ANALYSIS AND DESIGN OF CONTROLLERS FOR TWO WHEELED ROBOT

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF

MASTER OF TECHNOLOGY
IN
CONTROL AND INSTRUMENTATION

Submitted by:

VIDYA LIKHITHA KAKINADA
2K19/C&I/14

Under the supervision of

Mr. Kuldeep Singh

DEPARTMENT OF ELECTRICAL ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

2021

DEPARTMENT OF ELECTRICAL ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, VIDYA LIKHITHA KAKINADA, Roll No. 2K19/C&I/14 student of M.Tech (Control & Instrumentation), hereby declare that the project Dissertation titled "Analysis and design of controllers for Two Wheeled robot" which is submitted by me to the Department of Electrical Engineering Department, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi **Vidya Likhitha Kakinada**

Date: 13-09-2021

DEPARTMENT OF ELECTRICAL ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CERTIFICATE

I, VIDYA LIKHITHA KAKINADA, Roll No. 2k19/C&I/14 student of M. Tech. (Control & Instrumentation), hereby declare that the dissertation/project titled "Analysis and design of controllers for Two Wheeled Robot" under the supervision of Mr. Kuldeep Singh of Electrical Engineering Department Delhi Technological University in partial fulfilment of the requirement for the award of the degree of Master of Technology has not been submitted elsewhere for the award of any Degree.

Likhitha

Place: Delhi **(VIDYA LIKHITHA KAKINADA)**

Date: 13-09-2021

Kuldeep
14-09-2021

(KULDEEP SINGH)
ASSISTANT PROFESSOR
Supervisor

DEPARTMENT OF ELECTRICAL ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

I would like to express my gratitude towards all the people who have contributed their precious time and effort to help me without whom it would not have been possible for me to understand and complete the project.

I would like to thank Assistant Prof. Kuldeep Singh, DTU Delhi, Department of Electrical Engineering, my Project guide, for supporting, motivating and encouraging me throughout the period of this work carried out. Their readiness for consultation at all times, their educative comments, their concern and assistance even with practical things have been invaluable.

Date: 13-09-2021

Vidya Likhitha Kakinada
M.Tech (Control & Instrumentation)
Roll No. 2K19/C&I/14

ABSTRACT

An inverted pendulum inspired Two wheeled self-balancing robot (TWSBR) finds several applications in real life. It can also be used to test the working of different control algorithms. In this thesis work the two-wheeled system’s transfer function is modelled. In this thesis work the two-wheeled system’s transfer function is modelled. Further, for performing the simulation for stabilizing the system three different controllers namely PID, Fractional order PID (FOPID) and Fuzzy PD (FzPD) have been employed. It is common knowledge that controllers give better performance with proper values of the several parameters. These parameters can be adjusted to improve the controller's response. In this work water cycle algorithm (WCA) has been used to tune the parameters of all three controllers.

Comparative analysis among different controllers shows that FzPD controller outperforms the PID and FOPID controller in terms of stabilizing the system. Various graphs have been exhibited in the thesis work to demonstrate the working of the simulation.

Table of Contents

List of Figures

List of Tables

List of Acronyms

FOPID: Fractional Order Proportional Integral Derivative

FzPD: Fuzzy Proportional Derivative

IAE: Integral Absolute Error

ISE: Integral Squared Error

ITAE: Integral Time Absolute Error

ITSE: Integral Time Squared Error

PID: Proportional Integral Derivative

TWSBR: Two-wheel Self Balancing Robot

WCA: Water Cycle Algorithm

Chapter 1

Introduction

CHAPTER 1
INTRODUCTION

1.1. Introduction

The TWSBR has developed a warm subject in proving numerous control concepts, which mostly regulate the system during the unbalanced dynamic as well as nonlinearities conditions. Along with that, it has other benefits for self-governing mobile such as insignificant size, simple construction as well as flexibility in operation. As a result of this concept, it may be appropriate for some labor that takes place in a confined space or is hazardous. In general, a TWSBR is an excellent resource to test a variety of control algorithms in both civilian and military applications. [1-3].

Researchers have completed a great research on TWSBR at home as well as abroad and projected many approaches to stabilize the nonlinear system. Many researchers have implemented the linearization technique to stabilize the nonlinearity of the robot. After that, investigated the linearization technique in the view of the recent control theory like pole placement or LQR. These linear techniques, on the other hand, assume that the disposition of robot retains inside the prescribed small angle. However, in training, the possibility of the governable angle is considerably more. The linearization can achieve excellent response in the vicinity of the balance point and at the same time cannot retain stability when it goes far away from the stable point [4-6].

TWSBR is a multi-variable as well as undefined nonlinear model so that the operation of the robot rest on deeply on the signal processing as well as the control technique [7-8]. In recent centuries, the many researches have worked on backstopping control, which delivers a influential design device for nonlinear system for the propose of feedback. That’s why, it seems huge interest and reasonable to implement backstopping method to plan a compatible controller for TWSBR [9-10].

1.2. System Modelling

The transfer function of the self-balancing robot system is estimated by the help of mathematical calculations. A TWSBR is similar to an inverted pendulum in terms of design. The inverted pendulum falls apart during the absence of balancing force.

According to that, it can be decided that the proposed system is unbalanced. The position of the pendulum, on the other hand, is normal or not inverted, indicating the robust nature.

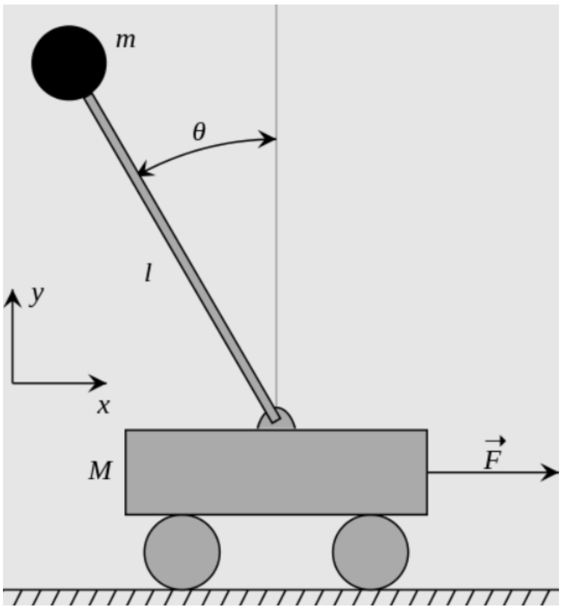

Fig. 1-1: Inverted pendulum's model.

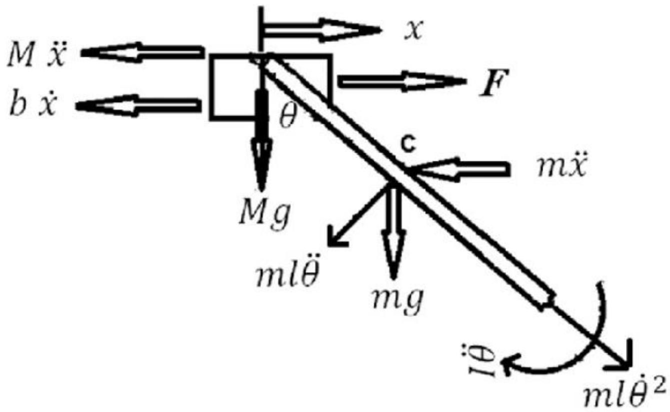

Fig. 1-2: Free body representation of the Inverted pendulum.

Figure 1-1 describes the structure of an inverted pendulum along with tilt angle θ. F signifies the force injected to the designed system. M symbolizes the mass of the model, x and y are the vertical as well as horizontal axis, correspondingly. During the practical execution, DC motors will afford the proper velocity, which helps the robot to remain stable. Deprived of the proper velocity, the proposed robot will be dropped.

The Newton II law is used to implement the dynamic equation for TWSBR. Figure 1-2 displays the Free Body representation of the proposed system. The pendulum's slope has various angles with the two components along the horizontal as well as vertical directions. Table 1 describes the constraints for TWSBR. Based on the Figure 1-2, the mathematical approach for modelling self-balancing robots, can be described as follows [17].

$$\sum f_x = 0 \tag{1.1}$$

$$F - b\dot{x} - M\dot{x} - m\ddot{x} - ml\ddot{\theta}cos\theta + ml\dot{\theta}^2 sin\theta = 0$$

$$\sum f_y = 0 \tag{1.2}$$

$$Mg + mg - ml\ddot{\theta}sin\theta + ml\dot{\theta}^2 cos\theta = 0$$

$$\sum M_A = 0 \tag{1.3}$$

$$Mglsin\theta + I\ddot{\theta} - ml^2\ddot{\theta} + ml\ddot{x}cos\theta = 0$$

By combining equations (1.1), (1.2), and (1.3), later the equation for the force is derived in both horizontal as well as vertical directions.

$$(I + ml^2)\theta + mglsin\theta = -ml\ddot{x}cos\theta \tag{1.4}$$

$$(M + m)\ddot{x} + b\dot{x} + ml\ddot{\theta}cos\theta - ml\dot{\theta}^2 sin\theta = F \tag{1.5}$$

Equations (1.4) and (1.5) represent the linearized equations of the transfer function, where $q = \pi$. With considerations $\theta = \pi + \emptyset$,

$$cos\theta = -1 \tag{1.6}$$

$$\sin\theta = -\emptyset \tag{1.7}$$

$$\frac{d^2}{dt^2} = 0 \tag{1.8}$$

After executing the equations (1.6), (1.7) as well as (1.8) method to non-linear, we find two discrepancies of motion equations. The U signifies the system's input.

$$(I + ml)^2\ddot{\emptyset} - mgl\emptyset = ml\ddot{x} \tag{1.9}$$

$$(M + m)\ddot{x} + b\dot{x} + ml\ddot{\emptyset} = U \tag{1.10}$$

1.3. Transfer Function Model

By implementing Laplace transform on equations (1.9) and (1.10),

$$(I + ml^2)\ddot{\emptyset}(s)s^2 - mgl\emptyset(s) = mlX(s)s^2 \qquad (1.11)$$

$$(M + m)X(s)s^2 + bX(s)s + ml\emptyset(s)s^2 = U(s) \qquad (1.12)$$

Table I: Physical parameters of the TWSBR

Variable	Description	Value
M	Cart's mass	0.3 Kg
m	Pendulum rod's mass	0.2 Kg
b	Viscous friction Co-efficient	0.1 N/m/sec
l	Pendulum's one half rod length	0.15m
I	Moment of inertia for the pendulum rod	0.0015 Kg.m^2
F	Applied force to the cart	$Kg.m/s^2$
g	Gravity	m/s^2
θ	Pendulum angle	rad

