CONTROLLING AN-INVERTED PENDULUM SYSTEM USING A MICROCONTROLLER

BY:

AHMAD AMIRUL BIN MOHAMAD ZAHARI

(Matrix no.:133925)

Supervisor:

Prof. Dr. Norzalilah Binti Mohamad Nor

This dissertation is submitted to

Universiti Sains Malaysia

As partial fulfilment of the requirement to graduate with honors degree in

BACHELOR OF ENGINEERING (MANUFACTURING ENGINEERING)

School of Mechanical Engineering

Engineering Campus

Universiti Sains Malaysia

24 July 2022

DECLARATION

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed………………………………………………… (Ahmad Amirul B. Mohd Zahari)

Date…………………………………………………………………….

STATEMENT 1

This thesis is the result of my own investigations, except where otherwise stated.

Other sources are acknowledged by giving explicit references.

Bibliography/references are appended.

Signed………………………………………………… (Ahmad Amirul B. Mohd Zahari)

Date…………………………………………………………………….

STATEMENT 2

I hereby give consent for my thesis, if accepted, to be available for photocopying and for interlibrary loan, and for the title and summary to be made available outside organizations.

Signed………………………………………………… (Ahmad Amirul B. Mohd Zahari)

Date…………………………………………………………………….

ACKNOWLEDGEMENT

First and foremost, I would like to thank the School of Mechanical Engineering at Universiti Sains Malaysia for providing the essential equipment and supplies to finish my Final Year Project (FYP). I am grateful for the wonderful chance to complete this four-year study in Mechanical Engineering, and I am confident that all of the information and technical skills I have gained will be extremely useful in my future work as an engineer. In addition, I'd like to thank my supervisor, Prof. Dr. Norzalilah Binti Mohamad Nor, for supervising and supporting my study. My supervisor's assistance and support have always encouraged and aided me in conquering problems. Furthermore, I would like to express my gratitude to the employees of the 3D printing Lab of the School of Mechanical Engineering, Universiti Sains Malaysia (USM), particularly Mr. Norijas Abd Aziz, who gave insight and skills that substantially aided the research. A special thanks also goes to the course coordinator, Dr. Muhammad Fauzinizam Bin Razali, who has put in countless hours of seminar lectures to assist us reach our objective. Last but not least, I must express my gratitude for the support and advice provided by other supervisors as well as the panels, particularly during the project presentation.

I owe my parents a debt of gratitude that will last forever since they have been the cornerstones in my life, providing both material and emotional support. I've been able to pursue my goals and do what I enjoy in life because to the freedom they've provided me. Last but not least, I want to express my gratitude to the rest of my family and friends for supporting me during this journey. Through the highs and lows of this endeavour and life in general, they have supported and placed their confidence in me. I appreciate all of your support.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATION

LQR	Linear-Quadratic Regulator
PID	Proportional Integral Differential
SMC	Sliding Mode Controller
FLC	Fuzzy Logic Controller
TWSB	Two Wheeled Self-Balancing
K_p	Proportional Control
K_i	Integral Control
K_d	Derivative Control
UAV	Unmanned Air Vehicles
FIS	Fuzzy Interference System
MIMO	Multiple-Input-Multiple-Output
SISO	Single-Input-Single-Output
I^2C	Integrated Circuit Communication
DOF	Degree of Freedom
IMU	Inertia Measurement Unit
MCU	Microcontroller Unit
LiPo	Lithium Polymer Battery
PWM	Pulse-Width Modulation
MATLAB	Matrix Laboratory
SCL	Serial Clock
SDA	Serial Data
KF	Kalman Filter
CF	Complementary Filter

ABSTRAK

Robot pengimbangan diri pada dasarnya adalah pendulum terbalik. Ia boleh mengimbangi lebih baik jika pusat jisim lebih tinggi daripada gandar roda.Pusat jisim yang lebih besar sama dengan momen inersia yang lebih tinggi, yang sama dengan pecutan sudut yang lebih rendah. Terutamanya semasa pergerakan, robot TWSB yang dilaksanakan dengan baik dapat mengekalkan pendirian tegak. Sebilangan besar kertas memberi tumpuan kepada sama ada membuat pengawal melalui pelaksanaan unit mikropengawal peringkat rendah, seperti Arduino Uno, atau pada ciri pemodelan dinamik di mana penemuan simulasi digunakan untuk menentukan hasil dan bukannya pendirian aplikasi dunia nyata. Kajian ini akan menumpukan pada membandingkan hasil simulasi dengan pemasangan sebenar robot TWSB kerana lebih sedikit penyelidik telah melakukannya. Projek ini berhasrat untuk mengkaji prestasi robot TWSB yang dihasilkan, mengkaji kebolehgunaan MATLAB dengan pengaturcaraan robot TWSB, dan membandingkan prestasi robot TWSB dengan hasil simulasi daripada MATLAB. Pada masa yang sama, perbandingan dibuat antara projek ini dan kerja terdahulu untuk menilai kelebihan dan kekurangan masing-masing. Dalam contoh ini, robot TWSB dibina menggunakan mikropengawal Arduino UNO dan pengawal algoritma PID.Giroskop MPU 6050 ditentukur sebelum dipasang pada robot untuk memaksimumkan ketepatan keputusan yang diperolehi dengan menentukan nilai ofset. MATLAB digunakan untuk mewujudkan nilai terma kawalan yang sesuai untuk pengawal PID untuk menggantikan prosedur penalaan manusia dan memudahkan penstabilan robot TWSB.Mengikut keputusan, nilai istilah kawalan Kp = 64, Ki = 45, dan Kd = 1.3 adalah mencukupi untuk mengekalkan postur robot TWSB, membolehkan ia mengekalkan kestabilan dalam pelbagai permukaan, termasuk permukaan rata dan tidak rata, dengan atau tanpa penggunaan kuasa dan halangan.

