Jul. 2012, doi: 10.1109/TCST.2011.2159718.
- [165] N. Liu, X. Shao, and W. Yang, “Integral Barrier Lyapunov Function Based Saturated Dynamic Surface Control for Vision-Based Quadrotors via Back-Stepping,” *IEEE Access*, vol. 6, pp. 63292–63304, 2018, doi: 10.1109/ACCESS.2018.2876762.
- [166] M. Sassano and A. Astolfi, “Dynamic Lyapunov functions: Properties and applications,” in *2012 American Control Conference (ACC)*, Jun. 2012, pp. 2571–2576, doi: 10.1109/ACC.2012.6314839.
- [167] T. Oktay and C. Sultan, “Robustness of variance constrained controllers for complex, control oriented helicopter models,” in *2013 American Control Conference*, Jun. 2013, pp. 794–799, doi: 10.1109/ACC.2013.6579933.
- [168] B. Xiao, S. Yin, S. Member, and H. Gao, “Reconfigurable Tolerant Control of Uncertain Mechanical Systems With Actuator Faults: A Sliding Mode Observer-Based Approach,” *IEEE Trans. Control Syst. Technol.*, vol. PP, no. XX, pp. 1–10, 2017, doi: 10.1016/j.yexcr.2008.02.019.
- [169] G. H. Yang and D. Ye, “Reliable  $H_\infty$  control of linear systems with adaptive mechanism,” *IEEE Trans. Automat. Contr.*, vol. 55, no. 1, pp. 242–247, 2010, doi: 10.1109/TAC.2009.2036293.
- [170] G.-H. Yang, S.-Y. Zhang, and J. Lam, “Reliable Control Using Multiple Similar Controllers,” *IFAC Proc. Vol.*, vol. 29, no. 1, pp. 3344–3349, 1996, doi: 10.1016/S1474-6670(17)58193-8.
- [171] R. J. Patton, “Fault-Tolerant Control: The 1997 Situation,” *IFAC Proc. Vol.*, vol. 30, no. 18, pp. 1029–1051, 1997, doi: 10.1016/S1474-6670(17)42536-5.
- [172] M. Benosman, “A survey of some recent results on nonlinear fault tolerant control,” *Math. Probl. Eng.*, vol. 2010, 2010, doi: 10.1155/2010/586169.
- [173] X. Yu and J. Jiang, “A survey of fault-tolerant controllers based on safety-related issues,” *Annu. Rev. Control*, vol. 39, pp. 46–57, 2015, doi: 10.1016/j.arcontrol.2015.03.004.
- [174] R. J. Patton, “Fault-Tolerant Control,” *Encycl. Syst. Control*, vol. 36, no. 3, pp. 1–8, 2014, doi: 10.1007/978-1-4471-5102-9\_226-1.
- [175] C. E. and H. A. M. T. Hamayun, “Design and Analysis of an Integral Sliding Mode

- Fault-Tolerant Control Scheme,” *IEEE Trans. Automat. Contr.*, vol. 57, no. 7, pp. 1783–1789, 2012, doi: 10.1109/TAC.2011.2180090.
- [176] H. Ríos, S. Kamal, L. M. Fridman, and A. Zolghadri, “Fault tolerant control allocation via continuous integral sliding-modes: A HOSM-Observer approach,” *Automatica*, vol. 51, pp. 318–325, 2015, doi: 10.1016/j.automatica.2014.10.085.
- [177] T. Steffen, *Control Reconfiguration of Dynamical Systems*. 2005.
- [178] B. Rabaoui, M. Rodrigues, H. Hamdi, and N. BenHadj Braiek, “A model reference tracking based on an active fault tolerant control for LPV systems,” *Int. J. Adapt. Control Signal Process.*, vol. 32, no. 6, pp. 839–857, 2018, doi: 10.1002/acs.2871.
- [179] S. Aouaouda, “Constrained Fault Tolerant Control Design for Takagi Sugeno Systems,” *2018 Int. Conf. Electr. Sci. Technol. Maghreb*, pp. 1–6, 2018.
- [180] N. Moustakis, B. Zhou, T. Le Quang, and S. Baldi, “Fault detection and identification for a class of continuous piecewise affine systems with unknown subsystems and partitions,” *Int. J. Adapt. Control Signal Process.*, vol. 32, no. 7, pp. 980–993, 2018, doi: 10.1002/acs.2881.
- [181] R. Hmidi, A. Ben Brahim, F. Ben Hmida, and A. Sellami, “Robust fault tolerant control for Lipschitz nonlinear systems with simultaneous actuator and sensor faults,” *2018 Int. Conf. Adv. Syst. Electr. Technol. IC\_ASET 2018*, pp. 277–283, 2018, doi: 10.1109/ASET.2018.8379871.
- [182] D. Tan, R. J. Patton, and X. Wang, “A relaxed solution to unknown input observers for state and fault estimation,” *IFAC-PapersOnLine*, vol. 28, no. 21, pp. 1048–1053, 2015, doi: 10.1016/j.ifacol.2015.09.665.
- [183] P. Niermeyer, M. Pak, and B. Lohmann, “A Control Effectiveness Estimator with a Moving Horizon Robustness Modification for Fault-Tolerant Hexacopter Control,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 270–275, 2017, doi: 10.1016/j.ifacol.2017.08.045.
- [184] P. Shi, M. Liu, and L. Zhang, “Fault-tolerant sliding-mode-observer synthesis of markovian jump systems using quantized measurements,” *IEEE Trans. Ind. Electron.*, vol. 62, no. 9, pp. 5910–5918, 2015, doi: 10.1109/TIE.2015.2442221.
- [185] S. M. De Oca and V. Puig, “Fault-tolerant control using a virtual actuator using LPV techniques: Application to a two-degree of freedom helicopter,” *IFAC Proc. Vol.*, vol. 18, no. PART 1, pp. 416–421, 2010, doi: 10.3182/20100906-5-JP-2022.00071.
- [186] N. E. Wu, K. Zhou, and G. Salomon, “Control reconfigurability of linear time-invariant systems,” *Automatica*, vol. 36, pp. 1767–1771, 2000, doi: doi.org/10.1016/S0005-

1098(00)00080-7.

- [187] N. Eva Wu, "Coverage in fault-tolerant control," *Automatica*, vol. 40, no. 4, pp. 537–548, 2004, doi: 10.1016/j.automatica.2003.11.015.
- [188] H. R. Shaker, "Control reconfigurability of bilinear systems," *J. Mech. Sci. Technol.*, vol. 27, no. 4, pp. 1117–1123, 2013, doi: 10.1007/s12206-013-0216-6.
- [189] W. Ren, H. Yang, B. Jiang, and M. Staroswiecki, "Fault recoverability analysis of switched nonlinear systems," *Int. J. Syst. Sci.*, vol. 48, no. 3, pp. 471–484, 2017, doi: 10.1080/00207721.2016.1186246.
- [190] J. Huang and N. E. Wu, "Fault-tolerant placement of phasor measurement units based on control reconfigurability," *Control Eng. Pract.*, vol. 21, no. 1, pp. 1–11, 2013, doi: 10.1016/j.conengprac.2012.09.001.
- [191] H. Yang, B. Jiang, M. Staroswiecki, and Y. Zhang, "Fault recoverability and fault tolerant control for a class of interconnected nonlinear systems," *Automatica*, vol. 54, pp. 49–55, 2015, doi: 10.1016/j.automatica.2015.01.037.
- [192] X. Qi *et al.*, "A review on fault diagnosis and fault tolerant control methods for single-rotor aerial vehicles," *J. Intell. Robot. Syst. Theory Appl.*, vol. 73, no. 1–4, pp. 535–555, 2014, doi: 10.1007/s10846-013-9954-z.
- [193] Z. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques-part 1&2: Fault diagnosis with knowledge-based and hybrid/active approaches," *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3768–3774, 2015, doi: 10.1109/TIE.2015.2419013.
- [194] M. J. Smith, C. S. Byington, P. Kalgren, A. Parulekar, and M. DeChristopher, "Layered Classification for Improved Diagnostic Isolation in Drivetrain Components," *2006 IEEE Aerosp. Conf.*, pp. 1–8, 2006, doi: 10.1109/aero.2006.1656097.
- [195] S. Lee, "Drawbacks of Principal component analysis," *arXiv:1005.1770*, no. 2, pp. 2–5, 2010, [Online]. Available: <http://arxiv.org/abs/1005.1770>.
- [196] V. Camerini, G. Coppotelli, and S. Bendisch, "Fault detection in operating helicopter drivetrain components based on support vector data description," *Aerosp. Sci. Technol.*, vol. 73, pp. 48–60, 2018, doi: 10.1016/j.ast.2017.11.043.
- [197] P. T. Monsen, E. S. Manolakos, and M. Dzwonczyk, "Helicopter gearbox fault detection and diagnosis using analog neural networks," pp. 381–385, 2002, doi: 10.1109/acssc.1993.342539.
- [198] L. Zhou, F. Duan, D. Mba, W. Wang, and S. Ojolo, "Using frequency domain analysis techniques for diagnosis of planetary bearing defect in a CH-46E helicopter aft