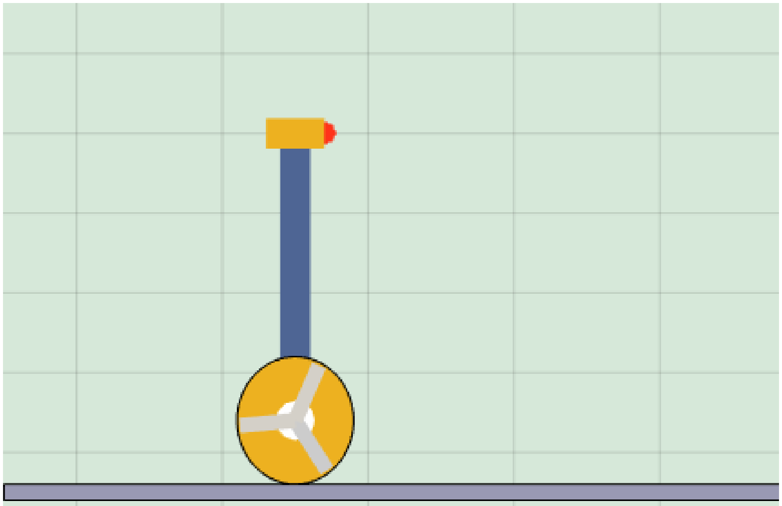

Fig. 1-3: Animation Model of the system

We generated the transfer function of TWBMR based on the equations (1.11) and (1.12).

$$\frac{\emptyset(s)}{U(s)} = \frac{\frac{ml}{q}s}{s^3 + \frac{b(I + ml^2)}{q}s^2 - \frac{(M + m)mgl}{q}S - \frac{bmg\grave{l}}{q}} \tag{1.13}$$

Where $q = [(M + m)(I + ml^2) - (ml)^2]$

The constraints given in Table 1 are replaced in equation (1.13), we get the transfer function of TWBMR.

$$\frac{\emptyset(s)}{U(s)} = \frac{14.286s}{s^3 + 0.286s^2 - 70s - 0.0294} \tag{1.14}$$

1.4. Thesis Organisation

This section provides the overview of the thesis. After providing the introduction in Chapter 1, Chapter 2 presents the literature survey conducted on the problem under observation. Chapter 3 presents the description of the controllers utilised in this research. Chapter 4 describes the optimization algorithm and the procedure to use optimization algorithm in this study. Chapter 5 lists out the results. Conclusion and future scope of the work done in this thesis has been provided in Chapter 6.

Chapter 2

Literature Survey

CHAPTER 2

LITERATURE SURVEY

2.1. Introduction

The TWSBR finds various applications in different areas. Several control experts in the control area have been drawn to this problem in order to design and develop improved control algorithms for the TWSBR. A lot of work have been done in the controller designing part of this problem. In this chapter, a detailed literature survey about work done by various researchers in this domain has been presented. Further conclusion has been drawn from the literature survey to do the presented study.

2.2. Work Done by Others

Jian Fang implemented the particle swarm optimization technique to tune the constraints matrix belonging to LQR controller for building the two-wheeled as well as self-balancing robot by reducing the overshoot and disturbance in frequency for maintaining the system's stability. The simulation investigation depicts the proposed LQR controller enhancing the system's steadiness, finds the good control consequence as well as has advanced application value via by employing the particle swarm algorithm [11].

Xiaogang Ruan et. al. proposed a two-loop cascade controller for a non-linear and non-stable coupling system introducing backstepping as well as fuzzy neural network (FNN). The implemented technique implements FNN to estimate unidentified nonlinear function. After that, implemented backstepping to execute adaptive controller to understand self-balancing control mechanism of robot [12].

He Bin et. al. described the kinematics architecture of a TWSBR. After the robot's mechatronics have been completed, the scrutiny of the complete kinematics model can be divided into two wheels and a body for inspection. Then the velocity breakdowns of robot's wheel as well as the body are examined correspondingly. The kinematic architecture of robot's self-balancing body is also evaluated using the method outlined above [13].

Rasoul Sadeghian e. al. r presented a dynamic structure as well as control approaches for a self-balancing robot, which is estimated by using the Newtonian method. The constraints of the designed robot are used to control tilt angle along with the displacement of the projected robot. The controller's constraints are optimized by Genetic optimization algorithm. Fuzzy logics are used to improve equilibrium, especially when external pressures are present. After executing the intended Fuzzy logic based PID controller to examine the TWSBR, the steadiness of the designed system is upgraded impressively [14].

Qin Yang et. al. executed a dynamic as well as the kinematics for non-holonomic based moving robot. A T-S fuzzy model with a parallel distribution compensator (PDC) structure has been devised to accomplish more regulation for self-balancing robots. [15].

Mohammad Mahdi Azimi et. al. presented a model predictive control (MPC) for a TWSBR, which is also called as a three-degree-of-freedom mobile robot along with multivariable dynamics. The dynamic equations along with input as well as output of the system are allotted. Hence forward, the actual system is converted into two subsystems by employing a compensator block [16].

Navid Razmjooy et. al. proposed a novel technique linked with interval analysis for designing the two-wheeled as well as self-balancing robot along with interval undefined parameters which need lower and higher bounds of undefined constraints without knowing regarding probability distributions. Due to system uncertainties, first analyzation has been done on controllability regarding interval arithmetic. Then, LQR based technique associated with Pontryagin principle has utilized to resolve the issuse [17].

P. frankovský et. al. presented the design of a TWSBR operated by using geared DC motors. A precise design comprises of two foremost parts, first one is the mechanical structure of robot and the second one is the actuator's modelling. For the purpose of state feedback controller design, linearized motion calculations are determined and the state-space modelling of the intended robot is provided. [18].

Congying Qiu et. al established a TWSBR architecture along with the hardware arrangement mainly comprising of a controller called as TMS320LF2407 DSP. Also a foremost sensor of Mio-x AHRS module, and many inexpensive components. The linear controllers have many critical faults to regulate the designed robot, like lengthy settling time as well as large overshoot. Hence forward, we have done additional research to design fuzzy adaptive PID controller along with an enhanced architecture [19].

Bernhard Mahler et. al presented the idea of TWSBR. A gyroscope is employed to determine the state by introducing a novel technique of measurement information blend for the accelerometer as well as the gyro by utilizing a drift compensation controller. Furthermore, measurement reports for the variables of an actual prototype robot are recommended as well as the control mechanism is modelled for this proposed robot. [20].

Anh Mai et. al established an intellectual system along with fuzzy controller utilizing Mamdani algorithm over a TWSBR. Hardware configuration of the designed robot as well as the operating procedure of sensor for signal processing were illustrated. Discrete complementary filter was utilised to filter the signals within the sensors. A mathematical design of the proposed robot has done with Newtonian mechanics. Pitch angle as well as tracking the required location of the robot are two important loops of the described control system. The PD controller was implemented to track the position. Adaptive fuzzy logic-based controller was employed to normalize balancing of the designed robot [21].

Omer Saleem Bhatti et. al demonstrated an advance attitude control along with stabilization method of a self-balancing robot. The foremost intention is to confirm its vertical steadiness in the existence of an exterior bounded impulsive force. The projected robotic arrangement can be stable at the required set-point angle by designing a hard-coded vertical position for reference. The direction and the degree of disposition of the robot are measured by an inertial-sensor. The projected structure implemented a mixture of first-order spatial filters in order to eliminate the noise as well as to combine the analogy sensor evaluations [22].

Tao Zhao et.al. offered numerous non-singleton general type-2 fuzzy logic controllers (NGT2FLCs) for an under-actuated mobile TWSBR to improve the structure's anti-interference ability. Four varieties of fuzzifiers, such as singleton fuzzifier, type-1 non-singleton fuzzifier, interval type-2 non-singleton fuzzifier and general type-2 non-singleton fuzzifier, are mentioned to design various general type-2 fuzzy logic controllers (GT2FLCs) [23].

Ali Ghaffari et. al introduced the mathematical illustration of two-wheeled self-balancing robots using Kane’s as well as Lagrangian approaches. The significant consequence of the new approaches on the outcome of the structure is revealed by showing the behaviour of the system beneath different situations. Then sliding-mode control methods are employed to execute the required operation [24].

Fengxin Sun et. presented an approach to regulate a TWSBR associated with Kalman filter as well as LQR algorithm. It is united with configuration of hardware, signal processing and design of control algorithm. The angle signs disposed with the help of Kalman filter from a gyroscope as well as an accelerometer, are united with LQR controller to complete the monitoring process of the robot. The intension of easening Kalman filter is to achieve the signal processing by the help of less system supplies and reduce the specified system's burden [25].

Chapter 3

Controllers

CHAPTER 3

CONTROLLERS

3.1. Introduction

Controllers are the essential part of the current project work. In this thesis three different controllers namely proportional-integral-derivative (PID), fractional order PID (FOPID) and fuzzy proportional-derivative (FzPD) have been used and tested for the stability performance of the system. The architecture of the individual controller is briefly discussed in this chapter.

3.2. PID Controller

The PID controllers are the utmost widespread controllers employed in industry due to their simplicity, sturdiness. An investigation has revealed that 90% of control methods are of PI as well as PID structure. Good responsiveness of design as well as tuning approaches is growing important to both operators as well as designers. The procedure of a PID controller should comprise three aspects such as identification of accurate process model, designing the structure of controller and finally tuning the PID parameters, which is an essential part to design the controller. The model selection is the foundation to tune the parameters as well as the structure of controller to decouple with the set-point and load fluctuation rejection.

It is needful to select the appropriate process model before designing a controller. It can help to determine the limits of control signals needed to vary the output of model over the required range, to choose the resolution of the sensor as well as to size actuators. It has been noticed that a control method should diminish the consequence of load turbulences, avoid noise to be injected into the designed system and be of vigorous towards reasonable variations during the process.

Each shortcoming and strength of the P, I, and D controllers can be compensated for by combining them in a PID controller parallely. The PID control system's combination of proportional, integral, and derivative controls serves a specific function. Fast rising times are a strength of proportional control, whereas steady-state error can be eliminated using integral control. Derivative control helps to reduce overshoot. When we combine these, we get a control with error-reducing qualities, with rise time reduction, settling, and overshoot times. [32-33]. The PID controller's transfer function is showed as follows:

$$K(s) = K_P + \frac{K_I}{s} + K_D s \tag{3.1}$$

In the above equation K_P is known as gain of the proportional controller, $K_I$is known as integral controller's gain, and K_D is known as differential controller's gain, correspondingly. If reducing the coefficient of proportional controller to diminish the overshoot, the system's response will be sluggish. Figure 3-1 signifies the block diagram of PID controller. Moreover, it may not be able to accomplish the decrease in overshoot, since when robot preserves in balance condition, its tilt angle usually doesn't surpass ± 3°. Integral separation optimization algorithm isn't appropriate. Though integral disturbance can decrease the overshoot, it deteriorates the volume of integral departure tracking. Expanding differential constraints can decrease overshoot, but the constraints are likely activated to be huge; and subsequently, the friction of the system will be too lengthy to bring a faster outcome to the system. In adding, inflated differential constraints will present additional speed disturbances to the system [34].