ABSTRACT

A self-balancing robot is basically an inverted pendulum. It can balance better if the centre of mass is higher than the wheel axels. A greater centre of mass equals a higher moment of inertia, which equals a lower angular acceleration. Particularly during movement, a well-implemented TWSB robot is able to maintain an upright stance. The majority of papers focus on either creating controllers through the implementation of low-level microcontroller units, such as Arduino Uno, or on dynamic modelling features in which simulation findings are used to decide results rather than real-world applications. This study will concentrate on comparing simulation results to the actual installation of a TWSB robot since fewer researchers have done so. This project intends to study the performance of the produced TWSB robot, examine the applicability of MATLAB to the programming of the TWSB robot, and compare the performance of the TWSB robot to the simulation results from MATLAB. Concurrently, a comparison is made between the present project and earlier work to assess the advantages and disadvantages of each. In this instance, a TWSB robot is constructed utilising an Arduino UNO microcontroller and a PID algorithm controller. The MPU 6050 gyroscope is calibrated before being mounted to the robot in order to maximise the accuracy of the acquired results by determining offset values. MATLAB is used to establish the appropriate control term values for the PID controller in order to replace the human tuning procedure and facilitate the stabilisation of the TWSB robot. According to the results, control term values of Kp = 64, Ki = 45, and Kd = 1.3 are adequate to maintain the posture of the TWSB robot, enabling it to maintain stability on a variety of surfaces, including flat and uneven surfaces, with or without the application of forces and obstructions.

Chapter 1 INTRODUCTION

1.1 Research Background

Without external control, an inherently self-balancing robot is unstable and would roll around the rotation axis of the wheels, finally falling. If motor driving happens in the correct direction, the robot returns to its original location. Although the robot is intrinsically unstable, it offers several advantages over statically stable multi-wheeled robots. The robot must be built on a unique electromechanical mechanism that balances itself on a pair of wheels while standing tall[1]. If the platform on which the robot stands is unstable or unbalanced, the robot hasa tendency to slip off its vertical axis. Due to the multiple variables, high order, nonlinearity, strong coupling, and instability of a two-wheeled self-balancing robot system, it is a standard research target for many modern control theory researchers and a varietyof abstract control principles, such as system stability, robustness, controllability, and anti- interference qualities, may be shown by experiments with a two-wheeled self-balancing robot system.

With the right amount of friction, the self-balancing robot is able to maintain its equilibrium on two wheels. Both the inclination angle and the movement of the motors to maintain the vertical axis must be measured and controlled in order to maintain a constant 0° angle[2]. Principle of an inverted pendulum may be used in a controlling task with the goal of balancing the pendulum without the need for previous understanding of dynamics. Because of its inherent instability, balancing an inverted pendulum is a difficult task. System instability increases as a result of even the tiniest inaccuracy or disruption from the equilibrium state. The fine control required to maintain balance at a little non-equilibrium demands fast correction of any tilt faults that occur. In comparison to four-wheeled robots, two-wheeled robots have a higher degree of agility and mobility owing to their capacity to make sharp turns and negotiate tight curves with more ease than four-wheeled robots[3].

Small carts constructed utilizing an inverted pendulum system may be adopted in industries with limited space to minimize the requirement for vehicles, which are more polluting, and to free up room for other applications. In recent years, the notion of an inverted pendulum mechanism has been extensively used to personal transporters such as David Kamen's 2001 Segway[4]. This robot is widely used by security officers at shopping malls to cover large areas. iBot, a mobile wheelchair designed by David Kamen in 2006, is capable of climbing stairs and supports disabled individuals by raising their height once balance on wheels is attained. In conclusion, the implementation of the TWSB robot is advantageous since it can address problems in both industry and society.

Even though the architecture of the TWSB robot is basic, requiring just a pair of wheels and motors, there are problems in constructing a two-wheeled self-balancing robot. The key obstacles that academics are interested in are robot balance issues, since robots must maintain an upright stance in both rest and movement postures[5]. In real-world applications, inclination planes and uneven floor surfaces need quick responses from robots in order to maintain stability.

1.2 Problem Statement

The challenge of this project is to keep the Two-wheeled self-balancing robot maintaining its upright posture during movement or when the disturbance approaches the robot. Most of the researched done are mostly focused on the simulation and dynamic modelling aspect of the robot. In this study, we will focus on the design and construction aspect of TWSB robot using CAD. Since less researcher focused on the weight distribution aspect when constructing TWSB robot. In addition, whereas the vast majority of research studies are done solely on flat surfaces, the robot will be tested on uneven surfaces such as off-road.

1.3 Objective

- To develop a mathematical model to control inverted pendulum with Arduino UNO.
- To simulate and fabricate TWSB robot
- To design the TWSB robot part assembly
- To analyze and compare the performance of a self-balancing robot to that of prior research.

1.4 Scope of Work

This project covers design, construction, experimentation, and result analysis. The primary emphasis of this research will be on the problems associated with constructing a two wheeled robot utilizing an Arduino UNO with PID controller and design of the robot. Due to the fact that earlierwork has been established, the results of previous work will be compared to the present results,and potential improvements will be explored. In summary, this research will concentrate on optimizing the performance of the two wheeled robots from prior work.

Chapter 2 LITERATURE REVIEW

2.1 Overview

The inspiration for this form of vehicle originates from the simple pendulum, in which a weight is suspended by a rope. When a pendulum is presented asymmetrically from its equilibrium position, gravity will cause it to accelerate back towards balance. When the pendulum mass is released, the force of gravity will drive it to oscillate back and forth until friction causes it to move. Inverting this setup and replacing the rope with a solid rod results in an inverted pendulum. However, a control system may be used to maintain the position of an inverted pendulum by monitoring the pole's angle and restoring the pivot point to the center of mass if the position becomes unstable due to environmental or human variables[6]. According to Chakraborty (2014), the concept of an inverted pendulum system may be used to a variety of modern technologies, such as satellites in space, missile guidance, biomechanics, and statistical applications[7].

