Self-Balancing Bicycle Using Reacting Wheel

A PROJECT REPORT

Submitted by

A. ARI RAM RAJA	**(811519105301)**
M. GOKULA KRISHNAN	**(811519105302)**
P. PREM KUMAR	**(811519105305)**
D. SRIRAM	**(811519105308)**
C. LALITHRAJ	**(811519105501)**

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

ELECTRICAL AND ELECTRONICS ENGINEERING

K. RAMAKRISHNAN COLLEGE OF ENGINEERING

TRICHY

(AUTONOMOUS)

ANNA UNIVERSITY: CHENNAI 600 025

JUNE 2022

INSTITUTE VISION AND MISSION

VISION

"To achieve a prominent position among the top technical institutions"

MISSION

- To bestow standard technical education par excellence through state of the art infrastructure, competent faculty and high ethical standards.
- To nurture research and entrepreneurial skills among students in cutting–edge technologies.
- To provide education for developing high-quality professionals to transform the society.

DEPARTMENT VISION AND MISSION

VISION

To emerge as a renowned department for high quality teaching, learning and research in the domain of Electrical and Electronics Engineering, producing professional engineers, to meet the challenges of society.

MISSION

M1. To establish the infrastructure resources for imparting quality technical education in Electrical and Electronics Engineering.

M2. To achieve excellence in teaching, learning, research and development.

M3. To impart the latest skills and developments through practical approach along with moral and ethical values.

PROGRAM SPECIFIC OUTCOME (PSO)

Students shall:

PSO 1: Students will qualify in National level Competitive Examinations for Employment & Higher

PSO 2: Students will have expertise in the design and development of Hardware and Software tools to solve complex Electronics and Communication Engineering problems in the domains like analog and digital electronics, embedded and communication systems.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)
Our graduates shall:

PEO1: Graduates will become experts in providing solution for the Engineering problems in Industries, Government and other organizations where they are employed.

PEO2: Graduates will provide innovative ideas and management skills to enhance the standards of the society by individual and with team works through the acquired Engineering knowledge.

PEO3: Graduates will be successful professionals through life long learning & contribute to the society technically and professionally.

PROGRAM OUTCOME (PO)

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations**.**

PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: The Engineer and Society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

COURSE OUTCOMES:

S.NO	BLOOMS LEVEL	DESCRIPTION	PO (1.12) &PSO (1.2) MAPPING
C207.1	K3	To expose the students to apply knowledge to solve problems.	PO1 & PSO1, PSO 2
C207.2	K3	To expose the students to find solutions to complex problems, issues for public and environmental concerns.	PO3PO7, PSO1, PSO 2
C207.3	K3	To expose the students to give conclusion, analyses methods for various scenarios.	PO4 & PSO 1, PSO 2
C207.4	K2	To expose the students to communicate efficiently their technical knowledge and concepts.	PO9, PO10 & PSO2
C207.5	K2	To expose the students to self-learning and long-term learning process.	PO12& PSO1

COURSE OUTCOMES VS POS MAPPING (DETAILED; HIGH: 3; MEDIUM: 2; LOW: 1**):**

SNO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
C411.1	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**2**	**3**
C411.2	**-**	**-**	**3**	**-**	**-**	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**2**	**3**
C411.3	**-**	**-**	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**2**	**3**
C411.4	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	**3**	**-**	**-**	**-**	**2**
C411.5	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	**2**	**-**

** For Entire Course, PO /PSO Mapping; 1 (Low); 2(Medium); 3(High) Contribution to PO/PSO*

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report "**SELF BALANCING BICYCLE USING REACTING WHEEL"** is the Bonafede work of **A. ARI RAM RAJA (811519105301), M. GOKULA KRISHNAN (811519105302), P. PREM KUMAR (811519105305), D. SRIRAM (811519105308), C. LALITHRAJ (811519105501)** who carried out the project work under my supervision.

SIGNATURE
Dr. K. Dhayalini Ph.D.,
HEAD OF THE DEPARTMENT
Professor
Electrical and Electronics Engineering
K. Ramakrishnan College
of Engineering(Autonomous)
Samayapuram,
Trichy-621 112.

SIGNATURE
Mr. G. Gabriel Santhosh Kumar M.E.,
SUPERVISOR
Assistant Professor
Electrical and Electronics Engineering
K. Ramakrishnan College
of Engineering(Autonomous)
Samayapuram,
Trichy-621 112.

This project report was submitted for the viva-voce held on …….at K. Ramakrishnan College of Engineering, Trichy-621112.

INTERNALEXAMINER **EXTERNALEXAMINER**

ACKNOWLEDGEMENT

We thank the Almighty God, for showing abundance of grace, without his blessings it would not have been possible for us to complete our project.

At this pleasing moment of having successfully completed our project, we wish to convey our sincere thanks and gratitude to our beloved kind Chairman **Dr. K. RAMAKRISHNAN**, who provided all the facilities to us.

Our sincere gratitude to **Dr. S. KUPPUSAMY**, Executive Director for his constant encouragement. We are also grateful to our Principal **Dr.D.SRINIVASAN**, for his constructive suggestions and encouragement during our project.

We wish to express the profound thanks to **Dr. K. DHAYALINI**, Professor and Head, Department of Electrical & Electronics Engineering, for providing all necessary facilities for doing this project.

We whole heartedly acknowledge our deep sense of gratitude and indebtedness to our beloved guide **Mr. G. GABRIEL SANTHOSH KUMAR**, Assistant Professor, Department of Electrical &Electronics Engineering, for his expert guidance and encouragement throughout the duration of the project.

We extend our gratitude to all the teaching & Non–teaching staff members of Electrical & Electronics Engineering Department, **K. RAMAKRISHNAN COLLEGE OF ENGINEERING** for their kind help and valuable support to complete the project successfully. We would like to thank our parents and friends for their constant support and encouragement throughout this project.

ABSTRACT

This work uses a control moment gyro as an actuator. The control moment gyro is typically used in a spacecraft to orient the vessel. Appling as an actuator to balance a bicycle is a creative and novel approach and is the first of its kind for balancing of a bicycle. In this project a Self-balancing bicycle using reacting wheel were constructed with the intent of it balancing on of one of its edges using a PID-controller to counteract the gravity pulling it down. To be able to this a good understanding of the size team was required to have the right components for the job. A DC motor was used reaction wheel on each side of its axle using the moment of inertia to convert the torque from the motor to an angular acceleration of the bicycle. An IMU was used to determine the angle difference of the center of mass from a fixed point in the room and this value was used as an input in the PID-controller. Different method was used to determine the P, I and D components of the PID-controller to create a stable system. Even though different methods were used no one successfully made the bicycle perfectly.

Keywords: PID-control, reaction wheel,

Pico Raspberry Pi, cube

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATION

DC	DIRECT CURRENT
AC	ALTERNATIVE CURRENT
IC	INTEGRATED CIRCUIT
I2C	INTER-INTEGRATED CIRCUIT
PID	PROPORTIONAL INTEGRAL DERIVATIVE
IDE	INTEGRATED DEVELOPMENT ENVIRONMENT
ROV	REMOTELY OPERATED VEHICLE
PWM	PULSE WIDTH MODULATION
VCC	VOLTAGE COMMON COLLECTOR
SPI	SERIAL PERIPHERAL INTERFACE
GPIO	GENERAL PURPOSE INPUT/OUTPUT
UART	UNIVERSAL ASYNCHRONOUS RECEIVER-TRANSMITTER
KD	DERIVATIVE COMPONENT
KI	INTEGRAL COMPONENT
KP	PROPORTIONAL COMPONENT
T	TORQUE
KG	KILOGRAM

CHAPTER – 1

INTRODUCTION

The idea is to make it balance by rotating a reaction wheel with a motor creating a torque with the moment of inertia making the bicycle counteracts its own weight. The acceleration of the wheels is then adjusted relative to the angle between the bicycle and the horizontal plane to keep the bicycle stable.

We inspired by the fact that the design of the physical product seems very simple, but in reality, it is based on a complex control theory problem. Although the final product will not solve any everyday problems, the technology is important as it provides insight into how control theory can be applied in real projects. The experience of controlling motors and handling signals can benefit later projects since similar problems exists in many areas. Examples of areas where this technology is useful are for designing Segway's, rockets and aircrafts etc.