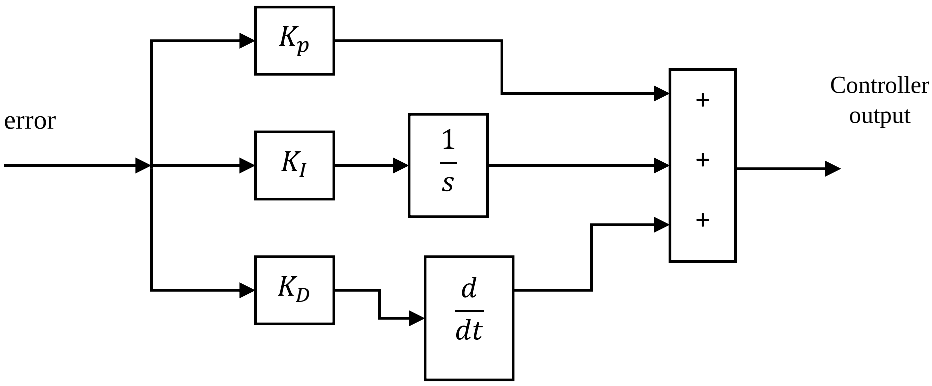

Fig. 3-1: Arrangement of the PID controller.

3.3. FOPID Controller

The Fractional Order Calculus comprises the mathematics calculations along with differentiation as well as integration under an arbitrary order of the process, where the order can be any real or complex number, not just an integer. In 1695, Leibniz possibly revealed the concept of non-integer derivative to L'Hospital. Following that, several researchers explained the innovative efforts of FOC. Though, nowadays, FOC is a helpful tool for a variety of challenges such as "indefinite memory," chaotic behaviour, and so on. As a result, the FOC is currently extremely useful in a variety of engineering fields, including viscoelasticity, robotics, and signal processing. Also, bioengineering, electronics, and control theory are three fields of study.

The fractional order PID controller is a fractional calculus extension of the regular PID controller. PID controllers have been gaining acceptance in various enterprises for decades. Their merit involves in uncomplicatedness of design as well as good performance, which belongs to small percentage of overshoot and insignificant settling time. Due to the extreme centrality of PID controllers, ongoing efforts are made to improve their quality as well as resilience. The FOPID controller, which is a simplification of typical integer order controllers, would result in more precise and robust control responses in the field of automatic control. Although it might be true that fractional order dynamics involve fractional order controllers to achieve the best results, in many other instances, FOPID controllers are often used to improve system control responses in normal linear or nonlinear dynamics.

In the field of control systems, fractional calculus can be used in a variety of different ways because the derivatives as well as integrals can be any real number. As a simplification of the differential as well as integral operators, the FOPID can be represented by an overall fundamental operator ${}_{\propto}D_t^q$,which is well-defined as follows.

$$
{}_{\propto}D_t^q = \left\{ \begin{array}{ll} \dfrac{d^{\propto}}{dt^{\propto}} & Re \propto > 0 \\ 1 & Re \propto = 0 \\ \displaystyle\int_{\propto}^{t} (d\tau)^{-\propto} & Re \propto < 0 \end{array} \right\} \tag{3.2}
$$

There are two commonly used definitions for the general fractional differentiation and integration, i.e., the Grünwald–Letnikov (GL) and the Riemann Liouville (RL) definitions.

The definition of GL is represented below as:

$$ {}_{\propto}D_t^q = lim_{h\to 0}h^{-\propto} = \sum_{j=0}^{\left[\frac{t-\propto}{h}\right]} (-1)^j \binom{\propto}{j} f(t-jh) \tag{3.3}$$

While the definition of RL is represented by:

$$ {}_{\propto}D_t^q = \frac{1}{\Gamma(\mathrm{n}-\mathrm{\alpha})}\left(\frac{d}{dt}\right)^n \int_{\alpha}^{t} \frac{f(\tau)d\tau}{(t-\tau)(\alpha-n+1)} (n-1 \le \alpha < n \tag{3.4}$$

Where n represents an integer and $\propto$ is a real number. $\Gamma(x)$ is the popularly known Euler's Gamma function. The fractional order FOPID controller's differential equation is defined by

$$u(t) = k_P e(t) + k_I D_t^{-\lambda} e(t) + k_D D_t^{\mu} e(t) \tag{3.5}$$

The FOPID's continuous transfer function of is attained via Laplace transform and is specified by

$$G_c(s) = k_P + k_I s^{-\lambda} + k_D s^{\mu} \tag{3.6}$$

Modelling of an FOPID controller includes tuning of three constraintsk_P, k_I, k_D, as well as two orders λ and μ that really aren't fundamentally integer. The proposed controller is a simplified version of the classic PID controller with integer order. This extension may provide more flexibility in meeting control goals.

The FOPID controller has the potential to improve the system's control functioning. The utmost significant benefits of the proposed controller are the improved control mechanism of dynamical control systems, which are defined with the help of fractional order mathematical structure. Additional benefit that the FOPID controllers are not much sensitive to the alteration of constraints of a controlled system. The presence of two more degrees of freedom improves the dynamical properties of a fractional order Control system.

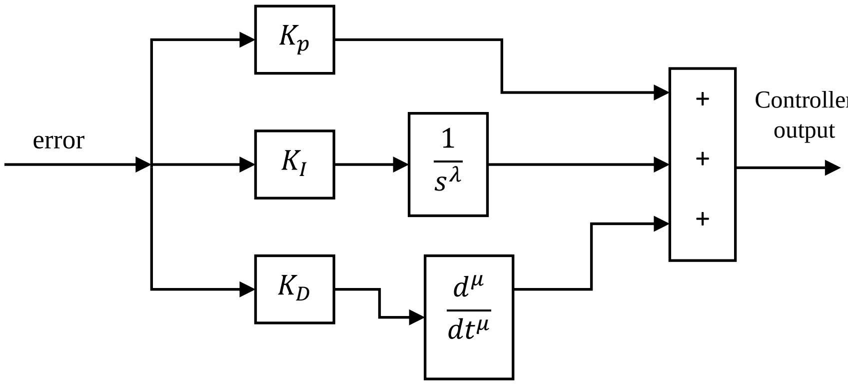

Fig. 3-2: FOPID controller's Block Diagram

3.4. Fuzzy PD (FzPD) Controller

The concept of fuzzy logic was first introduced by L.A. Zadeh in 1965. He discovered that the conventional notion places a greater emphasis on correctness than on simple and effective control methods. The sets of Fuzzy logic comprise of certain degree of membership for every element, which rest on certain instructions. The rule base is the utmost significant necessity to execute fuzzy logic. The rule base usually consists of many cases such as If-Then rules. Firstly the fuzzy sets as well as the membership functions must be defined. Henceforward, the If-Then rules will now be used to determine membership functions for a certain control. The output of the modelled system is regulated by these defined rules on input part. A general If-Then rule involves of two portions, such as Antecedent and Consequence.

Fuzzy logic control mostly relies on the rules designed with the help of Linguistic variables. The control methodologies of fuzzy logic do not necessitate compound analytical computations, such as alternative approaches. It merely uses modest arithmetical computations to regulate the process model. Regardless of depending on fundamental arithmetic investigation, it performs well to regulate the control system. Henceforward, this technique is the utmost approaches accessible as well as also simpler one to regulate a plant. According to the fuzzy set concept, every component has a degree of membership that is associated to a specific set. Fuzzy based Controllers are mostly employed when accuracy needed is modest. Also, the process plant must be empty to accomplish the sophisticated mathematical investigation [35-37]. Figure 3-3 represents the block schema of fuzzy logic based PD controller. Figure 3-4 represents the FzPD controller graphically.

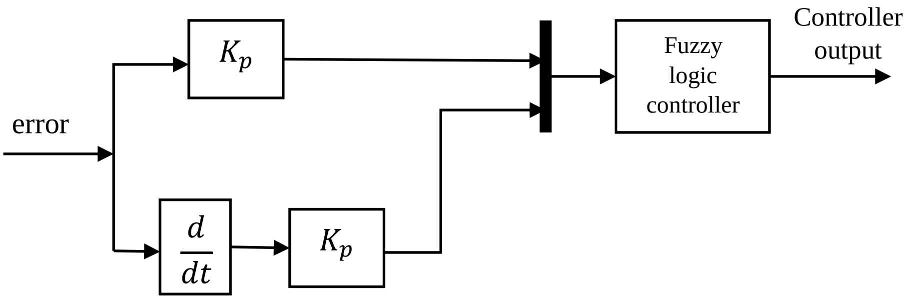

Fig. 3-3: Block design of the FzPD controller

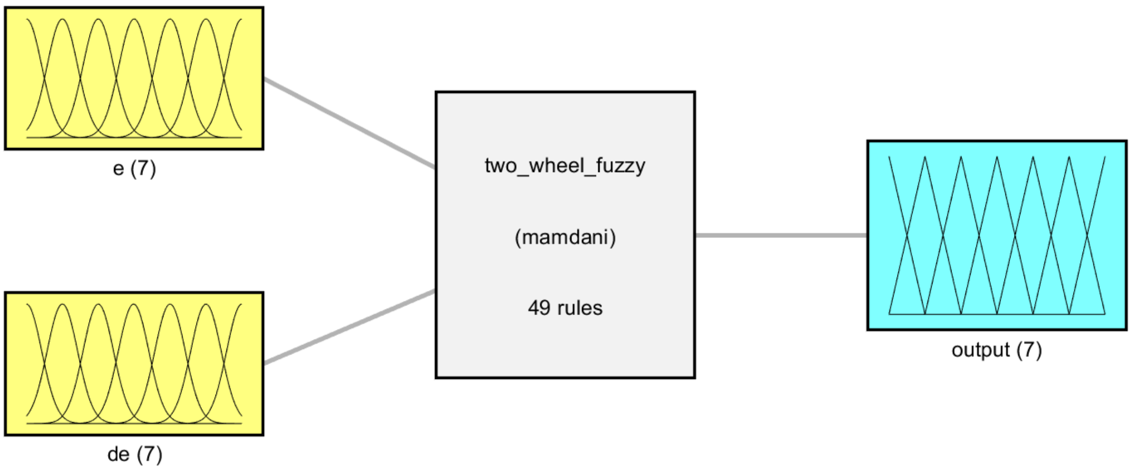

Fig. 3-4: Inside View of FzPD controller

The Fuzzy Logic controller's three foremost elements are

1. Fuzzification.

2. Fuzzy Rule base and Interfacing engine.

3. Defuzzification.

3.4.1.Fuzzification

The utmost imperative step to formulate the design of the fuzzy controller is to select the state variables which effectively regulate the plant. After computation of these variables, they are passed through the fuzzification process. The Fuzzy Rule base implements rules which only belongs to the linguistic variables. During fuzzification the numerical value of the input variable is converted into the fuzzy linguistic variable in which it is represented as the member of different membership function with a certain degree. The variable in Fuzzy-PD controller which are going to be fuzzified are error and derivative of the error or the area of the error. The MFs are the pictorial representation of the belonging of the input variable to the fuzzy set. Generally used MFs are trapezoidal, triangular, gaussian or most famous bell shaped. In this thesis work gaussian membership functions has been used for the input error and derivative of error.