Inverted pendulum systems are typically classified into two classes: static inverted pendulum and dynamic inverted pendulum. Typically, a wheeled mobile robot is coupled with an inverted pendulum installed on a two-wheeled cart[8]. This is also known as the notion for a robot with two wheels. The definition of wheeled mobile robot is a wheeled vehicle with autonomous movement capacity. The use of wheeled mobile robots has risen considerably over time in both the industrial and service sectors. In this project, a TWSB robot will be built and operated by a microcontroller chosen to guarantee its stability.

2.2 Robot Structure

The fundamental construction of the TWSB robot consists of a chassis part and a wheel’s section. The motors are coupled to the chassis (primary body), and the main construction is simplified for modelling purposes. First created by Felix Grasser, Silvio Colombi, Aldo D'arrigo, and Alfred Rufer, the two-wheeled mobile robot JOE is based on the notion of an inverted pendulum system that served as a reference for the community to further develop and refine the inverted pendulum cart[9]. As a consequence, this kind of inverted pendulum cart is extensively produced and used to facilitate human life. For instance, the Segway Human Transporter is being developed as a mobility solution to replace internal combustion engine cars.

Basically, the microcontroller unit, power supply, motor driver, gyro sensor, and DC geared motor used to construct a TWSB robot are essentially identical, based on observations gleaned from academic articles[10]. All explored two-wheeled robots have a similar configuration, with two wheels positioned coaxially and a platform consisting of a steering bar situated above. Likewise, the two-wheeled robot's power supply, which is typically a DC power source, will be situated atop the assembly. This is because placing the bulk above the center of gravity will boost the robot's stability.

2.3 Type of Controller System Design

Balancing an inverted pendulum is a difficult task since it is inherently unstable. The slightest inaccuracy or deviation from the equilibrium state pulls the system violently out of balance, further destabilizing the system[11]. Thus, maintaining balance in a slightly out-of-equilibrium state needs precision control to promptly rectify any tilt faults that occur. Microcontrollers such as the Proportional Fuzzy Logic Controller (FLC), the Integrated Derivative Controller (PID), the Sliding Mode Controller (SMC), and the Linear-Quadratic Regulator (LQR) are often available for implementation in two-wheeled self-balancing robots (LQR). Thus, the selection of a controller to maintain upright position independent of the need for external force is one of the most crucial considerations involved in maintaining the upright position of a TWSB robot.

2.3.1 PID Controller

PID controller utilises tilt feedback to manage the motors' torque and maintain the robot's balance. A PID controller continually monitors a process variable and calculates an error value, which represents the process variable's departure from a predetermined value[12]. The PID controller tries to reduce this type of error over time by constantly modifying a control variable according to the equation: u(t) is the control variable, e(t) is the current error in the process variable, and Kp, Ki, and Kd are tuning coefficients[13]. PID controller is one of the most often used controllers for two-wheeled robots owing to its ability to regulate unstable systems, simplicity of implementation, and removal of the need for a state space model. PID controller served as the real-time controller for the TWSB robot because it can take appropriate corrective action to minimise the difference between the target value and the actual value by adjusting the output[14].

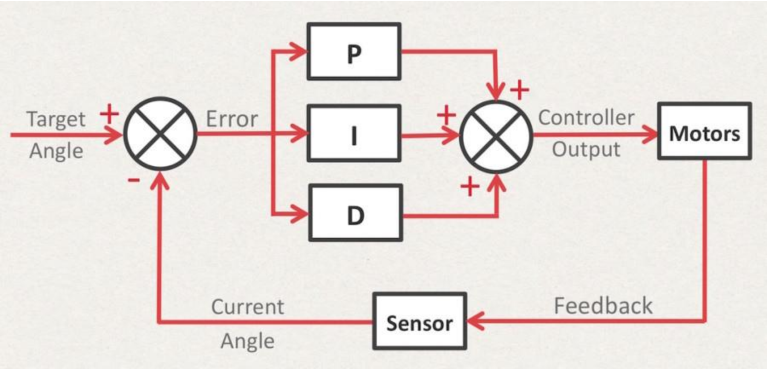

Figure 2.1: PID control system[36]

2.3.2 Fuzzy Logic Controller (FLC)

A fuzzy logic controller is a controller that determines a path of action using fuzzy logic. The control action is computed in four stages, scaling and shifting of the input, fuzzification, fuzzy inference, and defuzzification[15]. Crisp inputs may be processed and scaled properly at the input scaling and shifting stage. The fuzzification stage converts the crisp inputs to fuzzy values. A correct control action is found using fuzzy inference by looking through a previously created rule. Finally, the defuzzification process defines the deterministic output. The fuzzy controller receives inputs from the robot's position and speed, while the PD controller receives inputs from the robot's angle and angle rate[16].

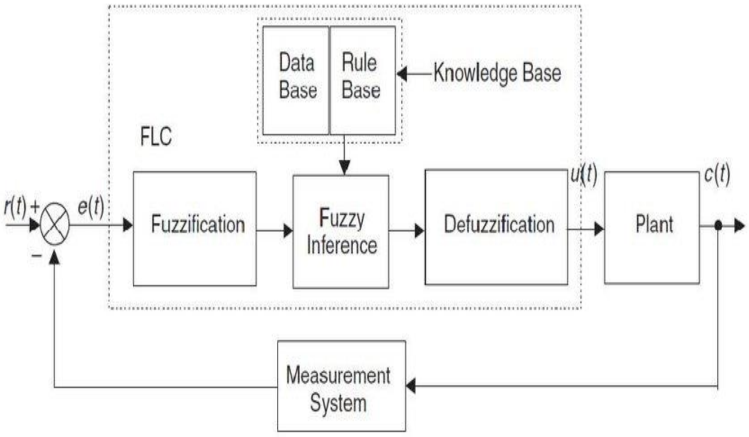

Figure 2.2: Fuzzy logic control system[37]

2.3.3 Linear-Quadratic Regulator Controller (LQR)

LQR is a technique in current control theory that analyses such a system using a state space approach. It is quite straightforward to deal with a multi-output system using state-space approaches. Full state feedback may be used to stabilise the system. When LQR is implemented, a substantial quantity of data gathered from experimental methods is necessary for network testing and training[17]. This problem limits the deployment of LQR controllers, despite the fact that this form of controller may include intelligence and automation elements into the system being created[18]. During the deployment of LQR, it is necessary to gather a considerable quantity of experimental data for use in network testing and training. This problem limits the implementation of LQR, despite the fact that this form of controller may give intelligent and automated system characteristics.