The focus of this project is to design and construct a fully functional prototype of a product. This regarding a budget of 1000 SEK and a project deadline of three months. Other control systems could work for the same control problem but in this project a PID controlled system will be analyzed.

CHAPTER – 2

BASIC CONCEPTS IN SELF BALANCING BICYCLE

2.1 SYSTEM DYNAMICS

To be able to stabilize the bicycle on its corner a relationship between the different affecting factors is needed and a model of the system dynamics is set up. Mmax is the maximum torque needed to counteract the construction´s own weight at a horizontal position and is given by equation 2.1 where a mechanical equilibrium gives this relationship between the bicycle ’s mass m, gravitation g and the horizontal

length to its center of gravity ½ according to

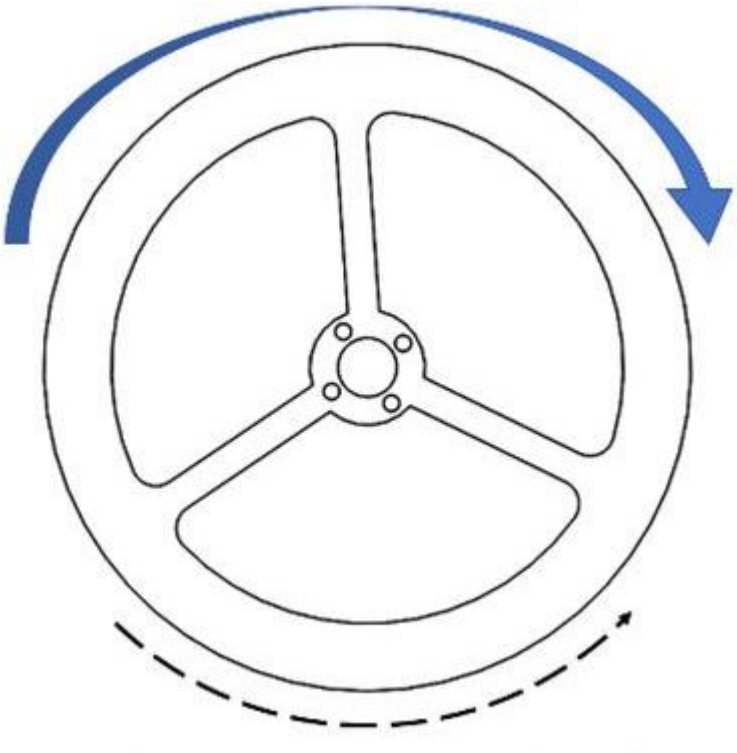

Figure 2.1- Reacting Wheel

$$M_{max} = mg\,½. \quad (2.1)$$

This sets the requirement for needed torque from the motor. An important aspect is to make the construction mechanically stable because the mechanical stability will set a fundamental stability to the construction that gives a good starting point for the automatic control later. This is done by letting the center of mass being centered in the middle of the bicycle.

2.2. REACTION WHEEL

The most suitable shape of a reaction wheel is a ring because it provides the greatest moment of inertia per mass that will result in a decreased acceleration. The theoretical moment of inertia for a ring can be calculated with the ring´s mass m and it´s radius r as

$$I_{ring} = mr^2. \quad (2.2)$$

When the motor accelerates the reaction wheel generates torque. Due to Newtons third law, a reaction torque that acts on the bicycle in the opposite direction is also generated. This can be described by the equation below with Iring as the wheels theoretical moment of inertia and the angle acceleration ω˙ motor of it as [2]

$$M_{motor} = I_{ring}\dot{\omega}_{motor} \quad (2.3)$$

By adjusting the speed and acceleration of the motor the angle between the bicycle and the surface can be controlled. In this project a Newtonian model was used that can be described as [2]

$$I_{tot}\ddot{\theta} = mgl\sin(\theta) - M_{motor} \quad (2.4)$$

2.3. PID-CONTROLLER

A PID-controller is the most commonly used controller in the industry. The con- troller use a reference point that is the value it wants to reach and compare it with its current value and this difference is the error. It then uses the error to stabilize

the system with three components. PID means that the controller consists of three components: a proportional, an integrative and a derivative component. The P-component is used to adjust the gain, the I-component is used to reduce the static error, and the D-component compensates for changes in the system that increases the stability.

Since Pico Pi includes a built in a PID-function within its libraries it will be used in this project. The PID-function has three parameters KP, KI, and KD that will be choose experimentally. The ideal PID-regulator can be written as:

$$u(t) = K_P e(t) + K_I \int_{t_0}^{t} e(\tau)d\tau + K_D \frac{de(t)}{dt}. \tag{2.5}$$

There are different ways a PID-controller can be tuned to stabilize a system. One method is the Ziegler-Nicholas method where analysis of the system could give a good approximation of the three components creating the PID-controller. Given a system the simplest method is to increase the gain for the system until a steady state oscillation occurs, giving the Kcr. This with the given oscillation period Pcr can with this method give a good starting point for tuning the PID. The method suggests the following values in relation with the analyzed valued above:

$$\begin{aligned} KP &= 0.6Kcr, \\ TI &= 0.5Pcr, \\ TD &= 0.125Pcr. \end{aligned} \tag{2.6}$$

Components

2.4. MOTORS

In this project a brushed DC motor was used. A brushed motor means that it rotates internally. DC stands for direct current which implies that ohm's law and Kirchhoff's voltage law can be used in calculations. The relationship between voltage, current and velocity can therefore

$$U_A = R_A I_A + K_2 \Phi \omega. \quad (2.)$$

The torque that the motor generates is proportional to the current and is mathematically described as

K2Φ is a constant that can be calculated by measuring the speed of the rotor shaft when there is no load on the motor. This gives that the voltage is direct proportional to the rotating speed (UA = K2Φω). In this project it was instead found in the data sheet for the motor. Ra is the terminal resistance can be measured, but was also found in the data sheet.

The angular velocity of the flywheel changes over time when the voltage 12V is applied on the motor. The angular velocity was calculated with the Euler forward method and depends on the torque that was received from equation 2.6. In the data sheet the" no load speed" for the motor is 300 rpm. The reason why the motor passes this speed despite there is load on the motor might be because the used model does not consider internal friction of the motor. A theoretical value of the load was calculated by approximating the flywheel as a ring.

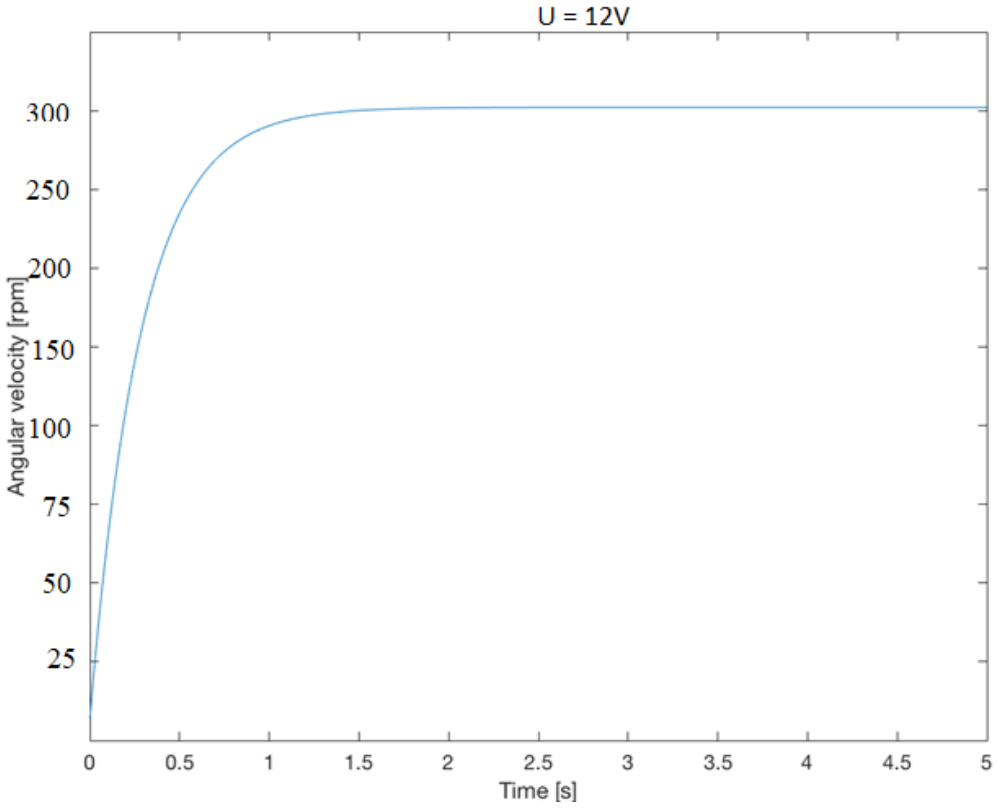

Figure 2.2 - Angular velocity over time when a voltage of 12V is applied on the motor.