- gearbox,” *Eng. Fail. Anal.*, vol. 92, no. May, pp. 71–83, 2018, doi: 10.1016/j.engfailanal.2018.04.051.
- [199] X. Qi *et al.*, “Fault diagnosis and fault tolerant control methods for manned and unmanned helicopters: A literature review,” *Conf. Control Fault-Tolerant Syst. SysTol*, pp. 132–139, 2013, doi: 10.1109/SysTol.2013.6693906.
- [200] Y. Xu, B. Jiang, G. Tao, and Z. Gao, “Fault Tolerant Control for a Class of Nonlinear Systems with Application to Near Space Vehicle,” *Circuits, Syst. Signal Process.*, vol. 30, no. 3, pp. 655–672, Jun. 2011, doi: 10.1007/s00034-010-9239-8.
- [201] J. D. Boskovic and R. K. Mehra, “A Decentralized Fault-Tolerant Control System for Accommodation of Failures in Higher-Order Flight Control Actuators,” *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 5, pp. 1103–1115, Sep. 2010, doi: 10.1109/TCST.2009.2033805.
- [202] 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,” in *2015 10th Asian Control Conference (ASCC)*, May 2015, pp. 1–6, doi: 10.1109/ASCC.2015.7244381.
- [203] E. D’Amato, M. Mattei, I. Notaro, and V. Scordamaglia, “UAV Sensor FDI in Duplex Attitude Estimation Architectures Using a Set-Based Approach,” *IEEE Trans. Instrum. Meas.*, vol. 67, no. 10, pp. 2465–2475, Oct. 2018, doi: 10.1109/TIM.2018.2838718.
- [204] A. Bounemour, 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, pp. 45–61, Aug. 2018, doi: 10.1016/j.isatra.2018.04.014.
- [205] Bin Zhang *et al.*, “Application of Blind Deconvolution Denoising in Failure Prognosis,” *IEEE Trans. Instrum. Meas.*, vol. 58, no. 2, pp. 303–310, Feb. 2009, doi: 10.1109/TIM.2008.2005963.
- [206] J. Huang and M. Wang, “Multiple Classifiers Combination Model for Fault Diagnosis Using Within-class Decision Support,” in *2010 International Conference of Information Science and Management Engineering*, Aug. 2010, pp. 226–229, doi: 10.1109/ISME.2010.99.
- [207] M. Electronics, “Faulhaber DC-Micromotors Series 2338.”
- [208] Quanser Inc., “Peripherals to Accelerate Control System Design and Implementation,” p. 12, 2010, [Online]. Available: [www.quanser.com](http://www.quanser.com).
- [209] S. Mochizuki and H. Ichihara, “Generalized Kalman-Yakubovich-Popov Lemma Based I-PD Controller Design for Ball and Plate System,” *J. Appl. Math.*, vol. 2013, pp. 1–9,

- 2013, doi: 10.1155/2013/854631.
- [210] F. Chaumette and S. Hutchinson, “Visual servo control. I. Basic approaches,” *IEEE Robot. & Autom. Mag.*, vol. 13, no. 4, pp. 82–90, Dec. 2006, doi: 10.1109/MRA.2006.250573.
- [211] A. K. Pinagapani, G. Mani, C. K R, 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.
- [212] R. Singh and B. Bhushan, “Real-time control of ball balancer using neural integrated fuzzy controller,” *Artif. Intell. Rev.*, 2018, doi: 10.1007/s10462-018-9658-7.
- [213] 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.
- [214] 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-ZA-1003.01074.
- [215] 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.
- [216] 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.
- [217] 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.
- [218] 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.
- [219] 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.

- [220] A. Wang, X. Li, and X. Cao, “Backstepping-based robust  $h_{\infty}$  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.
- [221] 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.
- [222] Y. Pan, T. Sun, Y. Liu, and H. Yu, “Composite learning from adaptive backstepping neural network control,” *Neural Networks*, vol. 95, pp. 134–142, 2017, doi: 10.1016/j.neunet.2017.08.005.
- [223] 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.
- [224] S. Mochizuki and H. Ichihara, “Generalized Kalman-Yakubovich-Popov Lemma Based I-PD Controller Design for Ball and Plate System,” *J. Appl. Math.*, vol. 2013, p. 9, 2013, doi: <https://doi.org/10.1155/2013/854631>.
- [225] Y. Wang, Q. Jin, and R. Zhang, “Improved fuzzy PID controller design using predictive functional control structure,” *ISA Trans.*, vol. 71, pp. 354–363, Nov. 2017, doi: 10.1016/j.isatra.2017.09.005.
- [226] M. M. Abdo, A. R. Vali, A. R. Toloei, and M. R. Arvan, “Stabilization loop of a two axes gimbal system using self-tuning PID type fuzzy controller,” *ISA Trans.*, vol. 53, no. 2, pp. 591–602, Mar. 2014, doi: 10.1016/j.isatra.2013.12.008.
- [227] 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.ymsp.2017.12.014.
- [228] 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.
- [229] J. Zavacka, M. Bakosova, and K. Matejickova, “Robust PID Controller Design for Unstable Processes with Parametric Uncertainty,” *Procedia Eng.*, vol. 42, pp. 1572–1578, 2012, doi: 10.1016/j.proeng.2012.07.550.
- [230] C.-F. Hsu and B.-K. Lee, “FPGA-based adaptive PID control of a DC motor driver via sliding-mode approach,” *Expert Syst. Appl.*, vol. 38, no. 9, pp. 11866–11872, Sep. 2011, doi: 10.1016/j.eswa.2011.02.185.
- [231] E. Cruz-Zavala, J. A. Moreno, and L. Fridman, “Uniform sliding mode controllers and

- uniform sliding surfaces,” *IMA J. Math. Control Inf.*, vol. 29, no. 4, pp. 491–505, 2012, doi: 10.1093/imamci/dns005.
- [232] H. Bang and Y. S. Lee, “Implementation of a Ball and Plate Control System Using Sliding Mode Control,” *IEEE Access*, vol. 6, pp. 32401–32408, 2018, doi: 10.1109/ACCESS.2018.2838544.
- [233] M. L. Hammadih, K. Al Hosani, and I. Boiko, “Interpolating sliding mode observer for a ball and beam system,” *Int. J. Control*, vol. 89, no. 9, pp. 1879–1889, 2016, doi: 10.1080/00207179.2016.1161235.
- [234] A. Das and P. Roy, “Improved Performance of Cascaded Fractional-Order SMC over Cascaded SMC for Position Control of a Ball and Plate System,” *IETE J. Res.*, vol. 63, no. 2, pp. 238–247, 2017, doi: 10.1080/03772063.2016.1258336.
- [235] R. Ortega, M. W. Spong, F. Gomez-Estern, and G. Blankenstein, “Stabilization of a class of underactuated mechanical systems via interconnection and damping assignment,” *IEEE Trans. Automat. Contr.*, vol. 47, no. 8, pp. 1218–1233, Aug. 2002, doi: 10.1109/TAC.2002.800770.
- [236] J. Zhang, Y. Lin, and G. Feng, “Analysis and synthesis of memory-based fuzzy sliding mode controllers,” *IEEE Trans. Cybern.*, vol. 45, no. 12, pp. 2880–2889, 2015, doi: 10.1109/TCYB.2015.2388691.
- [237] S. T. Kao and M. T. Ho, “Second-Order Sliding Mode Control for Ball-Balancing System,” *2018 IEEE Conf. Control Technol. Appl. CCTA 2018*, no. 1, pp. 1730–1735, 2018, doi: 10.1109/CCTA.2018.8511547.
- [238] H. Bang and Y. S. Lee, “Embedded Model Predictive Control for Enhancing Tracking Performance of a Ball-and-Plate System,” *IEEE Access*, vol. 7, pp. 39652–39659, 2019, doi: 10.1109/ACCESS.2019.2907111.
- [239] D. Limon, M. Pereira, D. Munoz De La Pena, T. Alamo, C. N. Jones, and M. N. Zeilinger, “MPC for Tracking Periodic References,” *IEEE Trans. Automat. Contr.*, vol. 61, no. 4, pp. 1123–1128, 2016, doi: 10.1109/TAC.2015.2461811.
- [240] H. Chen and C. W. Scherer, “Moving horizon H-infinity control with performance adaptation for constrained linear systems,” *Automatica*, vol. 42, no. 6, pp. 1033–1040, Jun. 2006, doi: 10.1016/j.automatica.2006.03.001.
- [241] Z. Li, Z. Duan, G. Wen, and J. Wang, “Distributed  $H_\infty$  and  $H_2$  consensus control in directed networks,” *IET Control Theory Appl.*, vol. 8, no. 3, pp. 193–201, Feb. 2014, doi: 10.1049/iet-cta.2013.0258.
- [242] X. Su, L. Wu, P. Shi, and C. L. P. Chen, “Model Approximation for Fuzzy Switched