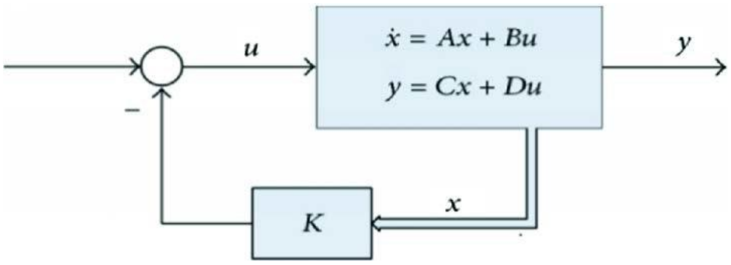

Figure 2.3: LQR control system[38]

2.3.4 Sliding Mode Controller (SMC)

The sliding mode controller (SMC) is made up of two components, the nominal sections and some extra terms that have the role of coping with uncertainty in the model. SMC is useful in that it allows you to drive a trajectory into a sliding surface while keeping the error trajectory constant over time. SMC has the benefits of being sensitive in terms of settling time and overshoot, which allows the pendulum to keep its upright position. Furthermore, SMC can operate without the need for linearization and can maintain performance even when a point is a significant distance away from the equilibrium location[19]. Based on the results of the simulation, SMC controllers are regarded superior than LQR controllers because SMC is more responsive and can accomplish desirable features in a shorter amount of time when compared to LQR.

2.4 Combination of The Control System

In certain implementations, combining two or more control systems may be necessary for the robot to function properly. For the whole system to work, the control system implementation is essential.

2.4.1 LQR-PID Controller

On the basis of the system structure model, the kinetic equation is constructed using the Lagrange technique, and the linearizing model in the region of the balance is obtained. The control strategy of integrating LQR and PID successfully overcomes the effects of system restrictions. The control algorithm that combines LQR and PID can achieve a strong balancing effect and has excellent anti-disturbance effects, and can quickly recover the dynamic[20].

For the PID controller, it is possible to manage the robot's balance, but the robot vibrates more near the balance point, resulting in poor static performance. As for the linear controller LQR, it has a good control effect within a small-angle range, but when the disturbance angle exceeds the linearization constraint requirements, the controller does not have a strong control effect and cannot even maintain robot balance.

2.4.2 Fuzzy-PID Controller

Fuzzy logics are used to tweak the suggested PID controller so that it can compute a high-performance controller. The Fuzzy-PID controller was utilised to lessen the influence of external pressures on the robot's balance-ability. When the Fuzzy-PID controller was implemented on the suggested robot, its stability and balance were enhanced, and the influence of external force was greatly reduced. The tuning criteria for PID gains are defined prior to their incorporation into a fuzzy system that adjusts PID gains based on theoretical and experimental study[19].

Self-tuning Fuzzy PID controllers can real-time optimise PID gains by adjusting controlling parameters. This is because fuzzy PID controllers may decrease overshoot while requiring less settling time than traditional PID controllers. Therefore, a real-world implementation of fuzzy PID controller is necessary in order to compare experimental technique control performance.

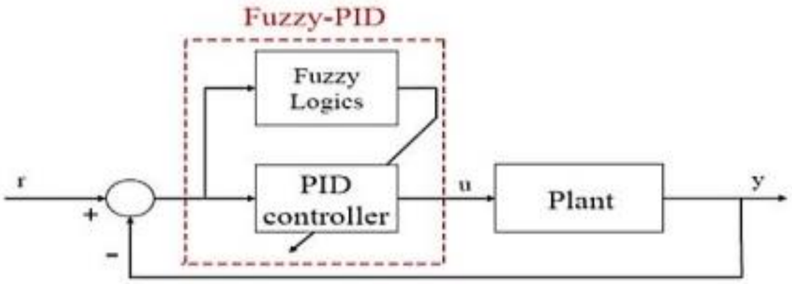

Figure 2.4: Fuzzy-PID block diagram[39]

2.5 Control System Implementation

2.5.1 Fuzzy Interference Control System

The primary focus is on tracking and control challenges for the XFS three-decoupled displacement flyer engine. The capability of XFS to fly appropriately in arc, straight, or even complicated trajectories is discussed. The implementation of an optimization approach enables online modification of all fuzzy controller settings. In flying robots, the On-line Optimize

Takagi-Sugeno Fuzzy Interference System (FIS) in zero order is a model that is simple, safe, and cost-effective[21]. Thus, FIS facilitates an approach for optimising that is both rapid and efficient. Observed results indicate that the descending gradient technique may adapt to unknown settings using an optimization strategy. In addition, fuzzy logic may be utilised to regulate the new configuration of the XFS engine due to its capacity for modelling empirical information. Nonetheless, real-time STFIS implementation is not commonly available. Future study can concentrate on overcoming barriers.