2.5. SENSORS

GYRO/ACCELEROMETE

To know how the bicycle is positioned according to the room a gyro/acetometer is used. The gyrometers origin is set to the position the bicycle is supposed to balance in. When the bicycle is falling in some direction the gyrometers will give a signal to the micro controller of this change in angle. The micro controller will then do actions to stabilize the bicycle and bring its position back to origin.

2.6. PICO RASPBERRY PI

Pi Pico is a micro-controller board based on in-house custom designed chip (RP2040) by Raspberry Pi. More experienced users can take advantage of Raspberry Pi Pico's rich peripheral set, including SPI, I2C, and eight Programmable I/O (PIO)

state machines for custom peripheral support.

The chip at the heart of Pico, is an ultra-powerful Dual Core ARM Cortex-M0+ clocked at 133MHz with 256KB RAM, 2MB of on-board QSPI Flash memory for code and data storage. It also has 30 GPIO pins and lots of interfacing options including 2 × SPI, 2 × I2C, 2 × UART, 3 × 12-bit ADC, 16 × controllable PWM channels. Moreover, 26 of the GPIO pins are multi-function, which means we can configure those pins to a different behavior like UART/I2C/SPI/GPIO

2.7. H-BRIDGE

To drive a DC-motor with higher voltage it is necessary to use a H-bridge. This is due to the Pico pi´s inability to output voltage over 5 V. A H-bridge is used to control an external power source with small PWM-signals described in section 2.2.3.

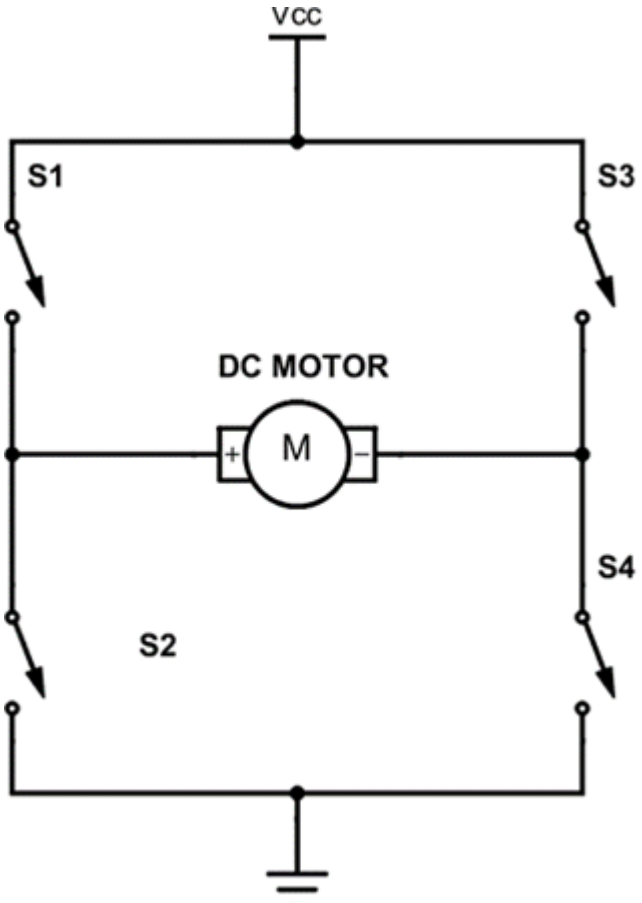

Figure 2.3 - A schematic image of a H-bridge

Another reason why it is important to use an H-bridge is because DC-motors also can work as generators. Therefore, if some external force would make the motor start spinning it would induce a current which could destroy the Pico pi. Conventionally an H-bridge is made of four transistors that not only makes it possible to change the output voltage, it can also change the current direction.

CHAPTER – 3

COMPONENTS IMPLEMENTATIONS IN BICYCLE

IMPLEMENTATIONS OF COMPONENTS

3.1. GYROSCOPE

The gyroscope used in this project is the MPU-6050 accelerometer and gyroscope. It gives values for the change of angle on the z-axle which was used as input for the PID-controller to control the control system.

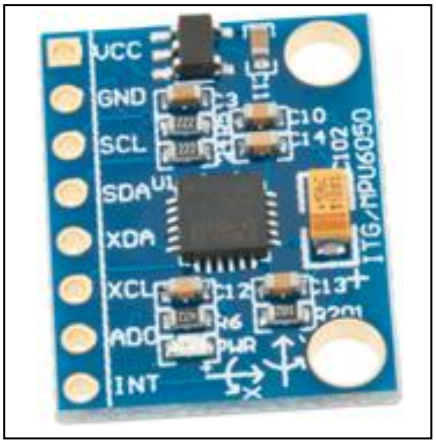

Figure 3.1 - MPU-6050 - Gyroscope/Accelerometer

3.2. SERVO MOTOR

There are some special types of application of electrical motor where rotation of the motor is required for just a certain angle not continuously for long period of time. For these applications, some special types of motor are required with some special arrangement which makes the motor to rotate a certain angle for a given electrical input (signal). This is normally a simple motor which is controlled for specific angular rotation with the help of additional servomechanism (a typical closed loop feedback control system). Servo motor is a special type of motor which is automatically operated up to certain limit for a given command with help of error-sensing feedback to correct the performance

3.3. L298N DRIVER

The L298N is a dual H-Bridge motor driver which allows speed and direction control of two DC motors at the same time. The module can drive DC motors that have voltages between 5 and 35V, with a peak current up to 2A. This depends on the voltage used at the motors VCC. The module has an onboard 5V regulator which is either enabled or disabled using a jumper.