- Systems With Stochastic Perturbation,” *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 5, pp. 1458–1473, Oct. 2015, doi: 10.1109/TFUZZ.2014.2362153.
- [243] X. Su, P. Shi, L. Wu, and Y.-D. Song, “Fault Detection Filtering for Nonlinear Switched Stochastic Systems,” *IEEE Trans. Automat. Contr.*, vol. 61, no. 5, pp. 1310–1315, May 2016, doi: 10.1109/TAC.2015.2465091.
- [244] W. Qin, J. Liu, G. Liu, B. He, and L. Wang, “Robust Parameter Dependent Receding Horizon  $H_\infty$  Control of Flexible Air-Breathing Hypersonic Vehicles with Input Constraints,” *Asian J. Control*, vol. 17, no. 2, pp. 508–522, Mar. 2015, doi: 10.1002/asjc.1084.
- [245] M. T. Hagan and H. B. Demuth, “Neural networks for control,” *Proc. Am. Control Conf.*, vol. 3, pp. 1642–1656, 1999, doi: 10.1201/9781420035520.ch5.
- [246] M. T. Hagan, H. B. Demuth, and O. D. E. Jesús, “AN INTRODUCTION TO THE USE OF NEURAL NETWORKS IN CONTROL SYSTEMS,” *Endeavour*, vol. 25, no. 95, p. 58, 1966, doi: 10.1016/0160-9327(66)90069-X.
- [247] P. H. Eaton, D. V. Prokhorov, and D. C. Wunsch, “Neurocontroller alternatives for ‘fuzzy’ ball-and-beam systems with nonuniform nonlinear friction,” *IEEE Trans. Neural Networks*, vol. 11, no. 2, pp. 423–435, 2000, doi: 10.1109/72.839012.
- [248] T. L. Chien, C. C. Chen, Y. C. Huang, and W. J. Lin, “Stability and almost disturbance decoupling analysis of nonlinear system subject to feedback linearization and feedforward neural network controller,” *IEEE Trans. Neural Networks*, vol. 19, no. 7, pp. 1220–1230, 2008, doi: 10.1109/TNN.2008.2000207.
- [249] K. C. Ng, M. M. Trivedi, and C. Vision, “Fuzzy-Neural Controller and Real-Time Implementation of A Ball Balancing Beam,” *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 1–17, 1996.
- [250] M. A. Moreno-Armendáriz, C. A. Pérez-Olvera, F. O. Rodríguez, and E. Rubio, “Indirect hierarchical FCMAC control for the ball and plate system,” *Neurocomputing*, vol. 73, no. 13–15, pp. 2454–2463, 2010, doi: 10.1016/j.neucom.2010.03.023.
- [251] A. B. Rad, P. T. Chan, W. L. Lo, and C. K. Mok, “An online learning fuzzy controller,” *IEEE Trans. Ind. Electron.*, vol. 50, no. 5, pp. 1016–1021, 2003, doi: 10.1109/TIE.2003.817690.
- [252] X. Fan, N. Zhang, and S. Teng, “Trajectory planning and tracking of ball and plate system using hierarchical fuzzy control scheme,” *Fuzzy Sets Syst.*, vol. 144, no. 2, pp. 297–312, 2004, doi: 10.1016/S0165-0114(03)00135-0.
- [253] Z. Zhang and D. Yuan, “Modelling and control scheme of the ball–plate trajectory-



- tracking pneumatic system with a touch screen and a rotary cylinder,” *IET Control Theory Appl.*, vol. 4, no. 4, pp. 573–589, 2010, doi: 10.1049/iet-cta.2008.0540.
- [254] X. Dong, Y. Zhao, Y. Xu, Z. Zhang, and P. Shi, “Design of PSO fuzzy neural network control for ball and plate system,” *Int. J. Innov. Comput. Inf. Control*, vol. 7, no. 12, pp. 7091–7103, 2011.
- [255] B. Ming, L. Huiqiu, S. Jintao, and T. Yantao, “Motion control of ball and plate system using supervisory fuzzy controller,” *Proc. World Congr. Intell. Control Autom.*, vol. 2, no. 20010320, pp. 8127–8131, 2006, doi: 10.1109/WCICA.2006.1713557.
- [256] M. A. Moreno-armend, E. Rubio, and A. P, “Design and Implementation of a Visual Fuzzy Control in FPGA for the Ball and Plate System,” pp. 3–8, 2010, doi: 10.1109/ReConFig.2010.69.
- [257] J. Cervantes, W. Yu, S. Salazar, and I. Chairez, “Takagi-Sugeno Dynamic Neuro-Fuzzy Controller of Uncertain Nonlinear Systems,” *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 6, pp. 1601–1615, 2017, doi: 10.1109/TFUZZ.2016.2612697.
- [258] K. Subramanian, R. Savitha, and S. Suresh, “A complex-valued neuro-fuzzy inference system and its learning mechanism,” *Neurocomputing*, vol. 123, pp. 110–120, 2014, doi: 10.1016/j.neucom.2013.06.009.
- [259] S. Kar, S. Das, and P. K. Ghosh, “Applications of neuro fuzzy systems: A brief review and future outline,” *Appl. Soft Comput. J.*, vol. 15, pp. 243–259, 2014, doi: 10.1016/j.asoc.2013.10.014.
- [260] G. Bosque, I. Del Campo, and J. Echanobe, “Fuzzy systems, neural networks and neuro-fuzzy systems: A vision on their hardware implementation and platforms over two decades,” *Eng. Appl. Artif. Intell.*, vol. 32, no. 1, pp. 283–331, 2014, doi: 10.1016/j.engappai.2014.02.008.
- [261] Z. J. Viharos and K. B. Kis, “Survey on Neuro-Fuzzy systems and their applications in technical diagnostics and measurement,” *Meas. J. Int. Meas. Confed.*, vol. 67, pp. 126–136, 2015, doi: 10.1016/j.measurement.2015.02.001.
- [262] C. F. Hsu, C. M. Lin, and R. G. Yeh, “Supervisory adaptive dynamic RBF-based neural-fuzzy control system design for unknown nonlinear systems,” *Appl. Soft Comput. J.*, vol. 13, no. 4, pp. 1620–1626, 2013, doi: 10.1016/j.asoc.2012.12.028.
- [263] K. Y. Lian, C. H. Su, and C. S. Huang, “Performance enhancement for T-S fuzzy control using neural networks,” *IEEE Trans. Fuzzy Syst.*, vol. 14, no. 5, pp. 619–627, 2006, doi: 10.1109/TFUZZ.2006.876728.
- [264] Y. H. Chang and W. S. Chan, “Adaptive dynamic surface control for uncertain nonlinear

