



الجمهورية الجزائرية الديمقراطية الشعبية

Popular Democratic Republic of Algeria

وزارة التعليم العالي و البحث العلمي

Ministry of Higher Education and Scientific Research

جامعة سعد دحلب البليدة

University of SAAD DAHLEB - Blida 1

كلية التكنولوجيا

Faculty of Technology

قسم الإلكترونيك

Electronics Department

Thesis Of Master

Mention Electronics

Speciality Electronics and Embedded System

presented by

Khaled AHMED HADJALLAH

&

Mohamed HADJ MOHAMED

Control And Realization Of Two Wheeled Self-Balancing Robot

Proposed by: D^r.D.NACEUR & Mr.W.RELLAM

Scholar Year 2017-2018

Acknowledgment

Firstly, I give thanks to God for protection and ability to do work.

This Master Thesis has been done with technical and moral support from several persons. I would like to thank all of them.

Secondly, I would like express my gratitude to the Faculty of Technology, Electronics and Embedded System for providing me the opportunity to study at the university

The successful completion of this project would not have been possible without the guidance by Dr D.NACEUR, Mr. W.RELLAM.

My brothers for the support, encouragement and company spent over the years.

*My parents for their love and moral support when I most needed it.
My relatives for the moral support.*

Finally, a word of appreciation goes out to all my friends and colleagues at the university and at our society ALGERIE TELECOM for showing interest in my work.

Thanks to all of you!

AHMED HADJ ALLAH KHALED

Firstly, I would like express my gratitude to the Faculty of Engineering, Electronics Department for giving us this opportunity to study master II as engineers.

The successful completion of this project would not have been possible without the guidance by D^r Nacer Djamila and Mr Rellam Wahid (always bursting with ideas and suggestions for us to try out for our project), whom I very much consider as our mentor since the day I got to know him.

Thanks to my parents and sisters for their unconditional love and support, I would not have made it this far without their sacrifices. Not forgetting my wife who have always been supportive of my work and lending a hand at times of need.

Finally, a word or appreciation goes out to all my friends and colleagues at the University for showing interest in my work.

ملخص:

يتألف هذا المشروع من بناء وتنفيذ روبوت ذي عجلتين قادر على تحقيق التوازن وسيتم تجميع المكونات الهيكلية والميكانيكية والإلكترونية للروبوت بطريقة تؤدي إلى إنتاج منصة غير مستقرة جوهرياً ومن المرجح جداً أن تميل في محور واحد.

إن عجلات الروبوت قادرة على الدوران المستقل في اتجاهين ، يتحكم في كل منهما بمحرك ذو تيار ثابت سيتم الحصول على معلومات حول زاوية الجهاز بالنسبة إلى الأرض (أي الميل) من المستشعرات الموجودة على الجهاز ، والتي ستعالج التغذية المرتدة باستخدام وحدة تحكم (متناسب ، متكامل ، مشتق) لتوليد إشارات التحكم في موضع التعويض إلى المحركات ذات التيار الثابت من أجل موازنة الجهاز

كلمات المفاتيح: روبوت ذي عجلتين ; منصة غير مستقرة ; متناسب ، متكامل ، مشتق ; المستشعرات.

Résumé :

Ce projet consistera à construire et à mettre en œuvre un robot à deux roues capable de s'équilibrer. Les composants structurels, mécaniques et électroniques du robot seront assemblés de manière à produire une plate-forme intrinsèquement instable et très susceptible de basculer dans un axe.

Les roues du robot sont capables de rotation indépendante dans deux directions, chacune entraînée par un moteur à courant continu. Les informations sur l'angle du robot par rapport au sol (c'est-à-dire l'inclinaison) seront obtenues à partir de capteurs sur le robot, qui traiteront la contre-réaction à l'aide d'un contrôleur PID afin de générer des signaux de commande de position de compensation pour les moteurs à courant continu afin d'équilibrer le Robot.

Mots clés : robot à deux roues ; plate-forme instable ; PID ; capteurs.

Abstract :

This project will undertake the construction and implementation of a two-wheeled robot that is capable of balancing itself. The structural, mechanical, and electronic components of the robot will be assembled in a manner that produces an inherently unstable platform that is highly susceptible to tipping in one axis.

The wheels of the robot are capable of independent rotation in two directions,

each driven by a DC-motor. Information about the angle of the device relative to the ground (i.e. tilt) will be obtained from sensors on the device, which will process the feedback using a PID Controller to generate compensating position control signals to the DC-motors in order to balance the device.