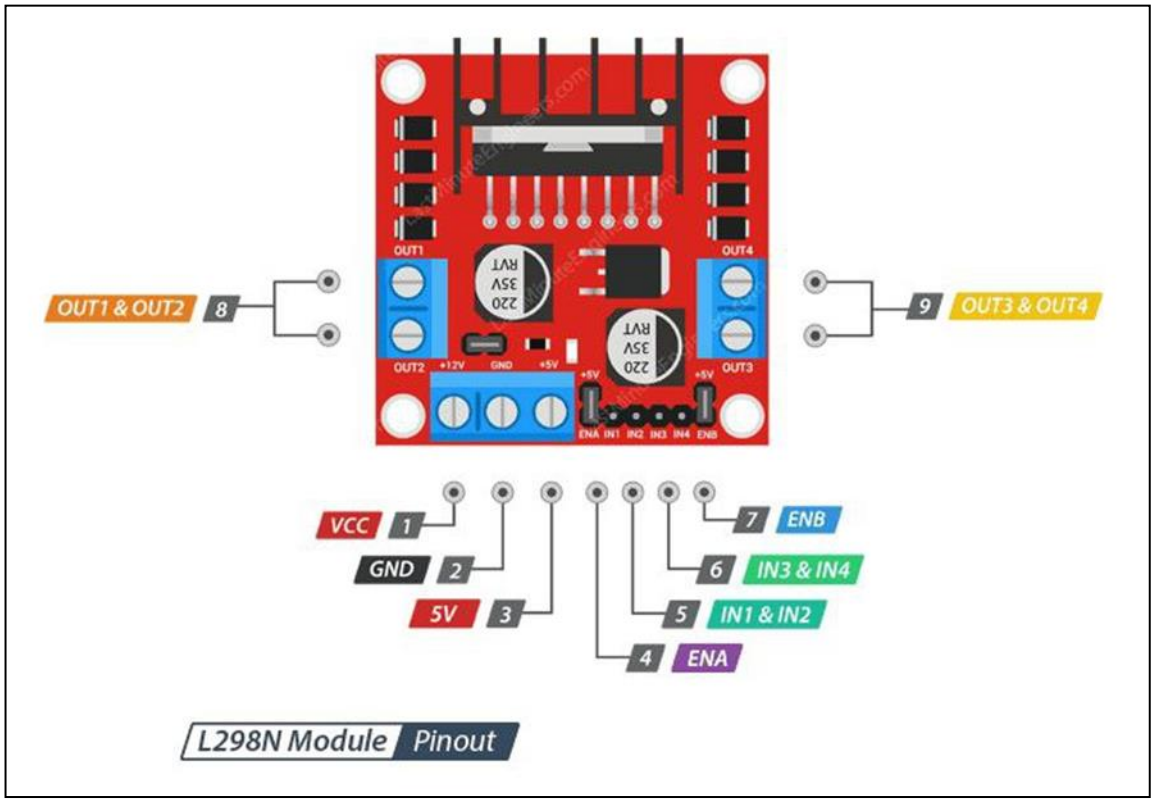

Figure 3.2 - L298N Driver

If the motor supply voltage is up to 12V we can enable the 5V regulator and the 5V pin can be used as output, for example for powering our Pico Pi board. But if the motor voltage is greater than 12V we must disconnect the jumper because those voltages will cause damage to the onboard 5V regulator. In this case the 5V pin will be used as input as we need connect it to a 5V power supply in order the IC to work properly.

3.4. BLUETOOTH MODULE (HC-05 Bluetooth Module)

HC-05 module is an easy-to-use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup. The HC-05 Bluetooth Module can be used in a Master or Slave configuration, making it a great solution for wireless communication. This serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate)3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR Blue core 04- External single chip Bluetooth system with CMOS technology and with AFH (Adaptive Frequency Hopping Feature).

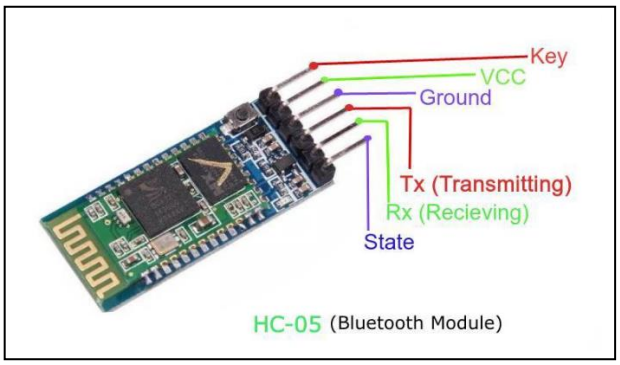

Figure 3.2 - Bluetooth module

3.5. BICYCLE FRAME

The bicycle is designed to have all of its components within its sides with the sides acting as a protective shell and its corner as good balancing points. The shell will consist of six plates that make up the bicycles sides that will be laser cut from acrylic glass with a thickness of three millimeter. Acrylic glass was chosen because of its low density and good enough yield strength. bicycle we are using two other plates acting as motor holders that is connected to two opposite sides with distance screws in each corner of the plates. The motor is mounted on the motor holders. The two reaction wheels is placed on the motor axle on opposite sides between the side

of the bicycle and the motor holders,

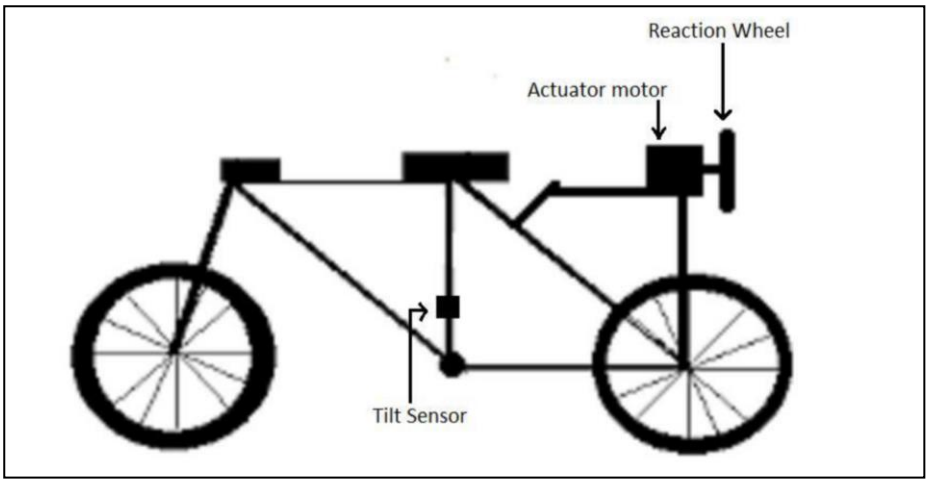

Figure 3.3 - The Bicycle with Reaction Wheels and Motor Installed

3.3 REACTION WHEEL

The wheel's purpose is to generate as much moment of inertia per mass, therefore the shape of a ring. Two wheels with a radius of 7 centimeter and the weight of 95 g were created to generate it.

The reaction wheel is made of acrylic sheet and cut because of its simplicity with the wheel only needed to be cut in one dimension. The wheel is then mounted on the motor axle with a hub that is connected directly on the reaction wheel with six M3 screws and then locked on the axle with two set screws from the hub. A requirement for the wheel is to be able to store enough energy that it takes for the bicycle to tilt to an angle of 45 ◦. The kinetic energy that the wheel can store is

where ωmax is the maximum speed [rad/s] and Iring is the moment of inertia that is described in equation 2.2. The energy it needs to store is the difference in potential energy between the states when it is lying horizontal on the surface and balancing at a 45 ◦ angle. This is mathematically described

$$V = \sqrt{2m\ bicycle\ G½} \quad (3.1)$$

where bicycle is the mass of the bicycle [kg], g is gravitational acceleration [m/s2] and l is the length of the bicycle. By using setting V equal to T the wheels minimum mass can be calculated to

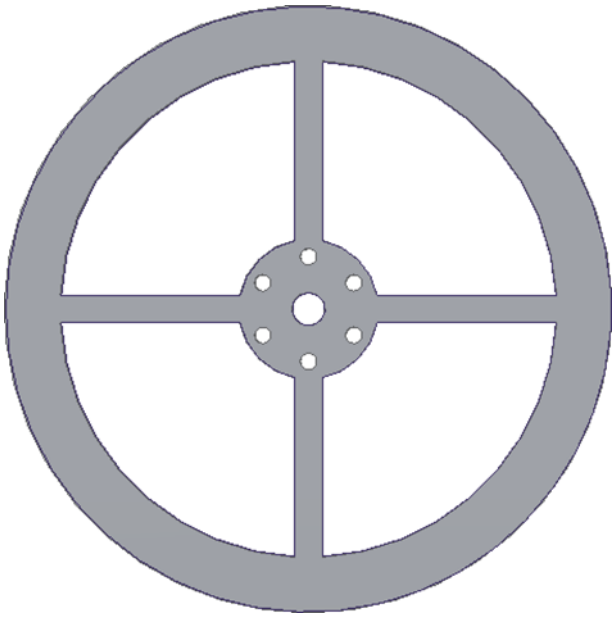

Figure 3.4 - Reaction wheel from the side. Made in Solid Edge

PID-CONTROLLER

To construct a stable system with a PID-controller, the three components KP, KI and KD need to be tuned right. Proportional-integral-derivative control (PID) combines the stabilizing influence of the derivative term and the reduction in steady-state error from the integral term.

$$u(t) = K_\mathrm{p} e(t) + K_\mathrm{i} \int_0^t e(\tau)\,\mathrm{d}\tau + K_\mathrm{d} \frac{\mathrm{d}e(t)}{\mathrm{d}t},$$

(3.2)