- systems with interval type-2 fuzzy neural networks,” *IEEE Trans. Cybern.*, vol. 44, no. 2, pp. 293–304, 2014, doi: 10.1109/TCYB.2013.2253548.
- [265] Y. Zhong, A. Dutta, C. Copot, C. M. Ionescu, and R. De Keyser, “Implementation of a fractional PD controller tuned by genetic algorithm for a Steward platform,” *2013 9th Asian Control Conf. ASCC 2013*, 2013, doi: 10.1109/ASCC.2013.6606245.
- [266] H. Gözde, “Comparative Analysis of Evolutionary Computation Based Gain Scheduling Control for Ball and Plate Stabilization System,” *Balk. J. Electr. Comput. Eng.*, vol. 7, no. 1, pp. 44–55, 2019, doi: 10.17694/bajece.466306.
- [267] D. H. Nguyen and T. H. Huynh, “A SFLA-based fuzzy controller for balancing a ball and beam system,” *2008 10th Int. Conf. Control. Autom. Robot. Vision, ICARCV 2008*, no. December, pp. 1948–1953, 2008, doi: 10.1109/ICARCV.2008.4795828.
- [268] J. Fan and M. Han, “Nonlinear model predictive control of ball-plate system based on Gaussian particle swarm optimization,” *2012 IEEE Congr. Evol. Comput. CEC 2012*, no. 61074096, pp. 10–15, 2012, doi: 10.1109/CEC.2012.6252950.
- [269] K. Han, Y. Tian, Y. Kong, J. Li, and Y. Zhang, “Tracking control of ball and plate system using an improved PSO on-line training PID neural network,” *2012 IEEE Int. Conf. Mechatronics Autom. ICMA 2012*, pp. 2297–2302, 2012, doi: 10.1109/ICMA.2012.6285702.
- [270] X. Dong, Z. Zhang, and C. Chen, “Applying genetic algorithm to on-line updated PID neural network controllers for ball and plate system,” *2009 4th Int. Conf. Innov. Comput. Inf. Control. ICICIC 2009*, pp. 751–755, 2009, doi: 10.1109/ICICIC.2009.113.
- [271] X. Fan, N. Zhang, and S. Teng, “Trajectory planning and tracking of ball and plate system using hierarchical fuzzy control scheme,” vol. 144, no. 60174015, pp. 297–312, 2003, doi: 10.1016/S0165-0114(03)00135-0.
- [272] Y. H. Chang, C. W. Chang, C. W. Tao, H. W. Lin, and J. S. Taur, “Fuzzy sliding-mode control for ball and beam system with fuzzy ant colony optimization,” *Expert Syst. Appl.*, vol. 39, no. 3, pp. 3624–3633, 2012, doi: 10.1016/j.eswa.2011.09.052.
- [273] M. J. Mahmoodabadi, R. A. Maafi, S. E. Haghghi, and A. Moradi, “Pareto design of decoupled fuzzy sliding mode controller for nonlinear and underactuated systems using a hybrid optimization algorithm,” *SAIEE Africa Res. J.*, vol. 111, no. 1, pp. 4–21, 2020, doi: 10.23919/saiee.2020.9007881.
- [274] J. Aspnes, “Notes on Randomized Algorithms,” 2020.
- [275] R. M. Karp, “An introduction to randomized algorithms,” *Discret. Appl. Math.*, vol. 34, no. 1–3, pp. 165–201, 1991, doi: 10.1016/0166-218X(91)90086-C.

- [276] J. C. Spall, “A one-measurement form of simultaneous perturbation stochastic approximation,” *Automatica*, vol. 33, no. 1, pp. 109–112, Jan. 1997, doi: 10.1016/S0005-1098(96)00149-5.
- [277] J. Zhu, L. Wang, and J. C. Spall, “Efficient Implementation of Second-Order Stochastic Approximation Algorithms in High-Dimensional Problems,” *IEEE Trans. Neural Networks Learn. Syst.*, pp. 1–13, 2019, doi: 10.1109/tnnls.2019.2935455.
- [278] J. C. Spall, “Adaptive stochastic approximation by the simultaneous perturbation method,” *IEEE Trans. Automat. Contr.*, vol. 45, no. 10, pp. 1839–1853, 2000, doi: 10.1109/TAC.2000.880982.
- [279] J. C. Spall, “Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation,” *IEEE Trans. Automat. Contr.*, vol. 37, no. 3, pp. 332–341, 1992, doi: 10.1109/9.119632.
- [280] T. L. Lai, “Stochastic approximation,” *Ann. Stat.*, vol. 31, no. 2, pp. 391–406, 2003, doi: 10.1214/aos/1051027873.
- [281] R. F. Liddle and J. F. Monahan, “A stationary stochastic approximation method,” *J. Econom.*, vol. 38, no. 1–2, pp. 91–102, May 1988, doi: 10.1016/0304-4076(88)90028-0.
- [282] L. Lukšan and E. Spedicato, “Variable metric methods for unconstrained optimization and nonlinear least squares,” *J. Comput. Appl. Math.*, vol. 124, no. 1–2, pp. 61–95, Dec. 2000, doi: 10.1016/S0377-0427(00)00420-9.
- [283] M. Olivares and P. Albertos, “On the linear control of underactuated systems: The flywheel inverted pendulum,” in *2013 10th IEEE International Conference on Control and Automation (ICCA)*, Jun. 2013, pp. 27–32, doi: 10.1109/ICCA.2013.6564905.
- [284] B. de Jager, “Control of Under actuated Systems: A Monocycle Example,” *IFAC Proc. Vol.*, vol. 31, no. 27, pp. 61–66, 1998, doi: 10.1016/s1474-6670(17)40006-1.
- [285] C. Aguilar-Avelar and J. Moreno-Valenzuela, “A composite controller for trajectory tracking applied to the Furuta pendulum,” *ISA Trans.*, vol. 57, pp. 286–294, Jul. 2015, doi: 10.1016/j.isatra.2015.02.009.
- [286] M. F. Hamza, H. J. Yap, I. A. Choudhury, A. I. Isa, A. Y. Zimit, and T. Kumbasar, “Current development on using Rotary Inverted Pendulum as a benchmark for testing linear and nonlinear control algorithms,” *Mech. Syst. Signal Process.*, vol. 116, pp. 347–369, 2019, doi: 10.1016/j.ymsp.2018.06.054.
- [287] M. M. Morato, J. E. Normey-Rico, and O. Sename, “Model predictive control design for linear parameter varying systems: A survey,” *Annu. Rev. Control*, no. xxxx, 2020,

- doi: 10.1016/j.arcontrol.2020.04.016.
- [288] Y. Tian, M. Bai, and J. Su, “A non-linear switching controller for ball and plate system,” *Int. J. Model. Identif. Control. Vol. 1, No. 3*, vol. 1, no. 3, 2006, doi: 10.1504/IJMIC.2006.011940.
- [289] C. C. Ker, C. E. Lin, and R. T. Wang, “Tracking and balance control of ball and plate system,” *J. Chinese Inst. Eng. Trans. Chinese Inst. Eng. A/Chung-kuo K. Ch’eng Hsueh K’an*, vol. 30, no. 3, pp. 459–470, 2007, doi: 10.1080/02533839.2007.9671274.
- [290] Y. Wang, M. Sun, Z. Wang, Z. Liu, and Z. Chen, “A novel disturbance-observer based friction compensation scheme for ball and plate system,” *ISA Trans.*, vol. 53, no. 2, pp. 671–678, 2014, doi: 10.1016/j.isatra.2013.11.011.
- [291] Z. Dong and J. Zhang, “An active disturbance rejection controller design for ball and plate system,” *Chinese Control Conf. CCC*, vol. 2015-Septe, pp. 928–933, 2015, doi: 10.1109/ChiCC.2015.7259759.
- [292] H. I. Ali, H. M. Jassim, and A. F. Hasan, “Optimal Nonlinear Model Reference Controller Design for Ball and Plate System,” *Arab. J. Sci. Eng.*, vol. 44, no. 8, pp. 6757–6768, Aug. 2019, doi: 10.1007/s13369-018-3616-1.
- [293] M. T. Ho, Y. Rizal, and L. M. Chu, “Visual servoing tracking control of a ball and plate system: Design, implementation and experimental validation,” *Int. J. Adv. Robot. Syst.*, vol. 10, 2013, doi: 10.5772/56525.
- [294] E. Fabregas, J. Chacón, S. Dormido-Canto, G. Farias, and S. Dormido, “Virtual Laboratory of the Ball and Plate System,” *IFAC-PapersOnLine*, vol. 48, no. 29, pp. 152–157, 2015, doi: 10.1016/j.ifacol.2015.11.229.
- [295] R. C. Hoover and J. St. Amand, “Design of an Approximate Control Law Using Input-State Linearization for the Ball on a Plate System,” *Vol. 4 Dyn. Control Uncertainty, Parts A B*, p. 203, 2012, doi: 10.1115/IMECE2012-88418.
- [296] L. I. U. Zhixiang, Z. Youmin, Y. Chi, and L. U. O. Jun, “An Adaptive Linear Parameter Varying Fault Tolerant Control Scheme for Unmanned Surface Vehicle Steering Control,” pp. 6197–6202, 2015.
- [297] Y. Wang, Q. Han, and S. Member, “Network-Based Fault Detection Filter and Controller Coordinated Design for Unmanned Surface Vehicles in Network Environments,” vol. 3203, no. c, pp. 1–13, 2016, doi: 10.1109/TII.2016.2526648.
- [298] E. Uney, “Parameter Identification Based Fault Tolerant Control Against Actuator Failures Applied to UUV Dynamics,” 2014.
- [299] X. Wang, Z. Fei, H. Gao, and J. Yu, “Integral-Based Event-Triggered Fault Detection