Keywords : two-wheeled robot; unstable platform; PID; sensors.

CONTENT

Acknowledgments

Abstract

Content

List of Tables

List of Figures

List of variables and Abbreviations

General Introduction

Chapter 1 – Background Information and Literature Review	4
I.1 Introduction	5
I.2 Inverted Pendulum Systems.....	5
I.3 Model definition.....	6
I.4 Physical Model of the Robot.....	7
I.5 Non-Minimum Phase Zeros and Transfer Function Analysis.....	8
I.6 Controllers.....	9
I.6.1 Types of controllers.....	11
I.6.2 PID Controller	14
a Role of a Proportional Controller (PC)	15
b Role of an Integral Controller (IC).....	16
c Role of a Derivative Controller (DC)	16
d PID controller (PIDC)	17
I.6.3 The transfer function of the PID controller	20
I.6.4 PID pole zero cancellation.....	20
I.6.5 The Ziegler–Nichols tuning method.....	22
I.7 Sensors.....	25
I.7.1 Overview of Gyroscopes	25
a Definition	25
b Historical Overview of Vibratory Gyroscope Technologies	26

c	Vibratory Gyroscope Dynamics	27
d	Rate Gyroscope Operation	29
e	Gyroscope Fundamentals	31
I.7.2	Accelerometer.....	32
a	Theory & Design	32
I.7.3	Interfacing the Arduino MPU 6050	35
a	Introduction	35
b	Structure of I2C	38
c	Addressing in I2C	40
d	Kalman Filter.....	41
I.8	DC MOTORS	41
I.8.1	Electromechanical Energy Conversion	42
a	Electric Motor	43
b	Electric Generator	43
I.8.2	Construction	43
a	Stator	43
b	Rotor	44
I.8.3	Winding	44
I.8.4	DC Motor Basic Principles	44
a	Energy Conversion	44
b	Value of Mechanical Force	45
Chapter 2	METHODOLOGY	48
II.1.	Introduction	49
II.2.	Hardware development	51
II.2.1	Arduino board	51
II.2.2	IMU sensor	52
II.2.3	I2C interface	53
II.2.4	DC Gear-Motor	55

II.2.5	L298 Dual H-Bridge DC Motor Controller	57
II.3	Software development	61
II.3.1	The characteristics of PID controller (Tuning Parameters)	61
CHAPTER 3 - DISCUSSION AND RESULTS		64
III.1	Discussion Results	65
III.1.1	Initialising and enabling the DMP	65
a	1st step: reading data from MPU6050	65
b	2nd step: Finding the minimum speed:	65
III.2.	Control Loop Performance	65
III.3.	Balancing Performance	66
III.3.1.	3rd step: PID Controller	69
General Conclusion		69
II.1.	Project review	70
II.2.	Recommendations for future work	70
References.....		67
Appendix		70
Building the Robots.....		72
Appendix A – EXAMPLE 1 (Control DC-Motor PWM Code .ino).....		73
Appendix B - LQR design.....		70
Appendix C – Nbar control		71
Appendix D – Estimator.....		71
Appendix E – Kalman Code.....		72
Appendix F – Main Code		77

Figures List

[a] [b] [c] Figure 1 [a] Self-Balancing Robot [8], [b] Inverted Pendulum on a Cart [9], [c] Inverted Pendulum on a Linear Track [10]	5
Figure 2 Simplification steps of the inverted pendulum.....	6
Figure 3 Exposure of the simplified inverted pendulum	6
Figure 4 Diagram showing basic two-wheeled robot, with dimensions	8
Figure 5 A PID control system.....	17
Figure 6 A structure of a PID control system	17
Figure 7 Parallel Form of the PID Compensator	18
Figure 8 . a) Step response of PID ideal form b) Step response of PID real form	19
Figure 9 Electronic circuit implementation of an analog PID controller.....	22
Figure 10 Digital PID Controller.....	23
Figure 11 Gyroscope	26
Figure 12 Graph of Angle (calculated using equation 1) against Time	30
Figure 13 Schematic of an accelerometer	31
Figure 14 Diagram that shows the components of gravity in each direction [26].....	34
Figure 15 Graph of Angle (Accelerometer) against Time.....	35
Figure 16 Serial Communication Overview	36
Figure 17 Communication in I2C.....	37
Figure 18 I2C addressing	38
Figure 19.....	40
Figure 20.....	41
Figure 21 Fleming's Left-Hand Rule.....	43
Figure 22 : Torque production in a DC motor	43
Figure 23.....	44
Figure 24 Flow chart of the project.....	47
Figure 25 Block diagram of the hardware.....	48
Figure 26 Arduino Uno board	49
Figure 27 IMU board	50
Figure 28 I2C protocol	51
Figure 29 DC geared motor	52
Figure 30 DC Gear-Motor and Wheel Parameters	53
Figure 31 Schematic diagram for motor drive module.....	54
Figure 32 L298 Dual H-Bridge Board DC Motor Controller	55
Figure 33 Input Output OF The L298 Dual H-Bridge Board	55
Figure 34 Step Response	59
Figure 35 Synoptic diagram.....	59
Figure 36 DMP Data from MPU6050	62
Figure 37 PID = 20_0_0	64
Figure 38 PID = 15_190_1.6	64
Figure 39 PID = 70_240_1.9	64
Figure 40 PID = 15_190_1.6	65