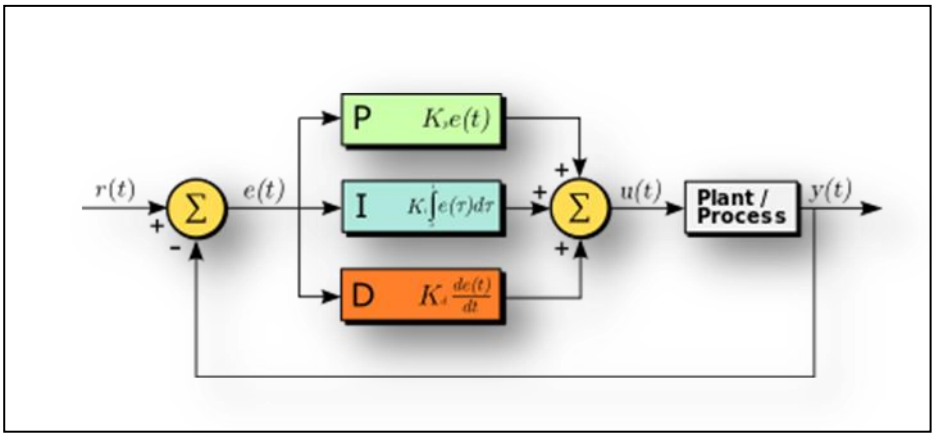

Figure 3.6 - Block diagram PID controller

3.7. PROPORTIONAL

The proportional term produces an output value that is proportional to the current error value. The proportional response can be adjusted by multiplying the error by a constant Kp, called the proportional gain constant.

$$P_\mathrm{out} = K_\mathrm{p} e(t).$$

(3.3)

3.8. INTEGRAL

The contribution from the integral term is proportional to both the magnitude of the error and the duration of the error. The integral in a PID controller is the sum of the instantaneous error over time and gives the accumulated offset that should have been corrected previously. The accumulated error is then multiplied by the integral gain (Ki) and added to the controller output.

$$I_{\text{out}} = K_{\text{i}} \int_0^t e(\tau)\, d\tau. \tag{3.4}$$

3.9. DERIVATIVE

The derivative of the process error is calculated by determining the derivative gain, Kd. The magnitude of the contribution of the derivative term to the overall control action is termed the derivative gain, Kd.

$$D_{\text{out}} = K_{\text{d}} \frac{de(t)}{dt}. \tag{3.5}$$

PWM CONTROL

3.10. PWM TO CONTROL MOTOR SPEED

Pulse width modulation (PWM) is a modulation technique that generates variable-width pulses to represent the amplitude of an analog input signal. The output switching transistor is on more of the time for a high-amplitude signal and off more of the time for a low-amplitude signal. The digital nature (fully on or off) of the PWM circuit is less costly to fabricate than an analog circuit that does not drift over time.

PWM is widely used in ROV applications to control the speed of a DC motor and/or the brightness of a lightbulb. For example, if the line were closed for 1 μs, opened for 1 μs, and continuously repeated, the target would receive an average of 50% of the voltage and run at half speed or the bulb at half brightness. If the line were closed for 1 μs and open for 3 μs, the target would receive an average of 25%.

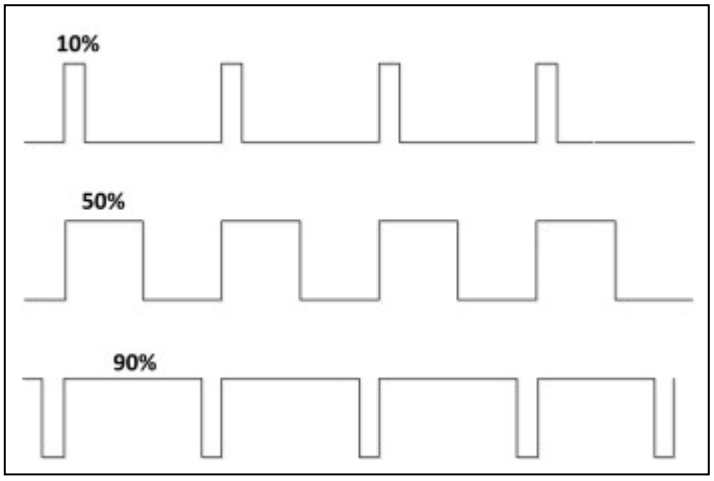

Figure 3.7 - PWM signals with different duty cycles.

CHAPTER – 4

DESIGN OF SELF BALANCING BICYCLE

4.1. BLOCK DIAGRAM OF SELF BALANCING BICYCLE

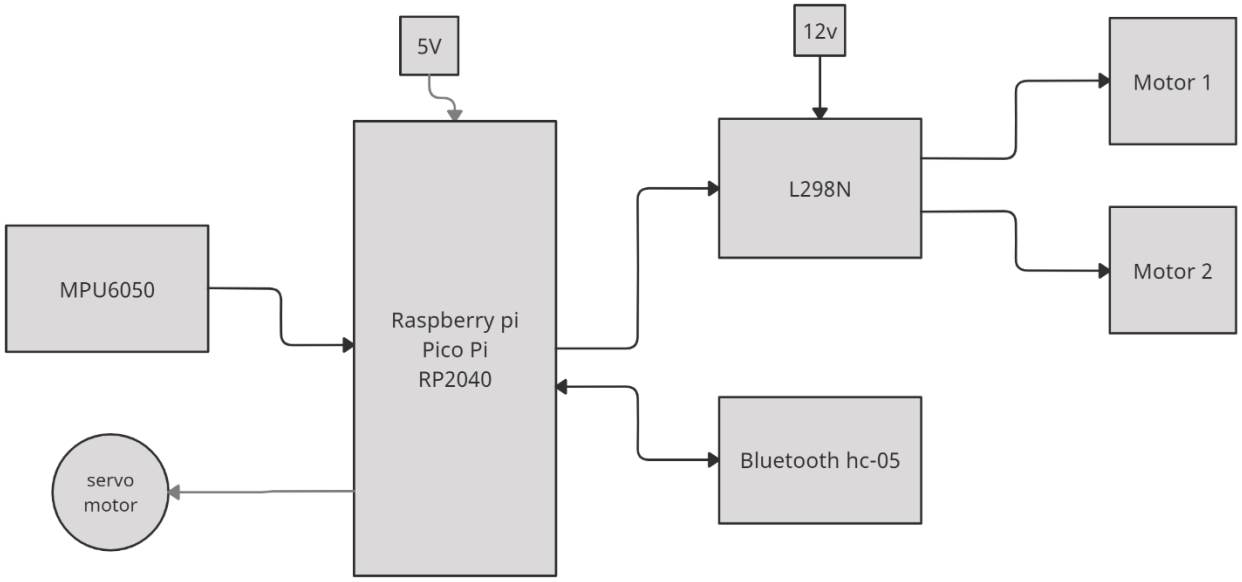

Figure 4.1 – Block diagram of self balancing bicycle using reacting wheel

A block diagram is as show a connecting line in all parts. It is carried data in bipolar and unipolar example I2c, UART, SPI, etc.

4.2. DESCRIPTION OF THE BLOCK DIAGRAM:

RASPBERRY PI PICO: The Pico Pi is used to perform all the operations in the system. The microcontroller is programmed to control the motor for control a self-balancing control to avoid falling ground and I²C received signal to microcontroller Pico pi its work in mathematical operation in PID to find an Error value.

Feature	Availability
Microcontroller	RP2040
Architecture	Dual-core Arm Cortex M0+ processor
Flash	2MB
SRAM	264KB
GPIO Pins	26
I2C	2
UART	2
SPI	2
PWM	8 PWM Blocks 16 Outputs
ADC	3 12-bit ADC
State Machines	8 × Programmable I/O (PIO) state machines
DAC	0
USB	USB 1.1 with device and host support
Timer	single 64-bit counter
Watchdog	one 32-bit
Real time clock	1- RTC

Table 4.1- Raspberry Pi Pico Features and Specifications

MPU6050: The MPU-6050 devices combine a 3-axis gyroscope and a 3-axis accelerometer on the same silicon die, together with an onboard Digital Motion Processor, which processes complex 6-axis Motion Fusion algorithms. The device can access external magnetometers or other sensors through an auxiliary master I²C bus, allowing the devices to gather a full set of sensor data without intervention from the system processor.