2.5.2 Robust H-Infinity Control System

Unmanned Air Vehicles, often known as UAVs, are described as aircraft that are flown using an on-board computer or a remote control. It is widely used because it may lower costs and risks that are related to human life, such as those associated with firefighting management. In the meanwhile, it is thought of as a MIMO system, which requires a vital selection procedure to pick the appropriate control system. In this scenario, the robust approach is used in the design of controllers in order to accomplish the goal of achieving robust performance[22].

The stability of an unmanned aerial vehicle (UAV) may be maintained even if the system deviates from its specified nominal design scenario by making use of resilient techniques. By using a Robust H Control System, it is possible to lessen the sensitivity at low frequencies, while simultaneously increasing the uniform gain and achieving the peak of response throughout the spectrum at high frequencies[23]. In conclusion, it has been shown that the Robust H Control Technique can follow a variety of actions even when there is noise present.

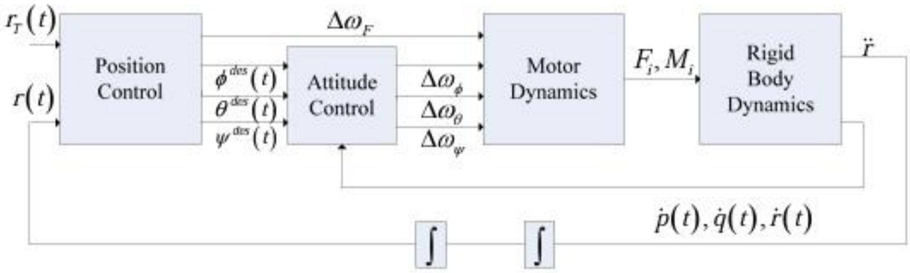

Figure 2.5: A general attitude control architecture for UAV[40]

2.6 Sensor Filter Algorithm

Sensor fusion operates by accepting sensor measurement data as input and thus compensating data gained via the use of digital filtering methods. The 6-axis MEMS sensor MPU6050 is often used in TWSB robot construction to assess the robot's posture[24]. Such a sensor often incorporates a 3-axis gyroscope and a 3-axis accelerometer simultaneously.

2.6.1 Kalman Filter Algorithm

The Kalman filter is an algorithm that utilises a sequence of data collected over time that includes noise and other imperfections to more accurately predict unknown variables. R. E. Kalman made the suggestion in 1960[25]. With the rising needs of target tracking, Kalman filter, a minimum-variance estimator for dynamic systems, has gained significant attention.

In the last thirty years, a number of Kalman filter methods have been introduced for obtaining optimum state estimate. The observation equation must be linear for the Kalman filter to work, which is only suited for linear systems. Since nonlinear systems are used in the majority of real-world applications, nonlinear filter research is crucial[26].

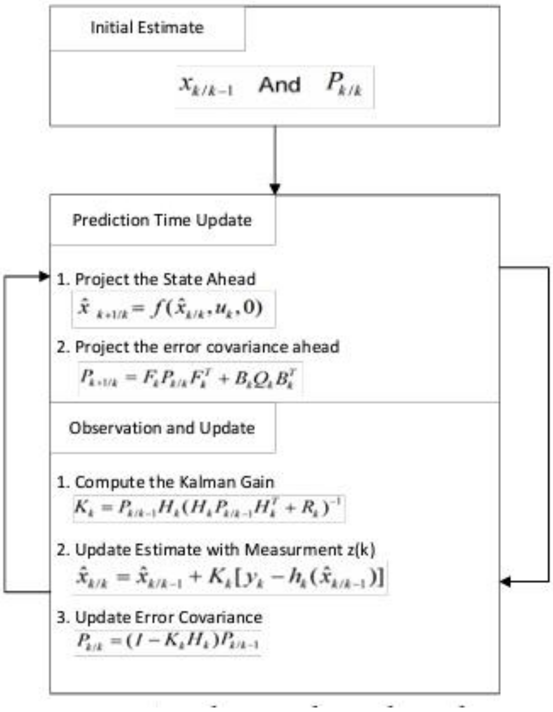

Figure 2.6: Kalman filter algorithm[41]

2.6.2 Complementary Filter Algorithm (CF)

The data from various sensors may be combined using complimentary filters to boost the system's dynamic performance and measurement precision. Since complementary filtering techniques need little computing, great dependability, and little precision from inertial equipment, they are often utilised in aircraft attitude estimation[27]. CF method is regarded as one of the sensor fusions since it can combine accelerometer and gyroscope data to provide very accurate dynamic attitude results. The time constant and the filtering efficiency of CF itself determine its performance[28]. Typically, pass filter time constants are used to fine-tune performance by finding the filter coefficient of the operation.

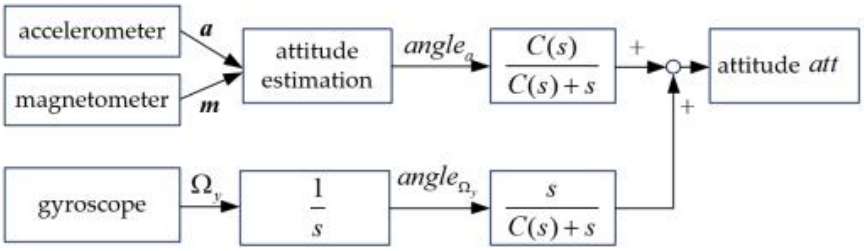

Figure 2.7: Principle of classic complementary filtering[42]