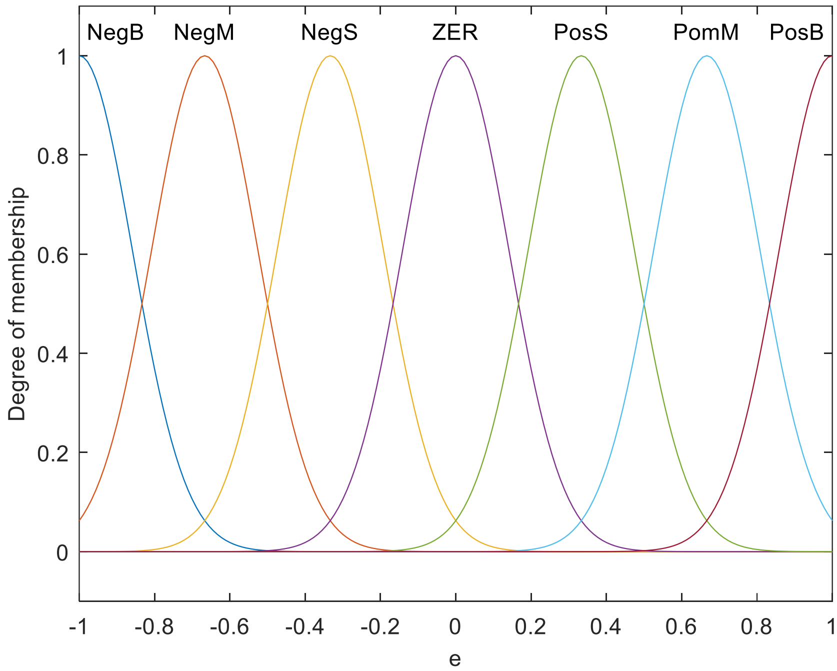

Fig. 3-5: 1st input error's Membership function.

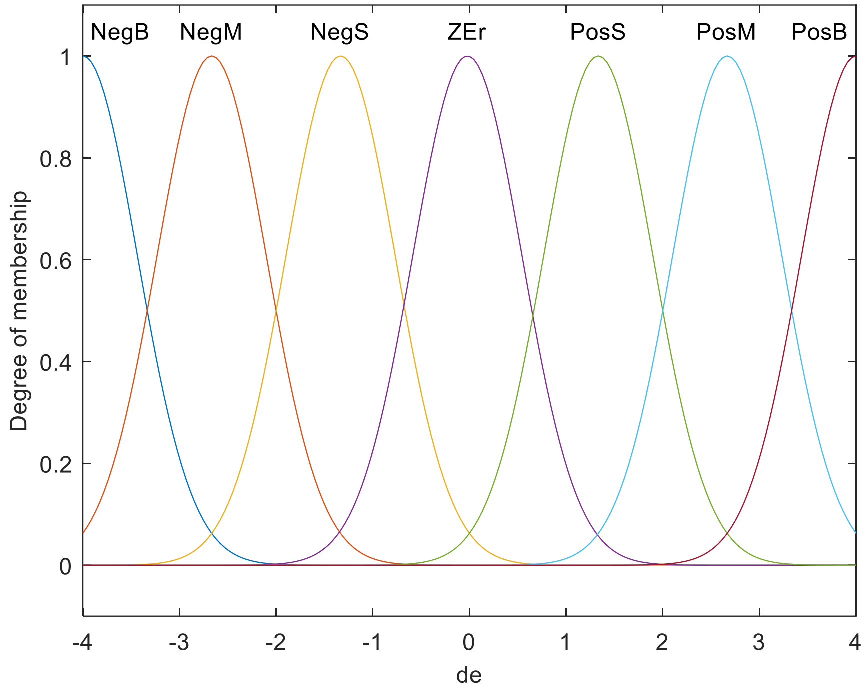

Fig. 3-6: 2nd input change in error's Membership function.

3.4.2.Fuzzy Inference

The two methods of Fuzzy inference are Mamdani as well as Sugeno. The method of Mamdani is the utmost normally employed technique in the fuzzy interference, which was first introduced by Ebrahim Mamdani to regulate a steam engine as well as boiler combination by analysing a set of linguistic control rules attained from skilled human operatives. This inference technique is based on the assumption that the output variables are fuzzy sets. It is more likely and effective to use a single spike in the output as a membership function instead of a scattered fuzzy set, which is recognised as a singleton output membership function. Because the more typical Mamdani approach, which calculates the centroid of the two-dimensional function, requires more processing, it enhances the Defuzzification procedure. However, the Sugeno type of inference can be used to create any inference method with a linear or constant output membership function. Rule base for proposed inference mechanism is shown by Table II. Total 49 rules were used in this thesis work. Figure 3-7 show the rule surface for the proposed FzPD system.

Table II: Rule Base for the FzPD

Sr. No	***e***	***de***	***Output***
1	NEGB	NEGB	NEGB
2		NEGM	NEGB
3		NEGS	NEGB
4		ZER	NEGB
5		POSS	NEGM
6		POSM	NEGS
7		POSB	ZER
8	NEGM	NEGB	NEGB
9		NEGM	NEGB
10		NEGS	NEGB
11		ZER	NEGM
12		POSS	NEGS
13		POSM	ZER
14		POSB	POSS
15	NEGS	NEGB	NEGB
16		NEGM	NEGB
17		NEGS	NEGM
18		ZER	NEGS
19		POSS	ZER
20		POSM	POSS
21		POSB	POSM
22	ZER	NEGB	NEGB
23		NEGM	NEGM
24		NEGS	NEGS
25		ZER	ZER
26		POSS	POSS
27		POSM	POSM
28		POSB	POSB
29	POSS	NEGB	NEGM
30		NEGM	NEGS
31		NEGS	ZER
32		ZER	POSS
33		POSS	POSM
34		POSM	POSB
35		POSB	POSB
36	POSM	NEGB	NEGS
37		NEGM	ZER
38		NEGS	POSS
39		ZER	POSM
40		POSS	POSB
41		POSM	POSB
42		POSB	POSB
43	POSB	NEGB	ZER
44		NEGM	POSS
45		NEGS	POSM
46		ZER	POSB
47		POSS	POSB
48		POSM	POSB
49		POSB	POSB

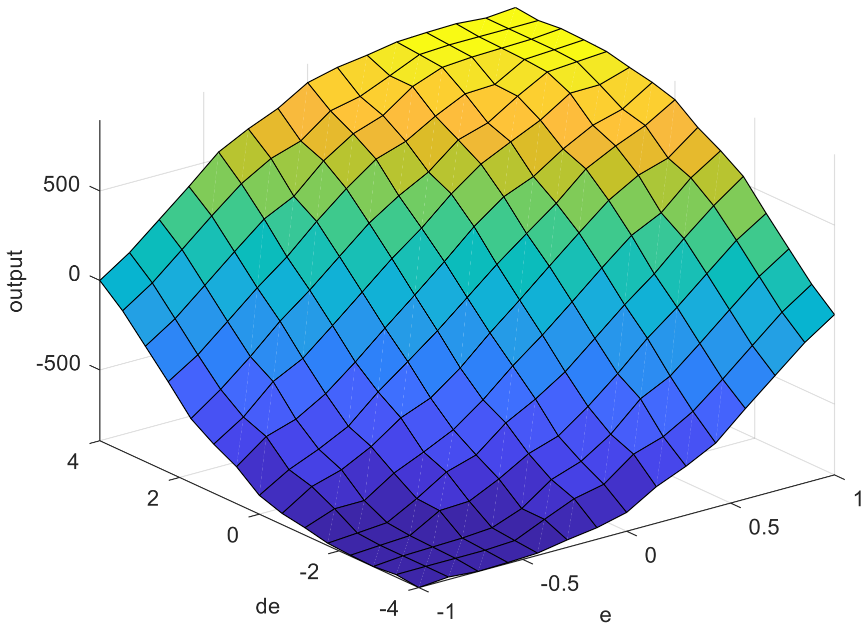

Fig. 3-7: Surface view of rules.

3.4.3.Defuzzification

Defuzzification is basically reversal of Fuzzification process. The Fuzzy Logic Controller generates a linguistic variable as the outcome. These variables must be translated to crisp output as described by legit fundamentals. There are several techniques available to do this task. In this work, centre of gravity approach has been used for the Defuzzification purpose. It obtains the centre of gravity of a section associated with fuzzy set. The results should be included in a useful Defuzzification procedure. The most powerful typical fuzzy set enrolling ability is shaped like a triangle. Assume that the triangle is cut in a straight level line somewhere between the topmost and the base, and that the uppermost portion is detached, the resultant shape is a trapezoid. If the membership function employed previously was triangular, this method eliminates sections of the figures to obtain trapezoids (or dissimilar outlines if the earlier outlines were not triangles). Typically, these trapezoids are stacked on top of one another to form a single outline. The centroid of this is then calculated. The defuzzified output is given by the abscissa of the centroid [38-40].

The membership functions utilised in the output for defuzzification are shown in Figure 3-8.

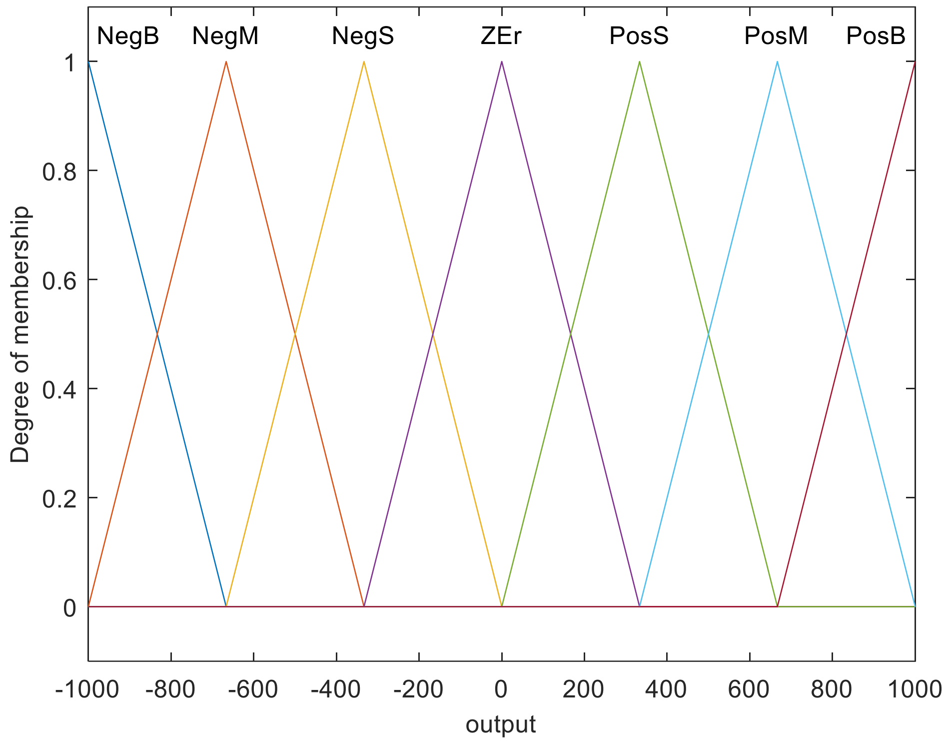

Fig. 3-8: Output MFs.