- Filter Design for Unmanned Surface Vehicles,” *IEEE Trans. Ind. Informatics*, vol. PP, no. c, p. 1, 2019, doi: 10.1109/TII.2019.2900230.
- [300] S. Campbell, W. Naeem, and G. W. Irwin, “Annual Reviews in Control A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance manoeuvres,” *Annu. Rev. Control*, vol. 36, no. 2, pp. 267–283, 2012, doi: 10.1016/j.arcontrol.2012.09.008.
- [301] E. H. Mamdani and S. Assilian, “An experiment in linguistic synthesis with a fuzzy logic controller,” *Int. J. Man. Mach. Stud.*, vol. 7, no. 1, pp. 1–13, Jan. 1975, doi: 10.1016/S0020-7373(75)80002-2.
- [302] T. Takagi and M. Sugeno, “Fuzzy identification of systems and its applications to modeling and control,” *IEEE Trans. Syst. Man. Cybern.*, vol. SMC-15, no. 1, pp. 116–132, Jan. 1985, doi: 10.1109/TSMC.1985.6313399.
- [303] J.-S. R. Jang, “Input selection for ANFIS learning,” in *Proceedings of IEEE 5th International Fuzzy Systems*, vol. 2, pp. 1493–1499, doi: 10.1109/FUZZY.1996.552396.
- [304] “Neuro-Adaptive Learning and ANFIS - MATLAB & Simulink - MathWorks India.” <https://in.mathworks.com/help/fuzzy/neuro-adaptive-learning-and-anfis.html> (accessed Jun. 20, 2020).
- [305] A. Tahir, J. Böling, M.-H. Haghbayan, H. T. Toivonen, and J. Plosila, “Swarms of Unmanned Aerial Vehicles — A Survey,” *J. Ind. Inf. Integr.*, vol. 16, p. 100106, Dec. 2019, doi: 10.1016/j.jii.2019.100106.
- [306] B. K. Patle, G. Babu L, A. Pandey, D. R. K. Parhi, and A. Jagadeesh, “A review: On path planning strategies for navigation of mobile robot,” *Def. Technol.*, vol. 15, no. 4, pp. 582–606, Aug. 2019, doi: 10.1016/j.dt.2019.04.011.
- [307] M. Arbulu, A. Padilla, and F. Ramirez, “Geometric balancing control of humanoid robots,” in *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dec. 2013, pp. 2136–2141, doi: 10.1109/ROBIO.2013.6739785.
- [308] B.-J. Lee, Y.-D. Kim, and J.-H. Kim, “BALANCE CONTROL OF HUMANOID ROBOT FOR HUROSOT,” *IFAC Proc. Vol.*, vol. 38, no. 1, pp. 215–220, 2005, doi: 10.3182/20050703-6-CZ-1902.02088.
- [309] Y. Wang, Q. Zhu, R. Xiong, and J. Chu, “Standing Balance Control for Position Control-Based Humanoid Robot,” *IFAC Proc. Vol.*, vol. 46, no. 20, pp. 429–436, 2013, doi: 10.3182/20130902-3-CN-3020.00064.
- [310] M. Shahriari-Kahkeshi, “Dead-Zone Model-Based Adaptive Fuzzy Wavelet Control for Nonlinear Systems Including Input Saturation and Dynamic Uncertainties,” *Int. J. Fuzzy*

- Syst.*, vol. 20, no. 8, pp. 2577–2592, Dec. 2018, doi: 10.1007/s40815-018-0515-2.
- [311] T.-T. Huynh, T.-L. Le, and C.-M. Lin, “Self-Organizing Recurrent Wavelet Fuzzy Neural Network-Based Control System Design for MIMO Uncertain Nonlinear Systems Using TOPSIS Method,” *Int. J. Fuzzy Syst.*, vol. 21, no. 2, pp. 468–487, Mar. 2019, doi: 10.1007/s40815-018-0550-z.
- [312] B. E. F. Tafti, M. Teshnehab, and M. A. Khanesar, “Recurrent Interval Type-2 Fuzzy Wavelet Neural Network with Stable Learning Algorithm: Application to Model-Based Predictive Control,” *Int. J. Fuzzy Syst.*, vol. 22, no. 2, pp. 351–367, Mar. 2020, doi: 10.1007/s40815-019-00766-z.
- [313] T. K. Sarkar, R. Adve, M. Salazar, and L. Garcia, “Engineering Perspective, Part 1: Discrete Wavelet Techniques’,” *IEEE Antennas Propag. Mag.*, vol. 40, no. 5, pp. 49–70, 1998, doi: 10.1109/74.735965.
- [314] R. Yan and R. X. Gao, “Tutorial 21 wavelet transform: A mathematical tool for non-stationary signal processing in measurement science part 2 in a series of tutorials in instrumentation and measurement,” *IEEE Instrum. Meas. Mag.*, vol. 12, no. 5, pp. 35–44, 2009, doi: 10.1109/MIM.2009.5270529.
- [315] V. F. Poterasu, “Wavelets transform for nonlinear control of multibody systems,” *J. Franklin Inst.*, vol. 338, no. 2–3, pp. 321–334, 2001, doi: 10.1016/S0016-0032(00)00090-9.
- [316] S. G. Mallat, “A Theory for Multiresolution Signal Decomposition: The Wavelet Representation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 7, pp. 674–693, 1989, doi: 10.1109/34.192463.
- [317] W. K. Ngui, M. S. Leong, L. M. Hee, and A. M. Abdelrhman, “Wavelet Analysis: Mother Wavelet Selection Methods,” *Appl. Mech. Mater.*, vol. 393, pp. 953–958, Sep. 2013, doi: 10.4028/www.scientific.net/AMM.393.953.
- [318] T. S. A. A. Sewilam, “Automated wavelet-based fault detection and diagnosis for smart distribution systems and microgrids,” UOIT, 2017.
- [319] W. G. Morsi and M. E. El-Hawary, “A new perspective for the IEEE standard 1459-2000 via stationary wavelet transform in the presence of nonstationary power quality disturbance,” *IEEE Trans. Power Deliv.*, vol. 23, no. 4, pp. 2356–2365, 2008, doi: 10.1109/TPWRD.2008.2002660.
- [320] J. G. Proakis and D. G. Manolakis, “Design of Hilbert Transformers,” *Digital Signal Processing*, pp. 657–662, 1996.
- [321] M. Vetterli and C. Herley, “Wavelets and Filter Banks: Theory and Design,” *IEEE*

- Trans. Signal Process.*, vol. 40, no. 9, pp. 2207–2232, 1992, doi: 10.1109/78.157221.
- [322] L. Debnath, “The Wigner-Ville Distribution and Time-Frequency Signal Analysis,” in *Wavelet Transforms and Their Applications*, Boston, MA: Birkhäuser Boston, 2002, pp. 307–360.
- [323] Y.-L. You, *Audio Coding: Theory and Applications*. 2010.
- [324] S. Parvez and Z. Gao, “A wavelet-based multiresolution PID controller,” *IEEE Trans. Ind. Appl.*, vol. 41, no. 2, pp. 537–543, 2005, doi: 10.1109/TIA.2005.844378.
- [325] J. L. Febin Daya, V. Subbiah, A. Iqbal, and S. Padmanaban, “Novel wavelet-fuzzy based indirect field oriented control of induction motor drives,” *J. Power Electron.*, vol. 13, no. 4, pp. 656–668, 2013, doi: 10.6113/JPE.2013.13.4.656.
- [326] J. L. Febin Daya, V. Subbiah, and P. Sanjeevikumar, “Robust Speed Control of an Induction Motor Drive using Wavelet-fuzzy based Self-tuning Multiresolution Controller,” *Int. J. Comput. Intell. Syst.*, vol. 6, no. 4, pp. 724–738, 2013, doi: 10.1080/18756891.2013.803741.
- [327] D. Wu, C.-T. Lin, J. Huang, and Z. Zeng, “On the Functional Equivalence of TSK Fuzzy Systems to Neural Networks, Mixture of Experts, CART, and Stacking Ensemble Regression,” *IEEE Trans. Fuzzy Syst.*, no. 3, pp. 1–1, 2019, doi: 10.1109/TFUZZ.2019.2941697.
- [328] G. Calcev, “A Passivity Result for Fuzzy Control Systems,” vol. 61, no. December, pp. 2727–2728, 1996, doi: 10.1109/CDC.1996.573518.
- [329] Y. C. Shin and Chengying Xu, “Stability Analysis Method,” in *Automation and control engineering*, 1st ed., United States, 2008, pp. 225–235.
- [330] M. Biglarbegian, W. W. Melek, and J. M. Mendel, “On the Stability of Interval Type-2 TSK Fuzzy Logic Control Systems,” *IEEE Trans. Syst. Man, Cybern. Part B*, vol. 40, no. 3, pp. 798–818, Jun. 2010, doi: 10.1109/TSMCB.2009.2029986.
- [331] C.-J. Lin and C.-H. Chen, “A SELF-ORGANIZING QUANTUM NEURAL FUZZY NETWORK AND ITS APPLICATIONS,” *Cybern. Syst.*, vol. 37, no. 8, pp. 839–859, Dec. 2006, doi: 10.1080/01969720600942742.
- [332] A. Mencattini, M. Salmeri, S. Bertazzoni, R. Lojacono, E. Pasero, and W. Moniaci, “Short Term Local Meteorological Forecasting Using Type-2 Fuzzy Systems,” 2006, pp. 95–104.
- [333] U. S. Tiwary, “Fuzzy Sets (Type-1 and Type-2) and their Applications,” 2009. [Online]. Available: <http://www.iitk.ac.in/ee/courses/archives/2013/intel-info/d1pdf3.pdf>.
- [334] W.-W. Tan and D. Wu, “Design of Type-Reduction Strategies for Type-2 Fuzzy Logic