Table List

Tableau 1 Previous technique to control two-wheel robots	14
Tableau 2 : A PID controller in a closed-loop system	16
Tableau 3 tuning methods of PID parameters	21
Tableau 4 Ziegler-Nicols method	22
Tableau 5 I2C Address.....	38
Tableau 6 Comparison between Accelerometer and Gyroscope in IMU	50
Tableau 7 Table DC gear-motor Specification.....	53
Tableau 8 Tires Parameter	53
Tableau 9 Effect of PID Tuning	58

list of variables and Abbreviations

x - process state vector.

u - piecewise continuous deterministic control input functions

F - system dynamics matrix.

B - deterministic input matrix.

G - noise input matrix.

z - discrete time measurement process.

H - measurement matrix.

v - discrete time white Gaussian noise.

w - Process noise

y_m Gyroscope measured velocity

y_r Real velocity

β Gyroscope bias

θ . Angular velocity

θ Angle

$m \tau$ - Motor torque (Nm).

$e \tau$ - Applied torque (Nm).

R – Nominal terminal resistance (Ohms).

L – Rotor inductance (H).

k_f – Frictional constant (Nms/rad).

k_m – Torque constant (Nm/A).

k_e – Back emf constant(Vs/rad).

θ - Angular position of shaft (rad).

ω - Angular velocity of shaft (rad/s).

α - Angular acceleration of shaft (rad/s²).

V_a – Applied terminal voltage (V).

V_e – Back emf voltage (V).

i – Current through armature (A).

I_R – Rotor inertia (kgm^2).

$m \tau$ - Motor torque (Nm).

M_w – mass of the wheel connected to both sides of the robot.

M_p – mass of the robot's chassis.

I_w – moment of inertia of the wheels.

I_p – moment of inertia of the robot's chassis.

H_L, H_R, P_L, P_R – reaction forces between the wheel and chassis.

l – distance between the centres of the wheel and the robot's centre of gravity.

C_L, C_R – applied torque from the motors to the wheels.

H_{fL}, H_{fR} – friction forces between the ground and the wheels.

θ_w - rotation angle of the wheels.

θ_p - rotation angle of the chassis.

$\dot{\theta}_w$ Angular velocity of wheel (rad/s)

V Applied terminal voltage (V)

$R(t)$ - Nominal terminal resistance (Ω)

k Back emf constant (V s/rad)

k Torque constant (N m/A)

$R(t)$ – Output.

K_p – Proportional gain.

K_i – Integral gain.

K_d – Derivative gain.

$e(t)$ – Error function

General introduction

General introduction

Self-balancing robots have been a topic of interest of many researchers, students and hobbyists worldwide. In essence, it is an inverted pendulum on wheels, a derivative of the inverted pendulum on a cart. Unlike traditional robots, which are in a constant state of equilibrium, the robot is a naturally unstable system. Its design is more complex, as it needs to be actively controlled to maintain its upright position, however, it benefits from being able to turn on the spot.

The primary practical application of a self-balancing robot is human transportation, which was popularized by the release of the Segway PT (Personal Transporter). It is used in many industries such as inside factory floors or for tourism in the park. It is more attractive compared to four or three wheeled vehicles as they can take sharp turns and navigate in tighter spaces.

The primary incentive of the project is to develop general understanding of control theory. For the last few decades, “the inverted pendulum has been the most popular benchmark, among others, for teaching and research in control theory and robotics [1].” Hence, developing a self-balancing robot is the ideal platform to put into practice what has been covered in Control Systems lectures. It would also be interesting to see the differences between the behavior in practice compared to simulations. Furthermore, the material and methods learnt have a wide array of applications; for example, inverted pendulums have been used to model human locomotion, which then was used to develop bipedal robots [2].