L298N: The L298N motor driver module has two screw terminal blocks for the connecting two motors A and B. There is also a power supply screw terminal block containing the Ground pin, the VCC for motor and a 5V pin which can either be an input or output. The diagram below shows the pin out of this motor driver. Input pins IN1, IN2, IN3 and IN4 are for controlling the direction of rotation of the motors where IN1 and IN2 control the direction of rotation of motor A while IN3 and IN4 control direction of rotation of motor B.

L298N IC pins	Name	Function
1,15	Sense A, Sense B	connected to control the current of the load.
2,3	Out 1, Out 2	Outputs of the Bridge A
4	VS	Supply Voltage
5,7	Input 1, Input 2	TTL Compatible Inputs of the Bridge A
6,11	Enable A, Enable B	TTL Compatible Enable Input
8	GND	Ground
9	VSS	Supply Voltage
10,12	Input 3, Input 4	Inputs of the Bridge B.
13,14	Out 3, Out 4	Outputs of the Bridge B.

Table 4.2 – L298N IC pin Features and Specifications

HC-05 Bluetooth Module: Make the connections and power on the Bluetooth Module. If this is the first time you are using your Bluetooth Module, then the LED will blink rapidly. In order to pair the module with your phone, open Bluetooth Settings in your phone and connect to “HC-05”,

Key Name	**Data of Key**
LEFT DIRECTION	1
CENTER	2
RIGHT DIRECTION	3
FORWARD DIRECTION	4
REVERSE DIRECTION	5

Table 4.1- Bluetooth Data configuration

4.3. FRITZING DIAGRAM OF SELF BALANCING BICYCLE USING REACTING WHEEL

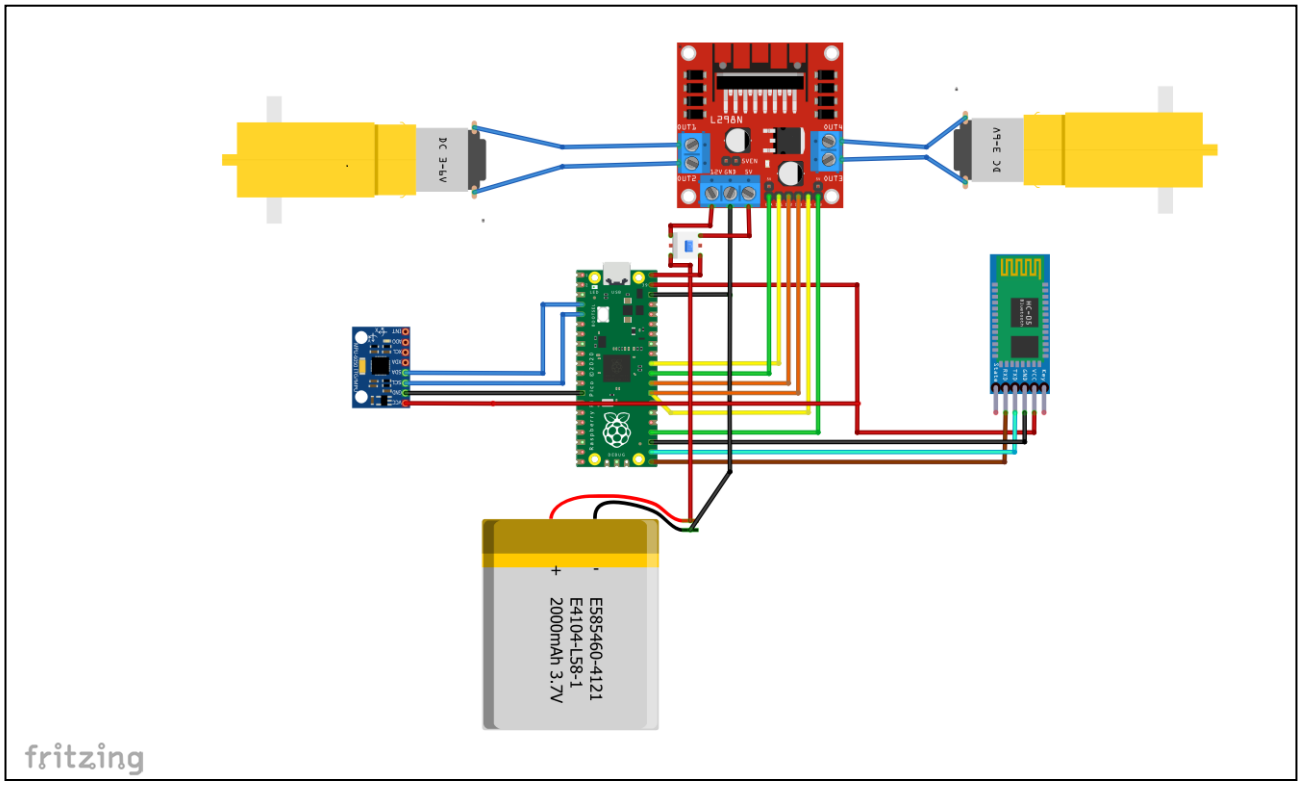

Figure 4.2 - Fritzing Diagram of Self Balancing Bicycle Using Reacting Wheel

CHAPTER – 5

DESCRIPTION OF HARDWARE AND SOFTWARE REQUIRED TO SELF BALANCING BICYCLE

HARDWARE REQUIRED:

- **BO MOTOT**
- **H BRIDGE**
- **BATTERY**
- **BLUETOOTH**
- **REACTION WHEELS**
- **MICROCONTROLLER**

5.1. BO MOTOR

Figure 5.4 - Bo Motor

Bo motor speed 300rpm. Then L type BO Motor shaft using in this self-balancing bicycle. This motor speed control using a PWM.

5.2. H BRIDGE

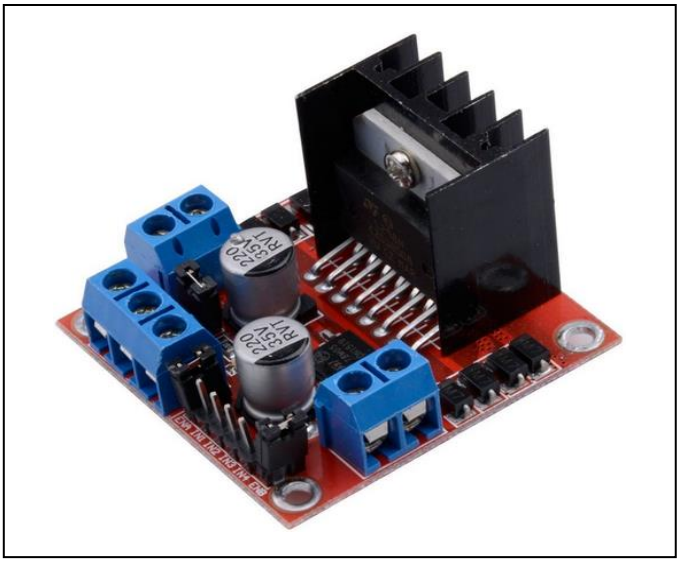

Figure 5.2 – L298N driver module

L298N is a motor control device and speed control using pulse with module. It is control using a MOSFET duty cycle

5.3. BATTERY

Figure 5.3 – Battery in series

Portable equipment needing higher voltages use battery packs with two or more cells connected in series. shows a battery pack with four 3.6V Li-ion cells in series, also known as 4S, to produce 14.4V nominal. In comparison, a six-cell lead acid string with 2V/cell will generate 12V, and four alkaline with 1.5V/cell will give 6V.