- Systems using Genetic Algorithms,” 2007, pp. 169–187.
- [335] M. Pratama, E. Lughofer, M. J. Er, W. Rahayu, and T. Dillon, “Evolving type-2 recurrent fuzzy neural network,” in *2016 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2016, pp. 1841–1848, doi: 10.1109/IJCNN.2016.7727423.
- [336] R. Chellappa *et al.*, “Gaussian Mixture Models,” *Encycl. Biometrics*, no. 2, pp. 659–663, 2009, doi: 10.1007/978-0-387-73003-5\_196.
- [337] L. E. Brito da Silva, I. Elnabarawy, and D. C. Wunsch, “A survey of adaptive resonance theory neural network models for engineering applications,” *Neural Networks*, vol. 120, pp. 167–203, Dec. 2019, doi: 10.1016/j.neunet.2019.09.012.
- [338] M. Kordestani, M. Dehghani, B. Moshiri, and M. Saif, “A New Fusion Estimation Method for Multi-Rate Multi-Sensor Systems With Missing Measurements,” *IEEE Access*, vol. 8, pp. 47522–47532, 2020, doi: 10.1109/ACCESS.2020.2979222.
- [339] K. J. Waldron, S.-L. Wang, and S. J. Bolin, “A Study of the Jacobian Matrix of Serial Manipulators,” *J. Mech. Transm. Autom. Des.*, vol. 107, no. 2, pp. 230–237, Jun. 1985, doi: 10.1115/1.3258714.
- [340] E. Sanchez and A. Alanis, “Adaptive Approximation Based Control: Unifying Neural, Fuzzy and Traditional Adaptive Approximation Approaches (Farrell, J.A. and Polycarpou, M.M. [Book review],” *IEEE Trans. Neural Networks*, vol. 19, no. 4, pp. 731–732, Apr. 2008, doi: 10.1109/TNN.2008.921577.
- [341] E. Kayacan, E. Kayacan, and M. Ahmadiéh Khanesar, “Identification of Nonlinear Dynamic Systems Using Type-2 Fuzzy Neural Networks—A Novel Learning Algorithm and a Comparative Study,” *IEEE Trans. Ind. Electron.*, vol. 62, no. 3, pp. 1716–1724, Mar. 2015, doi: 10.1109/TIE.2014.2345353.
- [342] E. Kayacan and O. Kaynak, “Sliding mode control theory-based algorithm for online learning in type-2 fuzzy neural networks: application to velocity control of an electro hydraulic servo system,” *Int. J. Adapt. Control Signal Process.*, vol. 26, no. 7, pp. 645–659, Jul. 2012, doi: 10.1002/acs.1292.
- [343] J. S. Arora, “Optimum Design Problem Formulation,” in *Introduction to Optimum Design*, Elsevier, 2017, pp. 19–70.
- [344] J.-B. Hu, G.-P. Lu, S.-B. Zhang, and L.-D. Zhao, “Lyapunov stability theorem about fractional system without and with delay,” *Commun. Nonlinear Sci. Numer. Simul.*, vol. 20, no. 3, pp. 905–913, Mar. 2015, doi: 10.1016/j.cnsns.2014.05.013.
- [345] G. C. Calafiore, F. Dabbene, and R. Tempo, “Randomized Algorithms for Probabilistic Robustness with Real and Complex Structured Uncertainty,” vol. 45, no. 12, pp. 2218–



- 2235, 2000.
- [346] G. Calafiore, F. Dabbene, and R. Tempo, “A survey of randomized algorithms for control synthesis and performance verification,” vol. 23, pp. 301–316, 2007, doi: 10.1016/j.jco.2007.01.003.
- [347] G. C. Calafiore, F. Dabbene, and R. Tempo, “Research on probabilistic methods for control system design,” *Automatica*, vol. 47, no. 7, pp. 1279–1293, 2011, doi: 10.1016/j.automatica.2011.02.029.
- [348] L. Asselborn, D. Groß, and O. Stursberg, *Control of uncertain nonlinear systems using ellipsoidal reachability calculus*, vol. 46, no. 23 PART 1. IFAC, 2013.
- [349] A. A. Kurzhanskiy and P. Varaiya, “Ellipsoidal techniques for reachability analysis of discrete-time linear systems,” *IEEE Trans. Automat. Contr.*, vol. 52, no. 1, pp. 26–38, 2007, doi: 10.1109/TAC.2006.887900.
- [350] N. Ramdani, N. Meslem, and Y. Candau, *Reachability of uncertain nonlinear systems using a nonlinear*, vol. 4981 LNCS, no. 2. IFAC, 2008.
- [351] A. Kurzhanski and I. Vályi, *Ellipsoidal calculus for estimation and control*. The International Institute for Applied Systems Analysis, 1997.
- [352] Y. Fujisaki and Y. Oishi, “Guaranteed cost regulator design: A probabilistic solution and a randomized algorithm,” *Automatica*, vol. 43, no. 2, pp. 317–324, 2007, doi: 10.1016/j.automatica.2006.08.020.
- [353] J. C. Spall, “Adaptive stochastic approximation by the simultaneous perturbation method,” in *Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No.98CH36171)*, vol. 4, pp. 3872–3879, doi: 10.1109/CDC.1998.761833.
- [354] J. C. Spall, “Multivariate stochastic approximation using a simultaneous perturbation gradient approximation,” *IEEE Trans. Automat. Contr.*, vol. 37, no. 3, pp. 332–341, Mar. 1992, doi: 10.1109/9.119632.
- [355] T. L. Lai, “Stochastic approximation: invited paper,” *Ann. Stat.*, vol. 31, no. 2, pp. 391–406, Apr. 2003, doi: 10.1214/aos/1051027873.
- [356] V. S. Borkar, *Stochastic Approximation*, vol. 48. Gurgaon: Hindustan Book Agency, 2008.
- [357] H. J. Kushner and G. G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*, vol. 35. New York: Springer-Verlag, 2003.
- [358] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, “Distributed Asynchronous Deterministic and Stochastic Gradient Optimization Algorithms,” in *1984 American*

- Control Conference*, Jul. 1984, pp. 484–489, doi: 10.23919/ACC.1984.4788427.
- [359] N. Amelina, A. Fradkov, Y. Jiang, and D. J. Vergados, “Approximate Consensus in Stochastic Networks with Application to Load Balancing,” *IEEE Trans. Inf. Theory*, vol. 61, no. 4, pp. 1739–1752, 2015, doi: 10.1109/TIT.2015.2406323.
- [360] A. D. Flaxman, A. T. Kalai, and H. B. McMahan, “Online convex optimization in the bandit setting: Gradient descent without a gradient,” *Proc. Annu. ACM-SIAM Symp. Discret. Algorithms*, pp. 385–394, 2005.
- [361] V. Malyshekin, Ed., *Parallel Computing Technologies*, vol. 4671. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.
- [362] J. C. Spall, “Overview of the simultaneous perturbation method for efficient optimization,” *JOHNS HOPKINS APL Tech. Dig.*, vol. 19, no. 4, pp. 482–492, 1999.
- [363] B. T. Polyak and A. B. Tsybakov, “Optimal rates of convergence for the global search stochastic optimization,” *Probl. Informa- tion Transm.*, vol. 26, no. 2, pp. 126–130, 1990.
- [364] H. F. Chen, T. E. Duncan, and B. Pasik-Duncan, “A Kiefer-Wolfowitz algorithm with randomized differences,” *IEEE Trans. Automat. Contr.*, vol. 44, no. 3, pp. 442–453, Mar. 1999, doi: 10.1109/9.751340.
- [365] A. T. Vakhitov, O. N. Granichin, and L. S. Gurevich, “Algorithm for stochastic approximation with trial input perturbation in the nonstationary problem of optimization,” *Autom. Remote Control*, vol. 70, no. 11, pp. 1827–1835, Nov. 2009, doi: 10.1134/S000511790911006X.
- [366] J. M. Soriano, “Global minimum point of a convex function,” *Appl. Math. Comput.*, vol. 55, no. 2–3, pp. 213–218, May 1993, doi: 10.1016/0096-3003(93)90022-7.
- [367] C. C. Pugh, “Lebesgue Theory,” 2002, pp. 363–429.
- [368] X. Mao and S. Sabanis, “Numerical solutions of stochastic differential delay equations under local Lipschitz condition,” *J. Comput. Appl. Math.*, vol. 151, no. 1, pp. 215–227, Feb. 2003, doi: 10.1016/S0377-0427(02)00750-1.
- [369] Y. Watanabe, I. Takami, and G. Chen, “Tracking control for 2DOF helicopter via robust LQ control with adaptive law,” *2012 2nd Aust. Control Conf. AUCC 2012*, no. November, pp. 399–404, 2012.
- [370] G. N. Roberts and R. Sutton, Eds., *Advances in Unmanned Marine Vehicles*. The Institution of Engineering and Technology, Michael Faraday House, Six Hills Way, Stevenage SG1 2AY, UK: IET, 2006.
- [371] P. Xie, R. Liang, and H. Zhang, “Fault Tolerant Control of Unmanned Aerial Vehicle