This project aims to design, construct and program a self-balancing robot with a self-developed and implemented PID controller. To achieve the aims of the project, following objectives have been set:

- Design and assemble the chassis of the robot.
- Develop the software to read from the sensors and to control the actuators
- Implement a PID controller to enable the robot to stay upright

In addition to control theory, learning about Kalman Filters (KF) was also a motivation to develop the self-balancing robot. Knowing the tilt angle is

General introduction

necessary in any balancing robot to apply the appropriate control action. However, different sensing devices have their compromises; the KF is used to fuse data from two sensors, such that a better estimate of the tilt angle can be obtained. Kalman filters specifically and not just the Complementary Filter, because it is considered to be “one of the most important data fusion algorithms in use today”, and it was famously used in the first manned mission to the moon [3]

Chapter 1:
Background Information and
Literature Review

Chapter 1: Background Information and Literature Review

I.1 Introduction

This chapter aims to provide an overview of the literature sources used throughout the development of the robot. First, the fundamentals of inverted pendulum systems are described: determination of the equilibrium point and what makes the system interesting to control engineers. Afterwards, a literature review on the most common control theory applied to self-balancing robots is performed and the last sub-section summarizes the main sensor fusion techniques.

I.2 Inverted Pendulum Systems

The inverted pendulum is a classical problem in control systems, and to explore the unstable dynamics, different platforms have been developed. These platforms are similar in many ways, leading to many of the behaviors being comparable. The most common types are the self-balancing robot, Inverted Pendulum on a cart and an inverted pendulum on a linear track, shown in the figure below:



[a]

[b]

[c]

Figure 1 [a] Self-Balancing Robot [8], [b] Inverted Pendulum on a Cart [9], [c] Inverted Pendulum on a Linear Track [10]

I.3 Model definition

The physical problem of the balancing robot is well described by the widely analyzed inverted pendulum. It is commonly modelled as a rigid rod fastened by a frictionless joint to a rigid cart moving in one direction. The simplification that

Chapter 1: Background Information and Literature Review

the wheel base can be seen as a cart sliding on a frictionless surface was made. This model definition is inspired by Math Works tutorial about inverted pendulum [4].

See Figure 2.1 for the simplification steps in this project.