Chapter 4
Optimization Algorithm

CHAPTER 4

OPTIMIZATION ALGORITHM

4.1. Introduction

In the last chapter, controllers' description has been provided. The performance of the controllers can be noticed or seen to be fully dependent on the various parameters of the controllers. In this chapter controller parameter tuning method using optimization algorithm has been discussed. Further, different optimization algorithms and objective functions to be used for this purpose have also been discussed in this chapter.

4.2. Controller Parameter Tuning

It is a well-known fact that performance of the PID controller depends upon the correct value of its parameters K_p, K_i and K_d. There are numerous methods for determining the best value for these factors so that the plant's output remains within acceptable limits. Out of several ways, optimization is most commonly used and powerful method to tune the parameter of any controller. Fig. 4-1 shows the optimization procedure to tune the parameters of underlying controller.

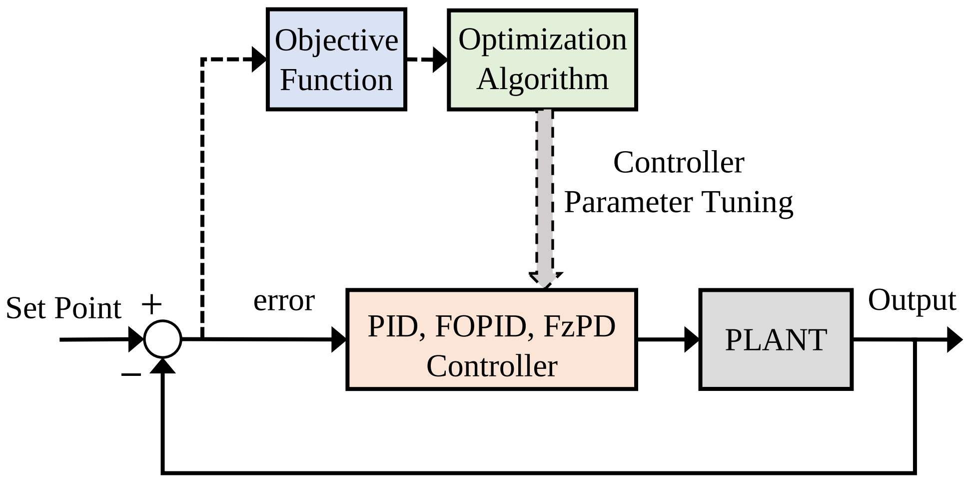

Fig. 4-1: Block Diagram of Optimization Procedure

From the Fig. 4-1, one can understand that objective function is computer from the error and then this value is used by optimization algorithm which in turn generate the different pairs of controller parameters. This process keeps going on till controller parameter values are achieved which results in the minimum value of objective function. Hence, through this process the plant output can be maintained at desired level with the help of optimized tanning parameters.

4.3. Objective Function

From section 4.2, it is clear that objective function plays a critical role in finding the best possible values of controller parameters. Optimization algorithm tries to minimize the objective function by generating different values of K_p and K_i. In many literature, objective function also known as cost function as well.

Objective function is a mathematical equation which depends upon the error value. In control system problem there mainly four types of objective functions are used. These objective functions are known as Integral Absolute Error (IAE), Integral Squared Error (ISE), Integral Time Absolute Error (ITAE) and Integral Time Squared Error (ITSE). Mathematical equations of the four objective functions are given by Eq. (4.1) – Eq. (4.4).

$$IAE = \int_0^t |e| \, dt \tag{4.1}$$

$$ISE = \int_0^t e^2 \, dt \tag{4.2}$$

$$ITAE = \int_0^t t * |e| \, dt \tag{4.3}$$

$$ITSE = \int_0^t t * e^2 \, dt \tag{4.4}$$

Optimization algorithm performance very much depends upon the kind of objective function as well. There are several research papers in control system literature, which have evaluated the performance of optimization algorithms for different objective functions. General conclusion is that ITAE based objective functions performs better than other objective functions. That’s why in this work we also have used the ITAE as an objective function for our optimization algorithms.

4.4. WCA Optimization Algorithms

In this part of the chapter, brief description of the optimization algorithm used for our work has been given. In this work we have used WCA optimization technique for tuning the controller's parameters.

The WCA imitators the movement of rivers as well as streams in the direction of the sea and observed by water cycle procedure. An initial streams' population is arbitrarily formed succeeding the raining procedure. The best stream having the least objective function, is selected as the sea [41-43]. Many reliable streams are selected as rivers, in the mean while others move to the rivers and sea. A stream of an array of $1 \times N$ is well-interpreted as follows:

$$steam\ Candidate = [x_i]_{1\times N}, \qquad \forall \in N \tag{4.5}$$

In the above equation N signifies the design control variable quantity; and x_i is considered as the values of decision variable.

To begin the optimization process, an initial population having streams of size $N_{pop} \times N$ is produced. N_{pop} is the over-all number of populations. The matrix having initial population, which is produced arbitrarily, is specified using Eq. (4.6).

$$Total\ Population = \left[x_{ij}\right]_{N_{pop}\times N} \qquad \forall \in N_{pop}, \forall j \in N \tag{4.6}$$

Every (x_j) can be signified as decimal numbers or real values as an undefined set for continuous as well as discrete issues, correspondingly. The stream's cost is found by the estimation of objective function (C) shown as follows:

First process, N_{pop} steams are formed. A quantity of $N_{sr} = N_r + 1$; N_r signifies the number of rivers along with one sea; out of the finest, individuals with least numbers are collected as the sea and rivers. The sea is supposed to be the stream with the lowest value among the others. The streams that go into rivers or right into the sea are computed employing (4.7).

$$N_{streams} = N_{pop} - N_{sr} \tag{4.7}$$

Eq. (8) represents the number of streams which move to the rivers or sea. Definitely, Eq. (4.7) is portion of Eq. (4.5).

$$population\ of\ stream = \left[x_{ij}\right]_{N_{stream} \times N} \quad \forall \in N_{stream}, \forall j \in N \tag{4.8}$$

The intensity of selected streams for every river as well as sea are estimated using:

$$C_n = Cost_n - Cost_{Nsr+1}, \forall n \in N_{sr} \tag{4.9}$$

$$NS_n = round\left\{\left|\frac{C_n}{\sum_{i=1}^{N_{sr}} C_i}\right| \times N_{stream}\right\} \quad \forall n \in N_{sr} \tag{4.10}$$

In the above NS_n signifies the quantity of streams that move to the particular rivers as well as sea.

Let's consider there are N_{pop} streams from which $N_{sr} - 1$ are picked-out as rivers as well as one is chosen as the sea. The displace X within the stream as well as the river may be arbitrarily updated employing the formula represented in Eq. (4.11)

$$X \in (0,\ C \times d), \qquad C > 1 \tag{4.11}$$

In the above equation $1 < C < 2$ and the finest value for C may be selected as 2; d represents the present distance within stream and river.

X agrees to an arbitrary value varies within 0 to $(C \times d)$. By assuming $C > 1$ permits streams to move in dissimilar orientations inclined towards rivers and after that rivers into the sea. Consequently, as the exploitation stage, the new locations for streams as well as rivers are defined as revealed in Eq. (4.12) - (4.14).

$$\vec{X}_{stream}^{t+1} = \vec{X}_{stream}^{t} + r(\dots) \times C \times \left(\vec{X}_{river}^{t} - \vec{X}_{stream}^{t}\right) \tag{4.12}$$

$$X_{stream}^{t+1} = \vec{X}_{stream}^{t} + r(\dots) \times C \times \left(\vec{X}_{sea}^{t} - \vec{X}_{stream}^{t}\right) \tag{4.13}$$

$$\vec{X}_{river}^{t+1} = \vec{X}_{river}^{t} + r(\dots) \times C \times \left(\vec{X}_{sea}^{t} - \vec{X}_{river}^{t}\right) \tag{4.14}$$

In the above equation, $r(\dots)$ signifies the random number $\in [0, 1]$. Position of river and stream is switched if stream's fitness function is better than its' connecting river. A related interchange is accomplished within a river and the sea. Mainly, sea water is formed due to evaporation and then it evaporated as rivers or streams move to the sea. In this algorithm, evaporation procedure is accountable for the exploration stage in the algorithm. For that resolution, the following condition is employed for evaporation situation:

$$if\ \|X_{sea}^{i} - X_{river}^{i}\| < d_{max}\ \ or\ rand(\dots) < 0.1, \forall i \in (N_{sr} - 1) \tag{4.16}$$

The newly produced sub-population, the finest stream flows like a new river. Another streams flow in the direction of the new river.

$$X_{stream}^{new} = L_B + r(..) \times (H_B - L_B) \tag{4.17}$$

In the above equation L_B and H_B are lesser as well as higher bounds well-defined by the specified issues, correspondingly. In the similar way, the finest newly generated stream is assumed to be a river moving into the sea while the rest are considered to move into the rivers or straight to the sea. A great value for d_{max} avoids additional searches as well as insignificant values inspire the severity of search closer to the sea. Consequently, d_{max} regulates the intensity of search closer to the sea such as finest obtained solution. The value of d_{max} adaptively reduces by using Eq. (4.18).

$$d_{max}^{i+1} = d_{max}^{i} - \frac{d_{max}^{i}}{Max_{iter}} \tag{4.18}$$

Chapter 5
Results & Discussion

CHAPTER 5

RESULTS & DISCUSSION

5.1. Introduction

As mentioned in previous chapters, in this thesis work three different controllers have been employed to control the TWSBR. This chapter presents the results obtained for individual controller have been demonstrated. Further performance comparison among different controllers have also been done.

The proposed work has been implemented with the help of MATLAB 2018. Plant model was simulated on SIMULINK. Other optimization related codes, objective function evaluation code and performance comparison coding was done on the editor of MATLAB. Personnel computer with 8GB RAM, Core i3 processor, 1 TB HDD and Windows 10 operation system was used for the entire simulation work.

5.2. Self-Balancing Robot using PID Controller

Figure 5-1 depicts the Simulink diagram of the proposed system model using PID controller. The following form of the PID controller was implemented for this work:

$$C(s) = K_p + \frac{K_i}{s} + \frac{K_d N}{1 + \frac{N}{s}}$$

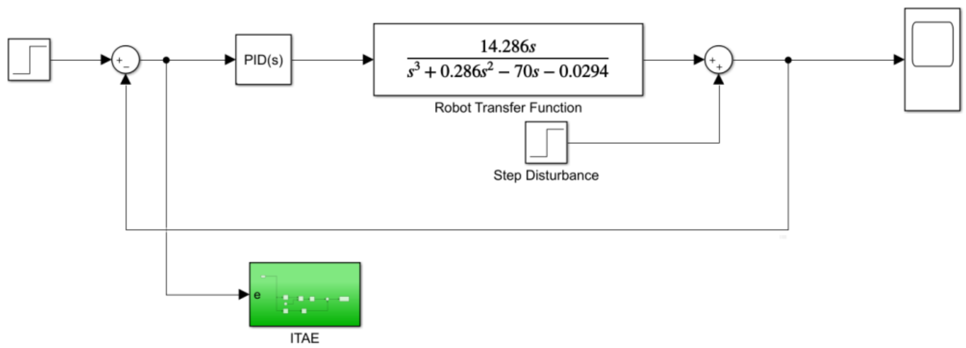

Fig. 5-1: Simulink model of the system with PID controller.