- Based on Adaptive Algorithm,” in *2019 International Conference on Smart Grid and Electrical Automation (ICSGEA)*, Aug. 2019, pp. 128–131, doi: 10.1109/ICSGEA.2019.00037.
- [372] V. S. Rao, V. I. George, S. Kamath, and Shreesha C., “Reliable H infinity observer-controller design for sensor and actuator failure in TRMS,” in *2014 International Conference on Advances in Electrical Engineering (ICAEE)*, Jan. 2014, pp. 1–5, doi: 10.1109/ICAEE.2014.6838536.
- [373] H. Yang, B. Jiang, H. H. T. Liu, H. Yang, and Q. Zhang, “Attitude Synchronization For Multiple 3-DOF Helicopters With Actuator Faults,” *IEEE/ASME Trans. Mechatronics*, vol. 24, no. 2, pp. 597–608, Apr. 2019, doi: 10.1109/TMECH.2019.2895222.
- [374] K. Yan, M. Chen, Q. Wu, and Y. Wang, “Adaptive flight control for unmanned autonomous helicopter with external disturbance and actuator fault,” *J. Eng.*, vol. 2019, no. 22, pp. 8359–8364, Nov. 2019, doi: 10.1049/joe.2019.1080.
- [375] H. Bouadi and F. Mora-Camino, “Direct Adaptive Backstepping Flight Control for Quadcopter Trajectory Tracking,” in *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, Sep. 2018, pp. 1–8, doi: 10.1109/DASC.2018.8569628.
- [376] S. Mallat, *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*. 2008.
- [377] A. Haque, V. S. B. Kurukuru, M. A. Khan, I. Khan, and Z. A. Jaffery, “Fault diagnosis of Photovoltaic Modules,” *Energy Sci. Eng.*, no. September, pp. 1–23, 2018, doi: 10.1002/ese3.255.
- [378] H. Fröhlich, J. K. Wegner, and A. Zell, “Towards Optimal Descriptor Subset Selection with Support Vector Machines in Classification and Regression,” *QSAR Comb. Sci.*, vol. 23, no. 5, pp. 311–318, Jul. 2004, doi: 10.1002/qsar.200410011.
- [379] H. Wang, G. Zhao, and N. Li, “Training support vector data descriptors using converging linear particle swarm optimization,” *Neural Comput. Appl.*, vol. 21, no. 6, pp. 1099–1105, Sep. 2012, doi: 10.1007/s00521-012-0872-y.
- [380] G. Strang, *Linear Algebra and Its Applications*, Second. Harcourt Brace Jovanovich College Publishers, 2010.
- [381] V. N. Vapnik, “An overview of statistical learning theory,” *IEEE Trans. Neural Networks*, vol. 10, no. 5, pp. 988–999, 1999, doi: 10.1109/72.788640.
- [382] A. J. Smola, B. Schölkopf, and K.-R. Müller, “The connection between regularization operators and support vector kernels,” *Neural Networks*, vol. 11, no. 4, pp. 637–649, Jun. 1998, doi: 10.1016/S0893-6080(98)00032-X.
- [383] W. Wang, Z. Xu, W. Lu, and X. Zhang, “Determination of the spread parameter in the

- Gaussian kernel for classification and regression,” *Neurocomputing*, vol. 55, no. 3–4, pp. 643–663, 2003, doi: 10.1016/S0925-2312(02)00632-X.
- [384] Liuping Wang, “PID Control of Multi-rotor Unmanned Aerial Vehicles,” in *PID Control System Design and Automatic Tuning using MATLAB/Simulink*, Wiley, 2020, pp. 305–326.
- [385] D. P. Morgan and C. L. Scofield, “Signal Processing and Feature Extraction,” in *Neural Networks and Speech Processing*, Boston, MA: Springer US, 1991, pp. 163–201.
- [386] I. Daubechies, “The wavelet transform, time-frequency localization and signal analysis,” *Inf. Theory, IEEE Trans.*, vol. 36, no. 5, pp. 961–1005, 1990, doi: 10.1109/18.57199.
- [387] Han Qiu Zhang and Yong Yan, “A wavelet-based approach to abrupt fault detection and diagnosis of sensors,” *IEEE Trans. Instrum. Meas.*, vol. 50, no. 5, pp. 1389–1396, 2001, doi: 10.1109/19.963215.
- [388] J.-S. R. Jang, “ANFIS: adaptive-network-based fuzzy inference system,” *IEEE Trans. Syst. Man. Cybern.*, vol. 23, no. 3, pp. 665–685, 1993, doi: 10.1109/21.256541.
- [389] J.-G. Juang, W.-K. Liu, and R.-W. Lin, “A hybrid intelligent controller for a twin rotor MIMO system and its hardware implementation,” *ISA Trans.*, vol. 50, no. 4, pp. 609–619, Oct. 2011, doi: 10.1016/j.isatra.2011.06.006.
- [390] A. Qi, Q. Han, and Y. Wang, “A Survey of Motion Control for Marine Vehicles,” pp. 4214–4218, 2015.
- [391] T. Xu, J. Chudley, and R. Sutton, “A FAULT TOLERANT MULTI-SENSOR NAVIGATION SYSTEM FOR AN UNMANNED SURFACE VEHICLE T,” pp. 1503–1508.
- [392] S. M. Zanolli, G. Astolfi, G. Bruzzone, M. Bibuli, and M. Caccia, *Application of Fault Detection and Isolation Techniques on an Unmanned Surface Vehicle (USV)*, vol. 45, no. 27. IFAC, 2012.
- [393] C. Cortes and V. Vapnik, “Support-Vector Networks,” *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995, doi: 10.1023/A:1022627411411.
- [394] J. Mercer, “Functions of Positive and Negative Type, and their Connection with the Theory of Integral Equations,” *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.*, vol. 209, no. 441–458, pp. 415–446, 2006, doi: 10.1098/rsta.1909.0016.
- [395] O. Chapelle, “Training a Support Vector Machine in the Primal,” *Neural Comput.*, vol. 19, no. 5, pp. 1155–1178, 2007, doi: 10.1162/neco.2007.19.5.1155.
- [396] D. Zhang and J. J. P. Tsai, *Machine Learning Applications in Software Engineering*, vol.

16. WORLD SCIENTIFIC, 2005.
- [397] Y.-H. Wang, T.-H. S. Li, and C.-J. Lin, “Backward Q-learning: The combination of Sarsa algorithm and Q-learning,” *Eng. Appl. Artif. Intell.*, vol. 26, no. 9, pp. 2184–2193, Oct. 2013, doi: 10.1016/j.engappai.2013.06.016.
- [398] G. F. Franklin, M. L. Workman, and D. J. Powell, *Digital Control of Dynamic Systems*. Addison-Wesley Longman Publishing Co., Inc.,
- [399] F. L. Lewis, Draguna Vrabié, and K. G. Vamvoudakis, “Reinforcement Learning and Feedback Control: Using Natural Decision Methods to Design Optimal Adaptive Controllers,” *IEEE Control Syst.*, vol. 32, no. 6, pp. 76–105, Dec. 2012, doi: 10.1109/MCS.2012.2214134.
- [400] S. A. A. Rizvi and Z. Lin, “An iterative Q-learning scheme for the global stabilization of discrete-time linear systems subject to actuator saturation,” *Int. J. Robust Nonlinear Control*, vol. 29, no. 9, pp. 2660–2672, Jun. 2019, doi: 10.1002/rnc.4514.
- [401] B. Kiumarsi, F. L. Lewis, H. Modares, A. Karimpour, and M.-B. Naghibi-Sistani, “Reinforcement -learning for optimal tracking control of linear discrete-time systems with unknown dynamics,” *Automatica*, vol. 50, no. 4, pp. 1167–1175, Apr. 2014, doi: 10.1016/j.automatica.2014.02.015.
- [402] D. P. Bertsekas, “Value and Policy Iterations in Optimal Control and Adaptive Dynamic Programming,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 28, no. 3, pp. 500–509, Mar. 2017, doi: 10.1109/TNNLS.2015.2503980.
- [403] J. Li, T. Chai, F. L. Lewis, J. Fan, Z. Ding, and J. Ding, “Off-Policy Q-Learning: Set-Point Design for Optimizing Dual-Rate Rougher Flotation Operational Processes,” *IEEE Trans. Ind. Electron.*, vol. 65, no. 5, pp. 4092–4102, May 2018, doi: 10.1109/TIE.2017.2760245.
- [404] J. Li, T. Chai, F. L. Lewis, Z. Ding, and Y. Jiang, “Off-Policy Interleaved \$Q\$ - Learning: Optimal Control for Affine Nonlinear Discrete-Time Systems,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 30, no. 5, pp. 1308–1320, May 2019, doi: 10.1109/TNNLS.2018.2861945.
- [405] Y. Jiang, J. Fan, T. Chai, J. Li, and F. L. Lewis, “Data-Driven Flotation Industrial Process Operational Optimal Control Based on Reinforcement Learning,” *IEEE Trans. Ind. Informatics*, vol. 14, no. 5, pp. 1974–1989, May 2018, doi: 10.1109/TII.2017.2761852.
- [406] D. P. Bertsekas, “Value and Policy Iterations in Optimal Control and Adaptive Dynamic Programming,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 28, no. 3, pp. 500–509,