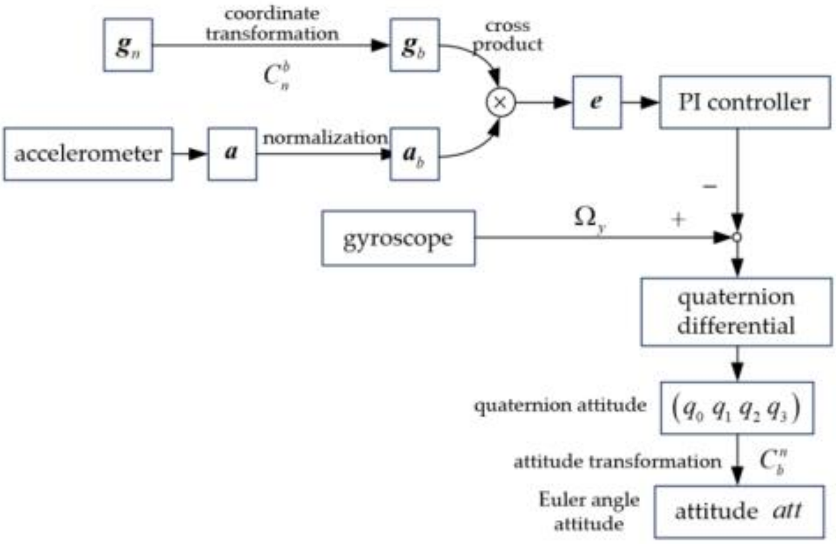

Figure 2.8: Mahony complementary filtering algorithm[42]

2.7 Software

Software is a collection of instructions, data, or computer programmes that are used to run machines and carry out certain activities. It is the antithesis of hardware, which refers to a computer's external components. A device's running programmes, scripts, and applications are collectively referred to as "software" in this context[29].

2.7.1 MATLAB and Simulink Sinscape

MATLAB is an interactive system in which the fundamental data element is a dimensionless array. This enables you to solve many technical computing issues, particularly those using matrix and vector formulations, in a fraction of the time required to build a programme in a scalar noninteractive language such as C or Fortran. This is due to MATLAB's ability to interact with various microcontroller devices such as the Raspberry Pi and Arduino[30]. Simulink is a block diagram environment for designing multidomain models, simulating before transferring to hardware, and deploying without writing code. Simulink's common uses include digital signal processing and control theory in multi-domain simulation or design.

2.7.2 I^{2}C Interface

Philips laboratories in Eindhoven, Netherlands, create Inter-integrated Circuit (I^{2}C), which requires just two wires to link all peripherals to the microcontroller unit. I^{2}C consists of two distinct signal cables called SCL and SDA, as well as a ground connection. IC will commence data transmission on the master bus, while other ICs are classed as bus slaves, per I^{2}C specifications[31]. Through the deployment of I^{2}C bus, master devices may concurrently write and follow bus lines. I^{2}C bus advantages include its ability to accommodate numerous masters and slaves with less hardware complexity than UARTs. I^{2}C bus has a slower data transmission rate than SPI, a smaller data frame size (8 bits), and more complicated hardware implementation requirements than SPI[32].

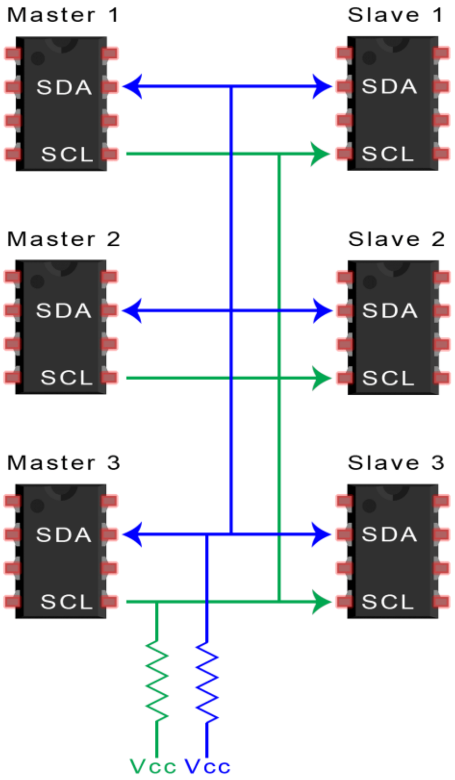

Figure 2.9: Basic of I^2C communication protocol[43]

Chapter 3 METHODOLOGY

3.1 Overview

This chapter provides an overview of every component needed to make a robot, as well as the structure of the robot's wire connections between its various parts using a microcontroller. The use of MATLAB and Simulink Simscape in software implementation to get appropriate robot gain is then described. To offer an example of the behaviour of the TWSB robot, an animation will be produced using Simscape. In addition, programming of the robot is decided, starting with basic component programming, moving to Arduino UNO, and finally achieving an appropriate gain to regulate the pivot point of the robot against the centre of gravity to enable self-balancing.

To offer an example of the behaviour of the TWSB robot, an animation will be produced using Simscape which directly integrate with Simulink. In addition, programming of the robot is decided, starting with basic component programming, moving to Arduino UNO, and finally achieving an appropriate gain to regulate the pivot point of the robot against the centre of gravity to enable self-balancing. The system modelling approach used to maintain the stability of the inverted pendulum system utilising PID controllers is the first step in the development of the TWSB robot. The model will next go through stimulation procedures in order to be analysed under imitation processes of real-world situations. MATLAB software will also be used to do control logarithm at the same time. The last step of creating an inverted pendulum system involves testing and refining the robot until the desired outcome is achieved.