From Fig. 5-1 it can be seen that disturbance has been added at the output of the plant. Disturbance of 0.1 rad was added to plant output. In this case reference angle value was set at 0 rad as per the requirement. PID controller's parameters were tuned with the help of WCA algorithm. Different parameters used in the optimization task has been provided in Table III. Fig. 5-2 shows the convergence curve of the WCA optimization algorithm. WCA converges to the minimum ITAE function value of 0.00032 within the 6 iterations. Tuned parameter values of the PID controller are given in Table 2. Fig. 5-3 depicts the response of the system with tuned PID controller parameters. The figure shows that the system was stabilised with the help of a PID controller, but the steady state value did not reach 0 as required.

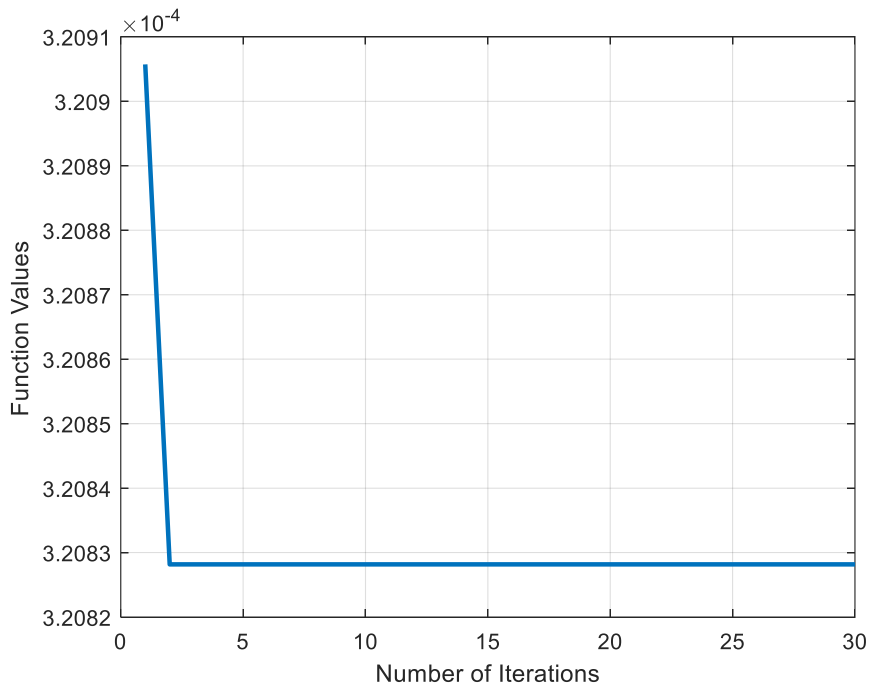

Fig. 5-2: WCA convergence graph for PID controller tuning.

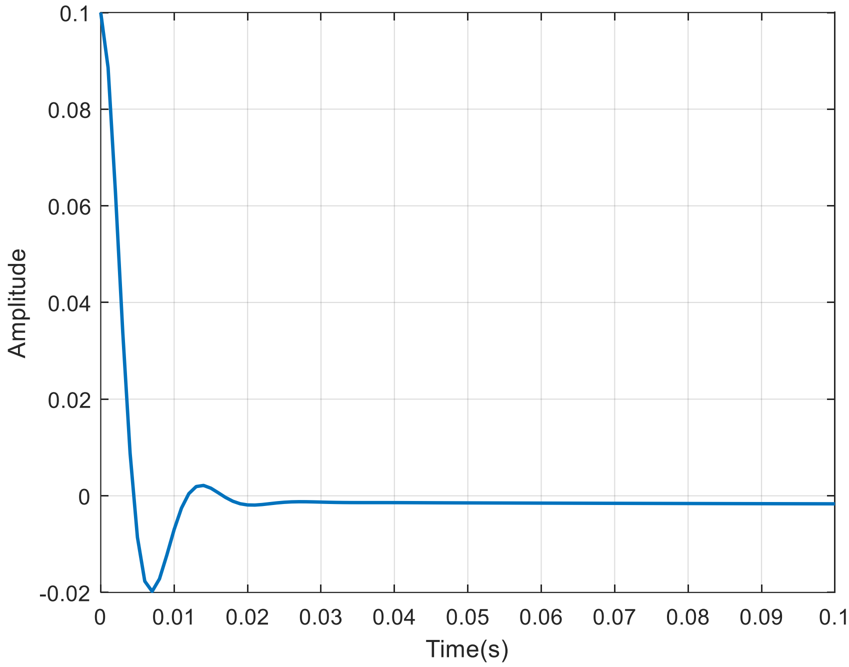

Fig. 5-3: Response of system with PID controller.

5.3. Self-Balancing Robot using FOPID Controller

The Simulink diagram of the proposed system model with the FOPID controller is shown in Figure 5-4. FOMCON toolbox was used to implement the FOPID controller.

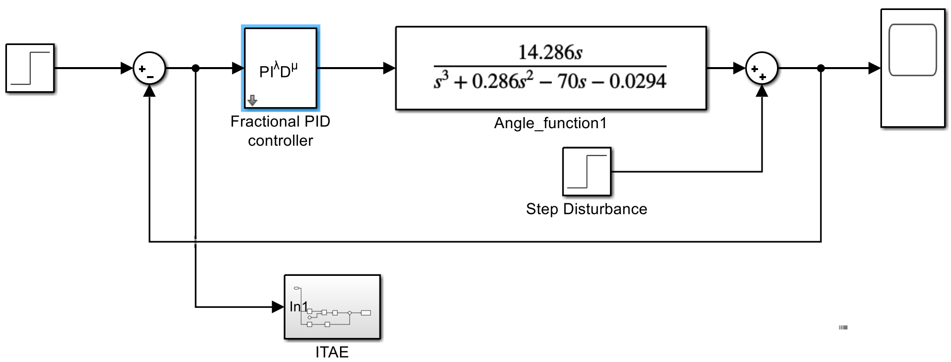

Fig. 5-4: System block diagram with FOPID controller.

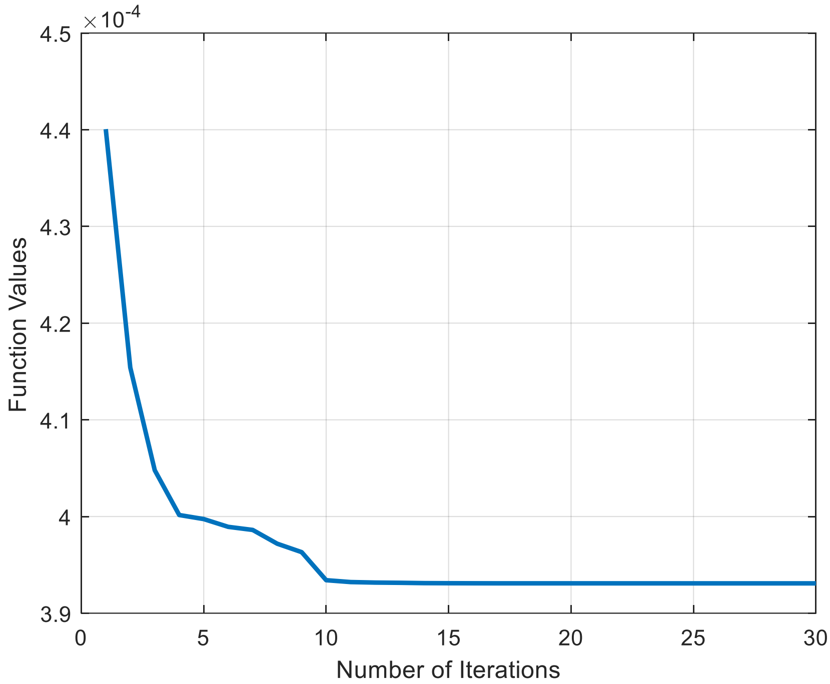

Fig. 5-5: Convergence curve of WCA for FOPID.

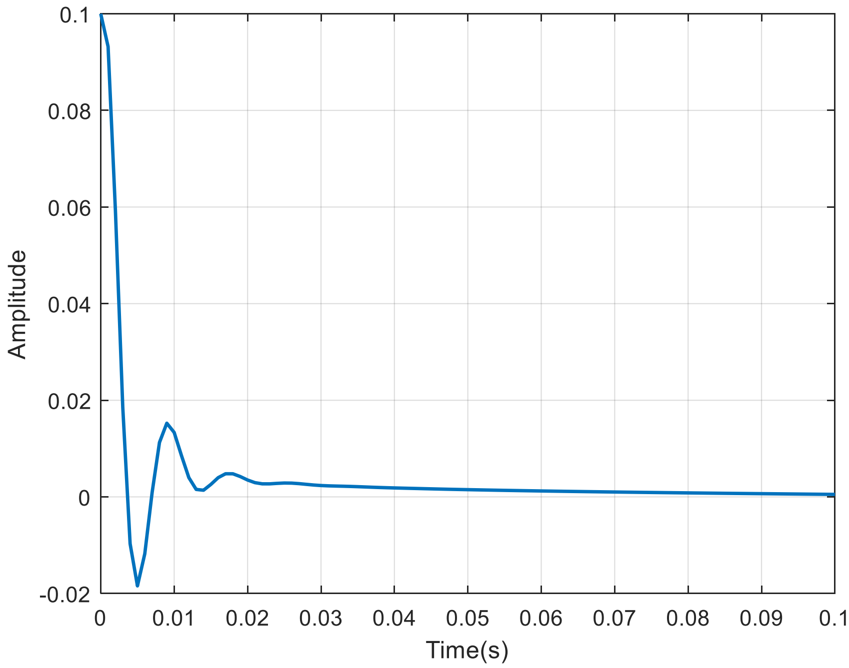

Fig. 5-6: System response with FOPID controller.

5.4. Self-Balancing Robot using FzPD Controller

Both the PI controllers used in the LFC were optimized first with the help of WCA algorithm. WCA algorithm was used with population size of 50, and for 30 iterations. Controller parameters were searched with in the lower bound (LB) and upper bound (UB) of -5 to +5 respectively. Table III gives the other parameters used in WCA optimization.

Fig. 5-2 shows the reduction in cost function value over the algorithm iterations. Best ITAE (objective function) value achieved with WCA algorithm was 0.10015. WCA optimization algorithm took nearly only 7 iterations to achieve the minimum ITAE cost function value.