- 2017, doi: 10.1109/TNNLS.2015.2503980.
- [407] L. Busoniu, D. Ernst, B. De Schutter, and R. Babuska, “Online least-squares policy iteration for reinforcement learning control,” in *Proceedings of the 2010 American Control Conference*, Jun. 2010, pp. 486–491, doi: 10.1109/ACC.2010.5530856.
- [408] M. G. Lagoudakis, R. Parr, and M. L. Littman, “Least-Squares Methods in Reinforcement Learning for Control,” 2002, pp. 249–260.
- [409] A. V. Oppenheim and G. C. Verghese, “Estimation with Minimum Mean Square Error,” in *Introduction to Communication, Control, and Signal Processing*, MIT OpenCourseWare, 2010.
- [410] E. Mizutani and S. E. Dreyfus, “Totally model-free reinforcement learning by actor-critic Elman networks in non-Markovian domains,” in *1998 IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98CH36227)*, vol. 3, pp. 2016–2021, doi: 10.1109/IJCNN.1998.687169.
- [411] D. Wang, D. Liu, Q. Wei, D. Zhao, and N. Jin, “Optimal control of unknown nonaffine nonlinear discrete-time systems based on adaptive dynamic programming,” *Automatica*, vol. 48, no. 8, pp. 1825–1832, Aug. 2012, doi: 10.1016/j.automatica.2012.05.049.
- [412] C. J. Gatti and M. J. Embrechts, “Reinforcement Learning with Neural Networks: Tricks of the Trade,” 2013, pp. 275–310.

# CURRICULUM VITAE

## RUPAM SINGH

---

### **Education:**

- (2016 - Pursuing) Ph.D. candidate in Control Systems from Delhi Technological University, New Delhi.  
Research Topic: Intelligent control of ball balancer and helicopter system  
Supervisor: Prof. Dr. Bharat Bhushan
- (2014 - 2016) Master of Technology (M.Tech) in Control Systems from Amity School of Engineering and Technology, Amity University, Uttar Pradesh in “First” division.
- (2009 - 2013) Bachelor of Technology (B.Tech) in Electrical and Electronics Engineering from Hindustan College of Science and Technology, Agra, affiliated to Gautam Buddha Technical University, Greater Noida, Uttar Pradesh in “First” division.
- (2007 - 2009) Senior Secondary School Certificate from S. G. Inter college, Kasganj, of Uttar Pradesh Board in “First” division.
- (2006 – 2007) Secondary School Certificate from D. D. J. S. B. Inter College, Kasganj, of Uttar Pradesh Board in “First” division.

### **Teaching Experience:**

I have been a Teaching Assistant at Department of Electrical Engineering, Delhi Technological University, New Delhi and have assisted faculty with following technical laboratory.

- Advance control system laboratory
- Instrumentation and measurement laboratory
- Control system laboratory
- Basic electrical laboratory
- Process control laboratory
- Engineering analysis and assign laboratory

### **Research Areas:**

Robotic systems, Unmanned Vehicles, Optimal Control Theory, Intelligent Control Techniques, Machine Learning, Fault Diagnosis.

### **Technical Skill:**

MATLAB & Simulink, LabVIEW, Supervisory control and data acquisition- Human machine interface, Quarc real-time control software, Hardware in Loop-Application programming interface.

### **Technical Training:**

- Training Program on “*Two Week Training Programme on Rooftop Solar Grid Engineers*” organized by National Institute of Solar Energy, MNRE in Collaboration with Jamia Millia Islamia (A Central University), New Delhi w.e.f. 11th Feb 2019 to 21 Feb 2019.
- Short term course on “*Advanced Optimization Techniques for Engineering Application*” organized by Department of Computer Engineering, National Institute of Technology, Kurukshetra during 13<sup>th</sup> Jan 2018 to 17<sup>th</sup> Jan 2018.
- GAIN Course (Ministry of Human Resource Development, Govt. of India) on “*Emerging Cutting-Edge Technologies in Advanced Electrical Machines and Drives: Design & Performance Issues, Fault Diagnosis, Failure Prognosis and Mitigation*” organized by Department of Electrical Engineering, Delhi Technological University during 6th Nov 2017 to 10th Dec 2017.
- GAIN Course (Ministry of Human Resource Development, Govt. of India) on “*Recent Trends in Power System Reliability Evaluation: Models, Statistical Methods*” Organized by Department of Electrical Engineering, Delhi Technological University during 9th Oct 2017 to 13th Oct 2017.
- M.Tech final Project on “*PID control for one-dimensional prototype of Cubli*” at Indian Institute of Technology, Mandi from Jan- May 2016.
- Summer Training at Prolific System & Technologies Pvt. Ltd., Noida was done during May-July 2015. Project work was titled as “*Automatic Car Parking System with the help of SCADA*”.
- Summer Training at Sofcon India Pvt. Ltd., Noida was done during June -July 2012.



### **Dissertation Topics:**

- M.Tech Dissertation topic, “*PID control for one-dimensional prototype of Cubli*”.  
The research is focussed on the evolution of a cube shaped device called the Cubli, which has the ability to leap up, and create equilibrium on different angles and edges. The dissertation develops control techniques for the sudden braking on high angular velocity rotation of reaction wheels, which are seated on the surface of Cubli. This phenomenon allowed the Cubli to leap up, first to its edge, and then to its corner without the need for any extrinsic control. Further, to achieve an effective controlling and balancing position with the Cubli, a Model Predictive Control was implemented and the future states of the system are analysed.
- B. Tech Dissertation topic, “*Three Phase Appliances with Protector Delay*”.  
The research is focussed at developing an automatic cut-off operation for an electrical appliance if any phase failure occurs in the three-phase supply. This is achieved by developing a prospective protector which maintains the efficiency of the appliance operating with the 3- phase supply. The outcome of the research ensured that the protector circuit capable of protecting the three-phase appliances from failure of any of the phases by disconnecting the power supply through the contactor and automatically restores the three-phase supply to the appliance with reasonable time delay when all the phases are available.

### **Membership of Academic Societies:**

- IEEE Student Member
- IEEE Control System Society
- IEEE Young Professionals Member

### **Reviewer of Scientific Journals:**

- IEEE Journal of Emerging and Selected Topics in Industrial Electronics
- IEEE Transactions on Neural Networks and Learning Systems
- International Journal of Adaptive control and Signal Processing (Wiley)

### **Achievements:**

- Member in Committee for IEEE International Conference on Power Electronics, Intelligent Control & Energy Systems which was held in Delhi Technological University on 22-24<sup>th</sup> October 2018.
- Successfully completed Level III course in German during 2014-16 at Amity University, Noida (NCR).

### **Extra-Curricular activities:**

- Attended '*Digital Shakti*' workshop on '*Digital Literacy and Online Safety*' conducted at Delhi Technological University on 9<sup>th</sup> April 2019
- Attended the workshop titled '*E-Resources: A Gateway for Research*' held from 5<sup>th</sup> Sep 2016 to 9<sup>th</sup> Sep 2016, organized by Central Library, Delhi Technological University.
- Member, *UMEED – A Drop of Hope*, for social work during September 2014

### **Personal Information:**

- Date of Birth: 28<sup>th</sup> May 1993
- Email: [singhrupam99@gmail.com](mailto:singhrupam99@gmail.com)
- Mobile: 9219273618
- Address: Electrical Engineering Department, Delhi Technological University, New Delhi-110085

I hereby declare that the information provided is true to best of my knowledge.

RUPAM SINGH