3.1.1 Process Flow Chart

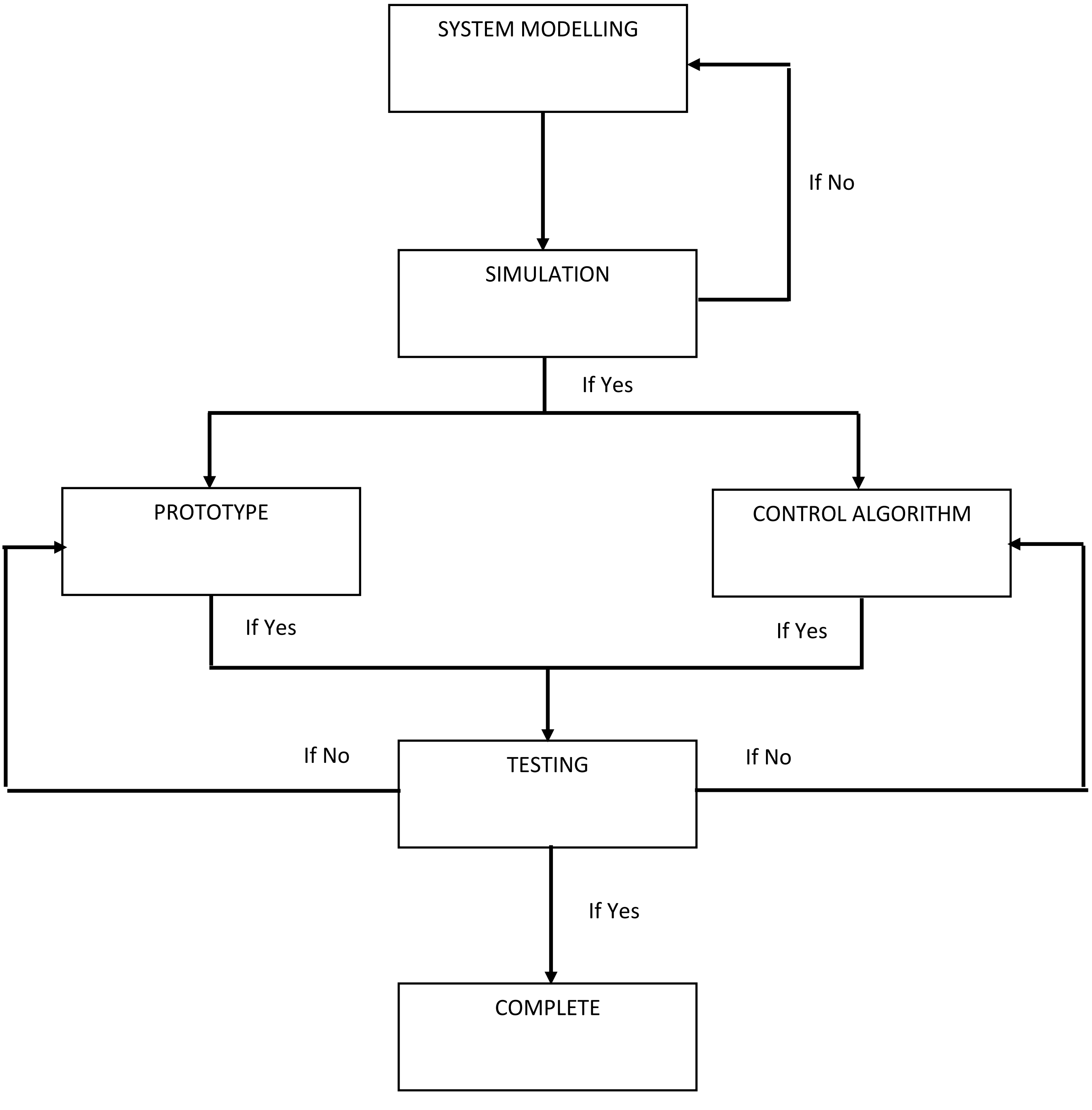

Figure 3.1:Process flow for developing TWSB

3.2 System Modelling

System modelling is performed to obtain the equation utilised for the TWSB robot's cart and pendulum. In order to get optimal control term values for the TWSB robot, the outcome of system modelling will be transferred into MATLAB for simulation. The equation is derived as stated below in accordance with Figure 3.2.

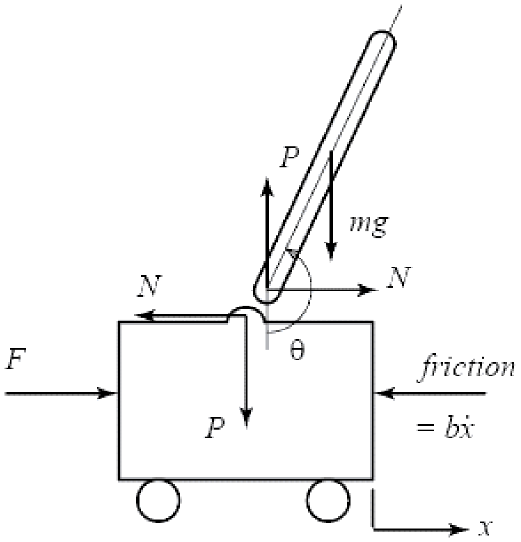

Figure 3.2: Inverted pendulum cart[44]

M	Mass of Cart (kg)
m	Mass of Pendulum (kg)
b	Coefficient of Friction (N/ms$^{-1)}$
l	Length to Pendulum Centre of Mass (m)
I	Mass Moment of Pendulum Inertia (kgm^2)
F	Force (N)
x	Position Coordinate
θ	Pendulum Angle (^{o})
Φ	Deviation of Pendulum Position from Equilibrium

Table 3.1: Parameters for inverted pendulum cart

By adding the forces in the horizontal free-body diagram of the cart, the following equation of motion is obtained.

$$M\ddot{x} + b\dot{x} + N = F \tag{1}$$

The response force N is calculated by adding the forces in the horizontal direction of the free-body diagram of the pendulum.

$$N = m\ddot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta \tag{2}$$

When this equation is substituted into the first equation, one of the two governing equations for this system is obtained.

$$(M+m)\ddot{x} + b\dot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta = F \tag{3}$$

The second equation of motion for this system is obtained by adding the forces perpendicular to the pendulum. Using this axis to solve the problem substantially simplifies the maths. You need to get the following equation:

$$P\sin\theta + N\cos\theta - mg\sin\theta = ml\ddot{\theta} + m\ddot{x}\cos\theta \tag{4}$$

To eliminate the P and N terms from the above equation, add the moments around the pendulum's centroid to get the following equation.