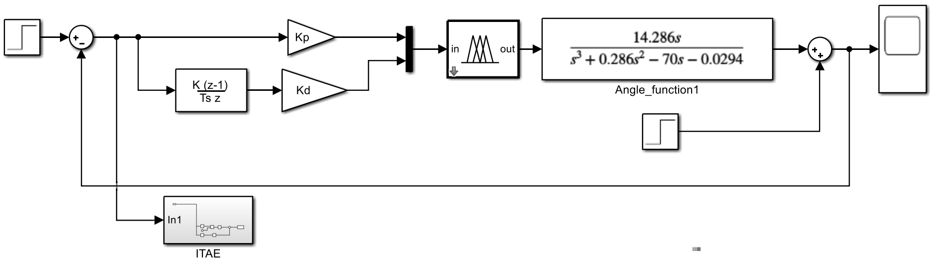

Fig. 5-7: System block diagram with FzPD controller.

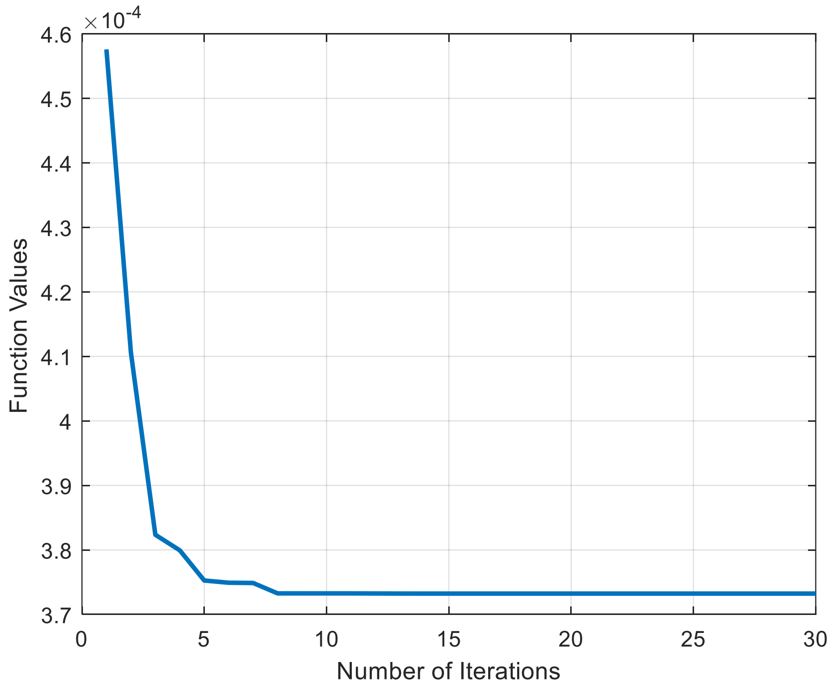

Fig. 5-8: WCA convergence curve for FzPD.

Table III: WCA Parameters for Different Controllers

Parameter Name (WCA)	**PID**	**FOPID**	**FzPID**
No. of Variables	4	5	2
Variables	K_p, K_i, K_d, N	$K_p, K_i, \lambda\ K_d$, μ	K_p, K_d
Lower Bound	0	[0 0 0 0 0]	[0, 0]
Upper Bound	1000	[1000 1000 2 1000 2]	[50, 10]
Population Size	50	50	50
No. of Iterations	30	30	30

Table IV: Optimized Controller Parameters

Parameter Name	**PID**	**FOPID**	**FzPD**
K_p	249.8	500	31.73
K_i	500	456	-
K_d	37.78	36	0.1718
N	500	-	-
λ	-	1.403	-
μ	-	1.08	-
ITAE	0.00047	0.00045	0.00037

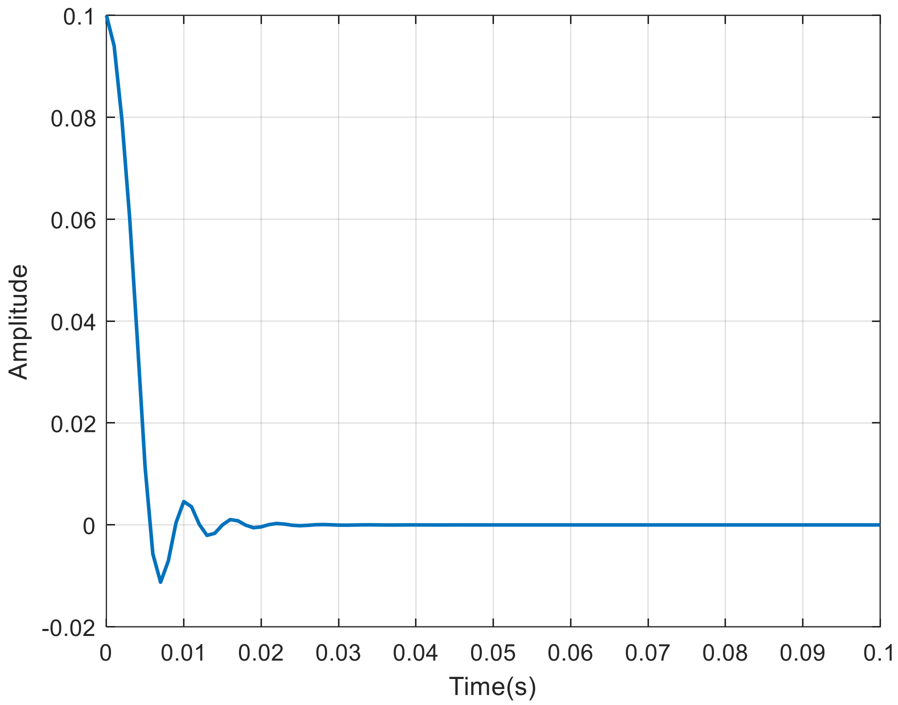

Fig. 5-9: System response with FzPD controller.

5.5. Comparison Between Different Controllers

The results of the comparative analysis of the various controllers employed in this study are presented in this section. The same has been shown by Fig. 5-10. From the Fig. 5-10, it is clear that FzPD controller outperforms the other two controllers in stabilizing the TWSBR within the reasonable amount of time. It is much closer to the balance point better than the rest. The code for the same is mentioned in the appendix.

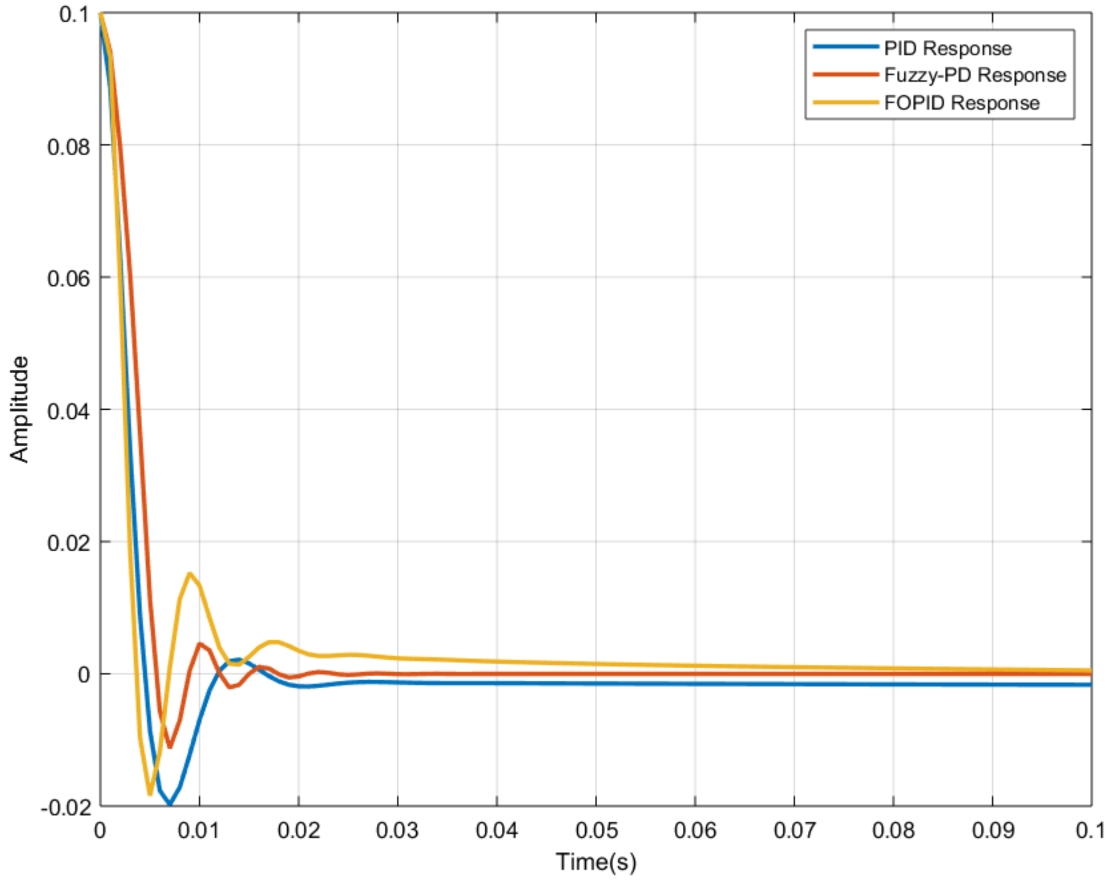

Fig. 5-10: Comparative analysis for different controllers.

Chapter 6
Conclusion & Future Work

CHAPTER 6

CONCLUSION & FUTURE WORK

6.1. Conclusion

In this work, the TWSBR has been designed and simulated with the help of MATLAB 2018. Three different control algorithms, namely PID, FOPID and FzPD were used for controlling the system. WCA optimization algorithm was utilized for tuning the different parameters of the proposed controllers. Comparative study between these controllers shows that the FzPD controller outperforms the other two controllers is stabilizing the self-balancing robot.

6.2. Future Work

In the future, further study will be done to implement the proposed controllers on real hardware.

References

References

[1]. L. Sun & J. Gan "Researching of two-wheeled self-balancing robot base on LQR combined with PID", *2nd International Workshop on Intelligent Systems and Applications*, pp. 1-5, 2010.

[2]. NGM Thao, DH Nghia, NH Phuc, "A PID backstepping controller for two-wheeled self-balancing robot", In *International Forum on Strategic Technology,* pp. 76-81, 2010.

[3]. W. An, Y. Li "Simulation and control of a two-wheeled self-balancing robot", *International Conference on Robotics and Biomimetics (ROBIO),* pp. 456-461, 2013.

[4]. J. Wu & W. Zhang, "Design of fuzzy logic controller for two-wheeled self-balancing robot", *In Proceedings of 2011 6th international forum on strategic technology*, Vol. 2, pp. 1266-1270, 2011.

[5]. J Wu, W Zhang, S Wang, "A two-wheeled self-balancing robot with the fuzzy PD control method" *Mathematical Problems in Engineering,* 2012.

[6]. W. Junfeng & Z. Wanying, "Research on control method of two-wheeled self-balancing robot", *Fourth International Conference on Intelligent Computation Technology and Automation,* Vol. 1, pp. 476-479, 2011.

[7]. J. Wu, Y. Liang & Z. Wang, "A robust control method of two-wheeled self-balancing robot", *In Proceedings of 2011 6th International Forum on Strategic Technology,* vol. 2, pp. 1031-1035, 2011.