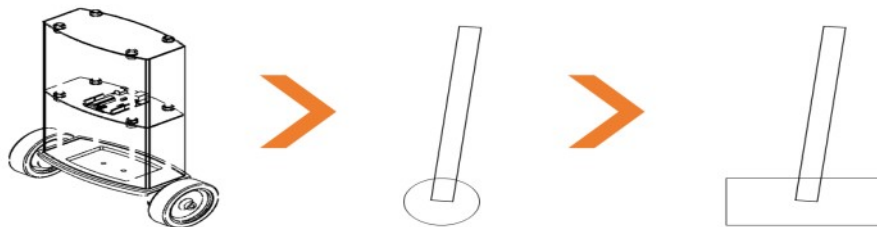


Figure 2 Simplification steps of the inverted pendulum

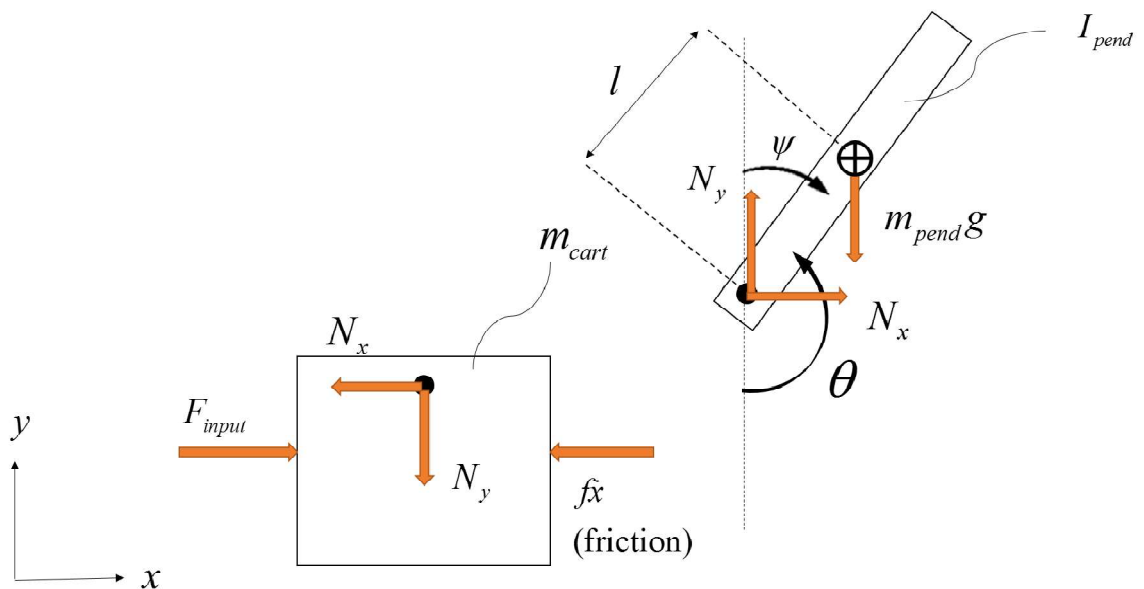


Figure 3 Exposure of the simplified inverted pendulum

I.4 Physical Model of the Robot

The two-wheeled robot consists of a long body with two wheels mounted at one end. For the simplicity of this derivation, the two wheels will be treated as a unit, and it will be assumed that the robot travels only in a straight line. By properly

Chapter 1: Background Information and Literature Review

designing the hardware, modeling can be simplified. Thus, the body can be treated as a point mass, rotating about the axis of the wheels.

The set of assumptions made to simplify modeling is as follows:

- The wheels are always in contact with the floor and experience rolling with no slip;
- The electrical and mechanical losses can be approximated to zero;
- The electrical system response is significantly faster than that of the mechanical system, so the dynamics of the electrical system may be neglected;
- The motion of the robot is constrained to a straight line, so that the system may be analyzed as a 2-dimensional system with only planar motion;
- All bodies are rigid;
- The angle of tilt from the vertical of the upper body is sufficiently small to allow linearization of the system ($\sin(\theta_2) \approx \theta_2$); And the angular velocity of the tilt of the upper body is sufficiently small ($\dot{\theta}_2 \approx 0$) such that the centrifugal force may be neglected.

Under these conditions, the robot can be modelled as shown in Figure 4.

The equations of motion resulting from this case are derived as the following:

$$\tau = H_1 \ddot{\theta}_1 + H_3 \ddot{\theta}_2 - m_2 r L \sin(\theta_2) \dot{\theta}_2^2 \quad (1)$$

$$-\tau = H_3 \ddot{\theta}_1 + H_2 \ddot{\theta}_2 - m_2 g L \sin(\theta_2) \quad (2)$$

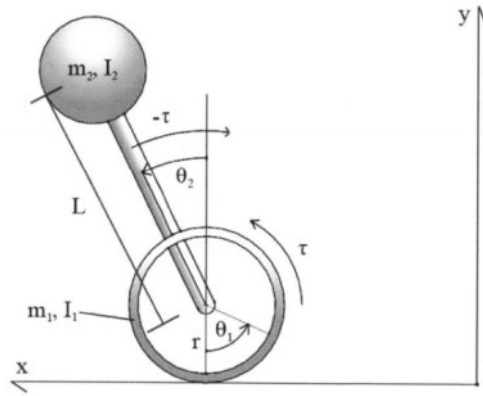


Figure 4 Diagram showing basic two-wheeled robot, with dimensions

where H_1 , H_2 , and H_3 are given by the matrix

$$H = \begin{bmatrix} H_1 & H_2 \\ H_2 & H_3 \end{bmatrix} = \begin{bmatrix} (m_1 + m_2)r^2 + I_1 & m_2 r L \cos \theta_2 \\ m_2 r L \cos \theta_2 & m_2 L^2 + I_2 \end{bmatrix} \quad (3)$$

I_1 and I_2 are the rotational moments of inertia of the wheel and the robot body, respectively; r is the radius of the wheels; L is the length between the center of mass of the body and the wheel axis; m_1 and m_2 are the masses of the wheels and the robot body, respectively. Linearization of these equations, based on the conditions listed above, yields the following equations of motion:

$$\tau = H_3 \ddot{\theta}_1 + H_2 \ddot{\theta}_2 \quad (4)$$

$$-\tau = H_3 \ddot{\theta}_1 + H_2 \ddot{\theta}_2 - m_2 g L \theta_2 \quad (5)$$