5.4. BLUETOOTH

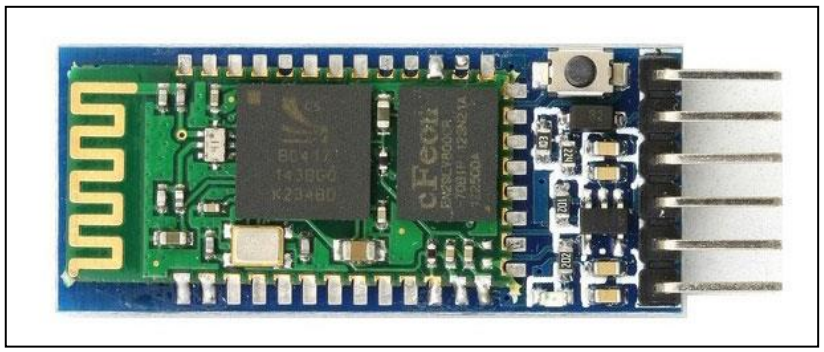

Figure 5.4 – Bluetooth module HC-05

The HC-05 has two operating modes, one is the Data mode in which it can send and receive data from other Bluetooth devices and the other is the AT Command mode where the default device settings can be changed.

5.5. REACTION WHEELS

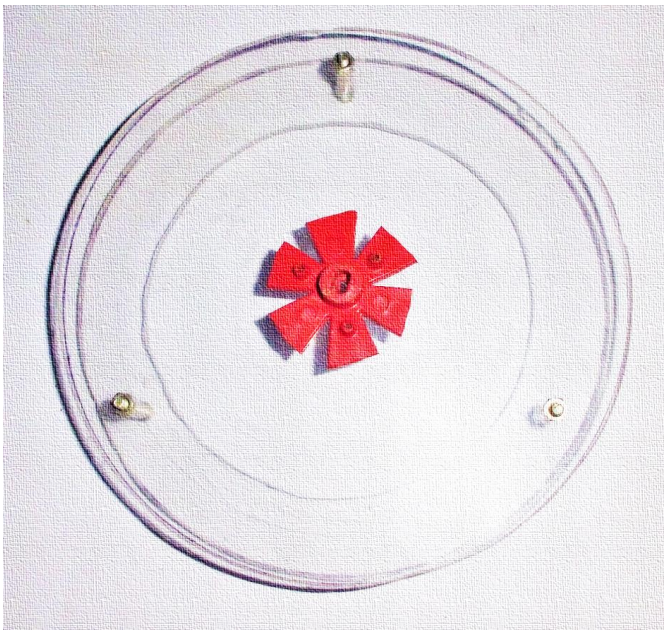

Figure 5.5 – Reacting wheel

A reaction wheel is a device consisting of a spinning wheel attached to an electric motor (brushless DC), whose speed can be controlled by onboard computer, producing the required torque for attitude control. It works on the principle of angular momentum conservation of flywheel. Each wheel can produce torque only along its axis of rotation.

5.6. MICROCONTROLLER

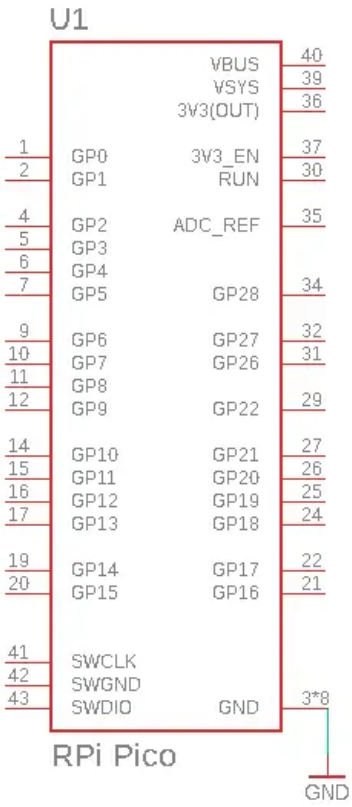

Figure 5.6 – Pico Pi Pin Configuration

The Raspberry Pi Pico is vastly different from any model before it. It is the first device to use RP2040 “Pi Silicon” which is a custom System on Chip (SoC) developed by the Raspberry Pi team which features a dual core Arm Cortex M0+ running at 133 MHz, 264KB of SRAM and 2MB of flash memory used to store files.

5.7. SOFTWARE REQUIRED

- Θ Thonny – To upload a programming IDE
- Θ Server Bluetooth Terminal – To control a Bluetooth hc-05 Module
- Θ Fritzing – Fritzing is an open-source hardware initiative that makes electronics accessible as a creative. the design of electronics hardware, intended to allow designers and artists to build more permanent circuits from prototypes.

ANALYSIS

The PID components were determined by experiments on the cube. The KP component was increased until the value made the cube counteract its own weight when falling to one side. Though the controller was able to counteract the cube falling it couldn't counteract the response from the controller and the cube fell to the other side, the system was unstable. To compensate for this response a KD component was added that will counteract the response based on the speed of the error change. Due to the controller having the error as input it will only respond when an error occurs making the system never settling to the set point. Adding a KI component calculating the integral of the error over time will make the system even more stable. Though using this method to find accurate components for the PID controller the system in this project could not stabilize completely, but made the system many times more stable than using no controller.

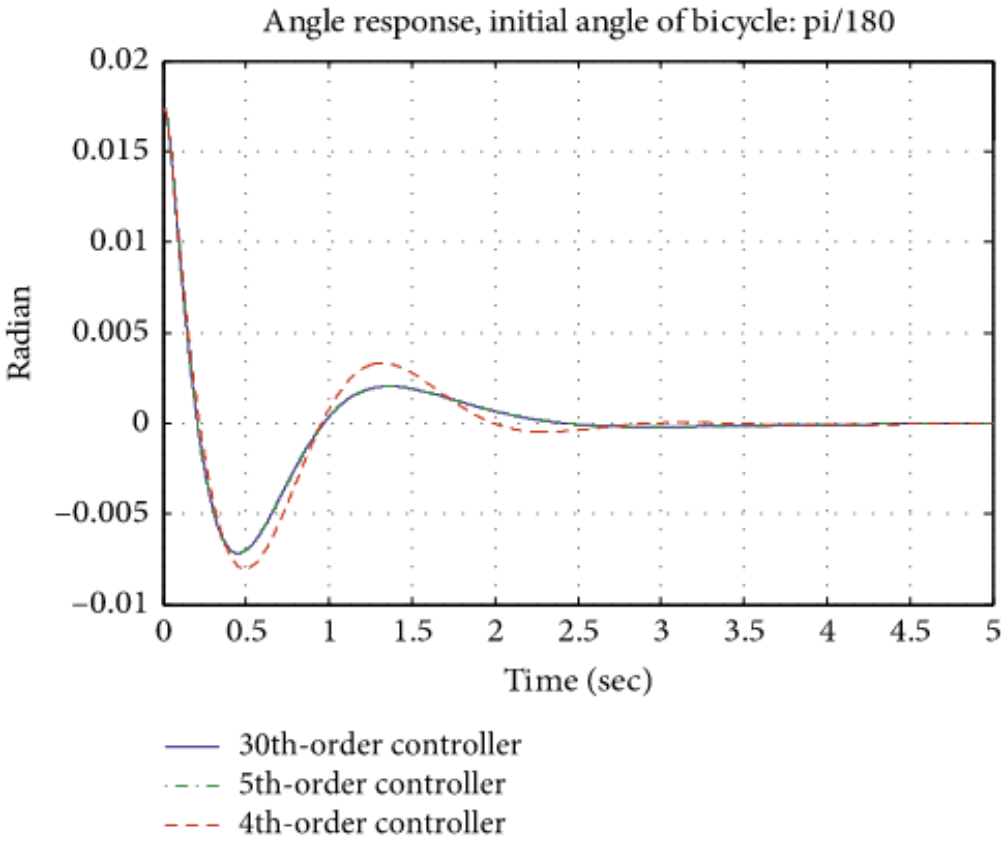

Figure 5.7 – Analysis of Stabilization

CHAPTER – 6

CONCLUSION AND FUTURE SCOPE

The simplest PID controller is applied to balance the bike about the axis and to attain an excellent stability. There are ongoing researches for development of highly featured control systems & control algorithms for the vertical balancing and to minimize effects of external disturbances, such as wind, sudden crash impact etc. and giving driving capabilities to the bike with straight line trajectory and a curved trajectory with maintained tilt angle such as to balance the centrifugal forces generated at the time of turning the bikes are sustainable and practical personal mobility solutions for all major kind of environments; However, many campus environments also experience traffic congestion, parking difficulties and so a self-balancing bike would be an optimal solution in such cases as it may encourage people to switch to self-balancing two wheelers and further there would be many research & design optimizations so as to develop a pollution free and environment friendly e-bike.