[8]. C. Iwendi, M.A Alqarni, J.H. Anajemba, A.S Alfakeeh, Z. Zhang, & A. K. Bashir, "Robust navigational control of a two-wheeled self-balancing robot in a sensed environment", *IEEE Access, 7,* pp. 82337-82348, 2019.

[9]. C. Sun, T. Lu, K. Yuan, "Balance control of two-wheeled self-balancing robot based on Linear Quadratic Regulator and Neural Network", *In 2013 Fourth International Conference on Intelligent Control and Information Processing (ICICIP),* pp. 862-867, 2013.

[10]. O. Jamil, M. Jamil, Y. Ayaz, " Modeling, control of a two-wheeled self-balancing robot", *In 2014 International Conference on Robotics and Emerging Allied Technologies in Engineering (iCREATE)* , pp. 191-199, 2014.

[11]. J. Fang, "The LQR controller design of two-wheeled self-balancing robot based on the particle swarm optimization algorithm", *Mathematical Problems in Engineering*, 2014.

[12]. X. Ruan and J. Cai, "Fuzzy backstepping controllers for two-wheeled self-balancing robot", *In 2009 International Asia Conference on Informatics in Control, Automation and Robotics*, pp. 166-169, 2009.

[13]. H. Bin, L. W. Zhen, L. Hai Feng, "The kinematics model of a two-wheeled self-balancing autonomous mobile robot and its simulation" *In 2010 Second International Conference on Computer Engineering and Applications*, vol. 2, pp. 64-68. 2010.

[14]. R. Sadeghian & M. T. Masoule, "An experimental study on the PID and Fuzzy-PID controllers on a designed two-wheeled self-balancing autonomous robot", *In 2016 4th International Conference on Control, Instrumentation, and Automation (ICCIA)*, pp. 313-318, 2016.

[15]. Q. Yong, L. Yanlong, Z. Xizhe, L. Ji, "Balance control of two-wheeled self-balancing mobile robot based on TS fuzzy model", *In Proceedings of 2011 6th International Forum on Strategic Technology*, vol.1, pp. 406-409, 2011.

[16]. M. M. Azimi; H. R. Koofigar, "Model predictive control for a two wheeled self-balancing robot", *In 2013 First RSI/ISM International Conference on Robotics and Mechatronics (ICRoM)*, pp. 152-157, 2013.

[17]. N. Razmjooy, M. Ramezani, "Optimal control of two-wheeled self-balancing robot with interval uncertainties using Chebyshev inclusion method", *Majlesi Journal of Electrical Engineering*, vol.12, no.1, pp.13-21, 2018.

[18]. P. frankovský, L. dominik, A. gmiterko, I. virgala, P. kurylo, O. perminova, "Modeling of two-wheeled self-balancing robot driven by DC gearmotors",

International Journal of Applied Mechanics and Engineering, vol.22, no.3, pp.739-747, 2017.

[19]. C. Qiu & Y. Huang, "The design of fuzzy adaptive PID controller of two-wheeled self-balancing robot", *International Journal of Information and Electronics Engineering*, vol.5, no.3, pp.193, 2015.

[20]. B. Mahler & J. Haase, "Mathematical model and control strategy of a two-wheeled self-balancing robot", *In IECON 2013-39th Annual Conference of the IEEE Industrial Electronics Society*, pp. 4198-4203, 2013.

[21]. T. A. Mai, DN. Anisimov, TS. Dang, V. N. Dinh, "Development of a microcontroller-based adaptive fuzzy controller for a two-wheeled self-balancing robot", *Microsystem Technologies*, vol.24, no.9, pp.3677-3687, 2018.

[22]. O. S. Bhatti, K. M. Hasan, M. A. Imtiaz, "Attitude control and stabilization of a two-wheeled self-balancing robot", *Journal of Control Engineering and Applied Informatics*, vol.17, no.3, pp.98-104, 2015.

[23]. T. Zhao, Q. Yu, S. Dian, R. Guo, S. Li, "Non-singleton general type-2 fuzzy control for a two-wheeled self-balancing robot", *International Journal of Fuzzy Systems*, vol.21, no.6, pp.1724-1737, 2019.

[24]. A. Ghaffari, A. Shariati, A.H. Shamekhi, "A modified dynamical formulation for two-wheeled self-balancing robots", *Nonlinear Dynamics*, vol.*83*, no.1, pp.217-230, 2016.

[25]. F Sun, Z Yu, H Yang, "A design for two-wheeled self-balancing robot based on Kalman filter and LQR," *In 2014 International Conference on Mechatronics and Control (ICMC)*, pp. 612-616, 2014.

[26]. H. M. Omar, A. M. Elalawy & H. H. Ammar, "Two-wheeled self-balancing robot modelling and control using artificial neural networks," *In 2019 Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, vol. 1, pp. 196-200, 2019.

[27]. M. Baloh & M. Parent, "Modelling and model verification of an intelligent self-balancing two-wheeled vehicle for an autonomous urban transportation system", *In*

The conference on computational intelligence, robotics, and autonomous systems, pp. 1-7, 2003.

[28]. A. I. Glushchenko, K. A. Lastochkin & V. A. Petrov, "Development of Two-Wheeled Balancing Robot Optimal Control System based on Its Feedback Linearization", *In 2019 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon)*, pp. 1-6, 2019.

[29]. RPM. Chan, KA. Stol, CR. Halkyard, "Review of modelling and control of two-wheeled robots", *Annual reviews in control*, vol.37, no.1, pp. 89-103, 2013.

[30]. N. Razmjooy & M. Ramezani, "Optimal control of two-wheeled self-balancing robot with interval uncertainties using Chebyshev inclusion method", *Majlesi Journal of Electrical Engineering*, vol.12, no.1, pp.13-21, 2018.

[31]. X. Wang, S. Chen, T. Chen, B. Yang, "Study on control design of a two-wheeled self-balancing robot based on ADRC", *In 2016 35th Chinese Control Conference*, pp. 6227-6232, 2016.

[32]. NGM. Thao, DH. Nghia, NH. Phuc, "A PID backstepping controller for two-wheeled self-balancing robot", In *International Forum on Strategic Technology 2010*, pp. 76-81, 2010.

[33]. C. Qiu & Y. Huang, "The design of fuzzy adaptive PID controller of two-wheeled self-balancing robot", *International Journal of Information and Electronics Engineering*, vol.*5, no.*3, pp.193, 2015.

[34]. L. Sun & J. Gan, "Researching of two-wheeled self-balancing robot base on LQR combined with PID", In *2010 2nd International Workshop on Intelligent Systems and Applications*, pp. 1-5, 2010.

[35]. C Qiu, Y Huang, "The design of fuzzy adaptive PID controller of two-wheeled self-balancing robot", *International Journal of Information and Electronics Engineering*, vol.*5, no.*3, pp.193,2015.

[36]. R. Sadeghian, MT. Masoule, "An experimental study on the PID and Fuzzy-PID controllers on a designed two-wheeled self-balancing autonomous robot", In *2016*

4th International Conference on Control, Instrumentation, and Automation (ICCIA), pp. 313-318, 2016.

[37]. J. Wu & W. Zhang, “Design of fuzzy logic controller for two-wheeled self-balancing robot”, *In Proceedings of 2011 6th international forum on strategic technology*, vol. 2, pp. 1266-1270, 2011.

[38]. J. Wu, W. Zhang, S. Wang, “A two-wheeled self-balancing robot with the fuzzy PD control method”, *Mathematical Problems in Engineering*, 2012.

[39]. X. Ruan & J. Cai, “Fuzzy backstepping controllers for two-wheeled self-balancing robot”, *In 2009 International Asia Conference on Informatics in Control, Automation and Robotics*, pp. 166-169, 2019.

[40]. ANK. Nasir, MA. Ahmad, R. Ghazali, N. S. Pakheri, “Performance comparison between fuzzy logic controller (FLC) and PID controller for a highly nonlinear two-wheels balancing robot”, *In 2011 First International Conference on Informatics and Computational Intelligence*, pp. 176-181, 2011.

[41]. H Eskandar, A Sadollah, A Bahreininejad & M. Hamdi, “Water cycle algorithm–A novel metaheuristic optimization method for solving constrained engineering optimization problems”, *Computers & Structures*, vol.*110*, pp.151-166, 2012.

[42]. A. Sadollah, H. Eskandar, J.H.Kim, “Water cycle algorithm for solving constrained multi-objective optimization problems”, *Applied Soft Computing*, vol.27, pp.279-298, 2015.

[43]. A. Sadollah, H. Eskandar, A. Bahreininejad, J.H. Kim, “Water cycle algorithm for solving multi-objective optimization problems”, *Soft Computing*, vol.19, no.9, pp.2587-2603, 2015.

APPENDIX

APPENDIX

A. MATLAB Code for ITAE Objective Function

```
function J = itae_cost(x)

%% Assign different variable values from input vector x

% Move variables into model parameter names

 Kp = x(1);

Ki = x(2);

Kd = x(3);

N = x(4);

 % Choose solver and set model workspace to this function

opt = simset('solver','SrcWorkspace');

try

% Simulate the model

sim('system_model_vary',[0 0.1],opt);

% Assign computer ITAE value to Function output

J = ITAE.Data(end);

catch

% in case of any error assign large value

J = 1e15;

end
```

B. MATLAB Code for Comparative analysis

```
% this file compares the results
clc; clear all; close all;
warning off;

% RUN PID code
load('wca_res.mat');
x = x_res;
Kp = x(1);      % Move variables into model parameter names
Ki = x(2);
Kd = x(3);
N = x(4);

% Choose solver and set model workspace to this function
opt = simset('solver','ode4','SrcWorkspace','Current');

sim('system_model_vary',[0 0.1],opt);

plot(PID_data(:,1),PID_data(:,2),'Linewidth',1.5);
grid on;
xlabel('Time(s)');
ylabel('Amplitude');
hold on;

error.pid = ITAE.Data(end);
%% run fuzzy file
load('wca_res_fuzzy_30.mat');
x = x_res;
Kp = x(1);      % Move variables into model parameter names
Ki = x(2);
Kd = x(3);

% Choose solver and set model workspace to this function
```

```
opt = simset('solver','ode4','SrcWorkspace','Current');

sim('system_model_fuzzy_vary',[0 0.1],opt);

plot(Fuzzy_PD_data(:,1),Fuzzy_PD_data(:,2),'Linewidth',1.5);
grid on;
xlabel('Time(s)'); ylabel('Amplitude');
%% run FOPID file
load('wca_res_FOPID_30.mat');
x = x_res;
Kp = x(1);      % Move variables into model parameter names
Ki = x(2);
lm = x(3);
Kd = x(4);
mu = x(5);

% Choose solver and set model workspace to this function
opt = simset('solver','ode4','SrcWorkspace','Current');

sim('system_model_FOPID_comp',[0 0.1],opt);


plot(FOPID_data(:,1),FOPID_data(:,2),'Linewidth',1.5);
xlabel('Time(s)'); ylabel('Amplitude');
legend('PID Response','Fuzzy-PD Response', 'FOPID Response');
```