$$-Pl\sin\theta - Nl\cos\theta = I\ddot{\theta} \tag{5}$$

By combining equations (3) and (4), it is possible to remove P and N terms.

$$(I + ml^2)\ddot{\theta} + mgl\sin\theta = -ml\ddot{x}\cos\theta \tag{6}$$

This set of equations must be linearized since the analysis and control design approaches, we shall utilise in this example apply solely to linear systems. Specifically, we will linearize the equations pertaining to the vertically upward equilibrium position, $\theta = \pi$, and assume that the system remains inside a limited region around this equilibrium. This assumption should be reasonable given that, while under control, the pendulum should not stray more than 20 degrees from its vertically upward position. $\emptyset$ must reflect the positional deviation of the pendulum from equilibrium that is $\theta = \pi + \emptyset$. Again, assuming a tiny ($\emptyset$) deviation from equilibrium, we may approximate the nonlinear functions in our system equations using the following small-angle approximations:

$$\cos\theta = \cos(\pi + \phi) \approx -1 \tag{7}$$

$$\sin\theta = \sin(\pi + \phi) \approx -\phi \tag{8}$$

$$\dot{\theta}^2 = \dot{\phi}^2 \approx 0 \tag{9}$$

After substituting the aforementioned approximations into our nonlinear governing equations, we get two linearized motion equations. The letter F has been replaced by the letter u.

$$(I+ml^2)\ddot{\phi} - mgl\phi = ml\ddot{x} \tag{10}$$

$$(M+m)\ddot{x} + b\dot{x} - ml\ddot{\phi} = u \tag{11}$$

To derive the transfer functions of the linearized system equations, we must first apply the Laplace transform to the system equations with starting conditions of zero. Below are the obtained Laplace transforms.

$$(I+ml^2)\Phi(s)s^2 - mgl\Phi(s) = mlX(s)s^2 \tag{12}$$

$$(M+m)X(s)s^2 + bX(s)s - ml\Phi(s)s^2 = U(s) \tag{13}$$

Remember that a transfer function describes the connection between a single input and a single output.

$$X(s) = \left[\frac{I+ml^2}{ml} - \frac{g}{s^2}\right]\Phi(s) \tag{14}$$

Then, substitute the above into the second equation.

$$(M+m)\left[\frac{I+ml^2}{ml}-\frac{g}{s^2}\right]\Phi(s)s^2+b\left[\frac{I+ml^2}{ml}-\frac{g}{s^2}\right]\Phi(s)s-ml\Phi(s)s^2=U(s) \qquad (15)$$

The transfer function may be rearranged as follows:

$$\frac{\Phi(s)}{U(s)}=\frac{\frac{ml}{q}s^2}{s^4+\frac{b(I+ml^2)}{q}s^3-\frac{(M+m)mgl}{q}s^2-\frac{bmgl}{q}s} \qquad (16)$$

The above transfer function demonstrates that the origin contains both a pole and a zero. These may be eliminated, rendering the transfer function as follows.

$$P_{pend}(s)=\frac{\Phi(s)}{U(s)}=\frac{\frac{ml}{q}s}{s^3+\frac{b(I+ml^2)}{q}s^2-\frac{(M+m)mgl}{q}s-\frac{bmgl}{q}} \qquad [\frac{rad}{N}] \qquad (17)$$

In a similar fashion, the transfer function with the cart position X(s) as the output may be constructed to get the following.

$$P_{cart}(s)=\frac{X(s)}{U(s)}=\frac{\frac{(I+ml^2)s^2-gml}{q}}{s^4+\frac{b(I+ml^2)}{q}s^3-\frac{(M+m)mgl}{q}s^2-\frac{bmgl}{q}s} \qquad [\frac{m}{N}] \qquad (18)$$

3.3 PID Algorithm

PID, which is a control loop feedback technique often used in control systems, is used to govern the speed at which wheels should pivot. PID controller is a real-time controller because it can take corrective action to minimise the difference between the intended and actual values by adjusting the output. PID controllers are often used in industries that need continuous modulated control. There is a total of three gains in a PID controller, K_p, K_i, and K_d. All given factors are interdependent, and values are determined through trial and error[33]. Therefore, PID controllers need a manual tuning procedure. Using the PID tuner in MATLAB, however, the physical tuning process is converted into an automated tuning process.

Error value in system is shown by unit step input created by random angle value input. The performance of the TWSB robot may be anticipated using a graphical technique that depicts performance parameters such as rising time, overshoot, steady-state error, and settling time[34].

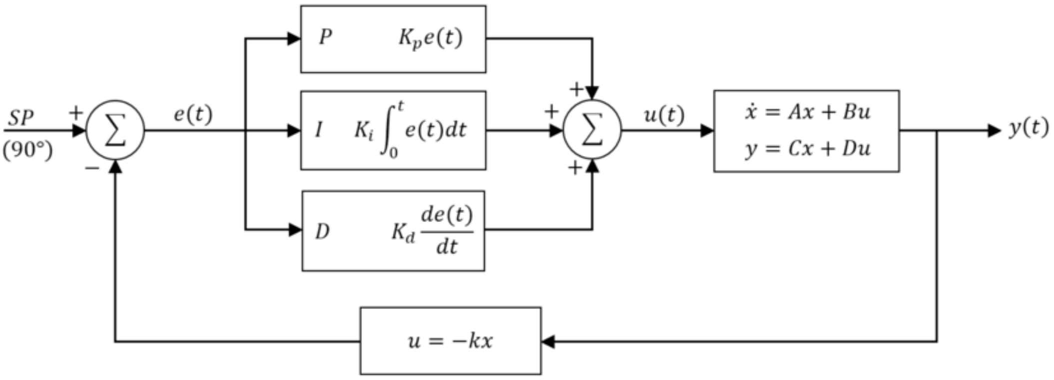

Figure 3.3: PID controller for self-balancing robot[36]