REFERENCES

[1]. Beznos A V et al (1998),” Control of autonomous motion of two-wheel bicycle with gyroscopic stabilization”, Robot Autom 3, pp.2670-2675.

[2]. Gallaspy, J.M. (1999), “Gyroscopic stabilization of an Unmanned Bicycle”, M.S. Thesis, Electrical Engineering Department, Auburn University, AL.

[3]. Jongkil Lee, W. K. Van Moorhem (1996), “Analytical and Experimental Analysis of a Self-Compensating Dynamic Balancer in a Rotating Machanism”, Journal of Dynamic Systems Measurement and Control 118(3), September.

[4]. Y. and Tanaka, T. Murakami (April 2004), “Selfsustaining bicycle robot with streeing controller”, IEEE Conference on Advanced Motion Control.

[5]. Murata boy and murata girl (2005), http:// www. murata. com/ about/mboymgirl/mboy.

[6]. Pom, Y., L. (17-19 September 2011), “Gyroscopic stabilization of a kid-size bicycle”, The 5th IEEE International Conference on Cybernetics and Intelligent Systems (CIS), pp. 247-252.

[7]. Suprato (May 2006), “Development of a gyroscopic unmanned bicycle”, Master Thesis, Asian Institution of Technology.

[8]. Thanh, B. T. (September 2008), “Balancing control of bicycle robot by particle swarm optimization-based structurespecified H2/H∞ Control”, Dissertation, Asian Institution of Technology.

[9]. Pornnutvuttikul, W. (December 2009), “Performance and robustness analysis of balancing control of a bicycle robot by Fuzzy Sliding Mode control”, Master Thesis, Asian Institution of Technology.

[10]. Python code. http://blog.bitify.co.uk/2013/11/reading-data-from-mpu-6050-on-raspberry.html

APPENDICES

Micropython Code

**********main.py**********

```
import math, utime, time
from time import sleep
from machine import Pin, I2C, PWM, UART
from imu import MPU6050
from PID import PID
import motor as MOTOR

#measuring angle x,y &z
gyro_scale = 131.0
accel_scale = 16384.0
RAD_TO_DEG = 57.29578
M_PI = 3.14159265358979323846

#onboad led
led_onboard = machine.Pin(25, machine.Pin.OUT)
led_onboard.value(1)
#Gyeroscop sensor
i2c = I2C(1, sda=Pin(2), scl=Pin(3), freq=500000)
imu = MPU6050(i2c)

#Defining UART channel and Baud Rate
uart= UART(0,9600)

data =2

K = 0.98
K1 = 1 - K

time_diff = 0.1
```

```
#sensor = MPU6050(bus, address, "MPU6050")
#sensor.read_raw_data()        # Reads current data from the sensor

rate_gyroX = 0.0
rate_gyroY = 0.0
rate_gyroZ = 0.0

gyroAngleX = 0.0
gyroAngleY = 0.0
gyroAngleZ = 0.0

raw_accX = 0.0
raw_accY = 0.0
raw_accZ = 0.0

rate_accX = 0.0
rate_accY = 0.0
rate_accZ = 0.0

pitch = 0.0
roll = 0.0

accAngX = 0.0

CFangleX = 0.0
CFangleX1 = 0.0

K = 0.98

FIX = -12.89

'''def pitch_ (pitch):
    pitch = 180 * math.atan2(imu.accel.x, math.sqrt(imu.accel.y*imu.accel.y +
imu.accel.z*imu.accel.z))/M_PI
```

```
    return pitch

while True:
    print "{0:.4f} {1:.2f} {2:.2f} {3:.2f} {4:.2f} {5:.2f} {6:.2f}".format( time.time() - now, (last_x),
gyro_total_x, (last_x), (last_y), gyro_total_y, (last_y))
    time.sleep(0.1)'"
def dist(a, b):
    return math.sqrt((a * a) + (b * b))

def get_y_rotation(x,y,z):
    radians = math.atan2(x, dist(y,z))
    return -math.degrees(radians)

def get_x_rotation(x,y,z):
    radians = math.atan2(y, dist(x,z))
    return math.degrees(radians)
p=PID(1.0,-0.04,0.0)
p.setPoint(0.0)

for i in range(0, int(300.0 / time_diff)):
    time.sleep(time_diff - 0.005)

    #chang
    # Gyroscope value Degree Per Second / Scalled Data

    rate_gyroX = imu.gyro.x
    rate_gyroY = imu.gyro.y
    rate_gyroZ = imu.gyro.z
    # The angle of the Gyroscope
    gyroAngleX += rate_gyroX * time_diff
    gyroAngleY += rate_gyroY * time_diff
    gyroAngleZ += rate_gyroZ * time_diff

    # Accelerometer Raw Value
```

```
raw_accX = imu.accel.x
raw_accY = imu.accel.y
raw_accZ = imu.accel.z

# Accelerometer value Degree Per Second / Scalled Data
rate_accX = imu.accel.x
rate_accY = imu.accel.y
rate_accZ = imu.accel.z

# http://blog.bitify.co.uk/2013/11/reading-data-from-mpu-6050-on-raspberry.html
accAngX1 = get_x_rotation(rate_accX, rate_accY, rate_accX)
CFangleX1 = ( K * ( CFangleX1 + rate_gyroX * time_diff) + (1 - K) * accAngX1 )

# Followed the Second example because it gives resonable pid reading
pid =int(p.update(CFangleX1))
speed_ = pid * 30
print(data,pid,"       ",end="\r")

if uart.any(): #Checking if data available in bluetooth
    data = uart.read() #Getting data
    data = int(data)

if(pid > 0):
    MOTOR.forward(speed_)
if(pid < 0):
    MOTOR.backward(abs(speed_))
if(data == 1):
    MOTOR.servo_left()
    sleep(1.5)
    data=2
if(data == 2):
    MOTOR.servo_forward()
if(data == 3):
    MOTOR.servo_right()
```

```
        sleep(1.5)
        data=2
    if(data == 4):
        MOTOR.up()
    if(data == 5):
        MOTOR.down()
    if(data == 6):
        MOTOR.stop_()
```

**********motor.py**********

```
from machine import Pin, PWM
from time import sleep

# pin declarations
    # pins for motor 1
en1 = Pin(8, value = 0, mode=Pin.OUT)
cw = Pin(7, value=0, mode=Pin.OUT)
acw =Pin(9, value=0, mode=Pin.OUT)

    # pins for motor 2
en2 = Pin(20, value=0, mode=Pin.OUT)
in2a = Pin(19, value=0, mode=Pin.OUT)
in2b = Pin(21, value=0, mode=Pin.OUT)
#servo value set
MID= 1300000
MIN = 900000
MAX= 1900000

#servo pin for PWM
pwm = PWM (Pin (15))
```

```
Speed1=PWM (en1)
speed=PWM (en2)

def backward(speed_):
    Speed1.freq(50)
    Speed1.duty_u16(int(speed_/100*65536))
    cw.value(0)
    acw.value(1)
def forward(speed_):
    Speed1.freq(50)
    Speed1.duty_u16(int(speed_/100*65536))
    cw.value(1)
    acw.value(0)
def stop():
    cw.value(0)
    acw.value(0)
def up():
    speed.duty_u16(10000)
    speed.freq(50)
    in2a(1)
    in2b(0)
def down():
    speed.duty_u16(10000)
    speed.freq(50)
    in2a(0)
    in2b(1)
def stop_():
    in2a(0)
    in2b(0)
```

```
def servo_left():
    pwm.freq(50)
    pwm.duty_ns (50)
    pwm.duty_ns (MIN)

# Backward
def servo_forward():
    pwm.freq(50)
    pwm.duty_ns (50)
    pwm.duty_ns (MID)
#Turn Right
def servo_right():
    pwm.freq(50)
    pwm.duty_ns (50)
    pwm.duty_ns (MAX)
```