1.5 Non-Minimum Phase Zeros and Transfer Function Analysis

Perhaps the most fascinating aspect of inverted pendulum systems is the notion of non-minimum phase zeros. Following [12], a summary of the consequences of the non-minimum phase zeros are given. As explained in the previous section, inverted pendulums are unstable; attempting to stabilize an unstable plant using feedback will give rise to non-minimum phase zeros. The effect of these zeros can be observed by a simple experiment of trying to balance a long stick on the palm of one's hand. The stick is analogous to the inverted pendulum and the hand to the cart. When attempting to move the hand to right,

Chapter 1: Background Information and Literature Review

initially one tends to move to the left briefly then move to right. The zero in the system's transfer function has caused an initial undershoot when a 'step response' is applied. The conclusive effect of non-minimum phase zeros will vary depending on the controller used, it may cause initial undershooting, direction reversals and/or overshooting.

Given the transfer function

$$G(s) = \frac{N(s)}{D(s)} \quad (6)$$

a zero is the root in the numerator, $N(s)$, of $G(s)$. A non-minimum phase zero is a positive zero that is in the right half of the pole-zero plot. To exhibit the behaviour, in which direction reversals occurs, has to be an odd number of zeros in the transfer function.

From the transfer function, further conclusions can be made. If there are positive poles, the plant is unstable and based on parity interlacing principle, if the plant has odd number of positive real poles to the right of the non-minimum poles the plant cannot be controlled by a stable controller. This is actually present in a linearized transfer function of an inverted pendulum on a cart. The overall effect of the non-minimum phase zeros is constraints in closed loop performance. They restrict the bandwidth and it causes a limited gain margin (suggesting limited robustness). The non-minimum phase zeros cannot be cancelled in practice as a small difference in the zero and poles will lead to instability in the plant.

1.6 Controllers

A control regulates the behaviour of other sub-systems. It needs to have all the knowledge about the current and subsequent stages of all systems that it controls. A Control System has two loops – open and closed. Robotic systems are now widely implemented in "Smart Manufacturing". Robotics found one of its first applications in the machine tool industry. Back then, the robots were designed to have a stiff-hand mechanism with each joint controlled separately as a SISO linear system.

Chapter 1: Background Information and Literature Review

Besides that, there was a Point-to-Point Control system in place for tasks such as material transferring, and Continuous-path tracking system to carry out tasks such as welding and spray-painting. [5] In today's world, robot control systems have greatly advanced in terms of technique, and vision systems have increased the purpose of robots. Layering the Control System helps increase the ability and flexibility of the robot. Human-robot teamwork has a crucial part in enhancing the efficiency of the mechanism in robots. Many complex tasks are achieved in real time by autonomous mobile robots.

Modern-day robotic control systems have revolutionized the production industry, making it super versatile and user-friendly. PID controllers are used in more than ninety percent of the industrial controllers. This is mainly because it allows for a variety of tuning rules that have been proposed in relevant literature, and most of these controllers can be adjusted on-site by using these rules.

Some controllers may also feature on-line automatic tuning capabilities and automatic tuning methods. Improvised form of PID control, like multi-degrees of freedom PID control and I-PD control are used in the industry. There have been massive advancements in the region of Artificial Neural Network (ANN), Fuzzy Logic theory (FL), Artificial Intelligence (AI), and evolutionary computational methods like Genetic Algorithm (GA), Particle Swarm Optimization (PSO), etc. are all intelligent computational techniques that have provided us a new array of solutions to problems in different control system. Linear Quadratic Regulator (LQR) is a modern feedback control method, which uses the state space approach to analyze systems.

At the present, the LQR has been used with a combination of PID and FLC to help control a two-wheeled, self-balancing robot.

1.7 Types of controllers

To balance two wheel robot there are a few types of the controller that have been developed by other researchers. Each of the controllers has its own method of designing which contain advantages and disadvantages. Ali. Unluturk [12] states that balancing a two-wheeled mobile robot is feasible by using PID control method.

Chapter 1: Background Information and Literature Review

The aim of the study is to enable the robot to balance in upright position. As the result of the study, it has been concluded that P controller is not 11 enough for robot's balance. When the appropriate K_p and K_i gain values are chosen for PI controller, it has been observed that the robot can balance itself for a short time and try to maintain its balance by swinging.

In addition, when PID controller is applied, the two-wheel robot can stand in upright position longer compare to the previous two cases. This can be happen if only appropriate value of K_p , K_i , and K_d gain are chosen. Meanwhile, Nasir [13] states that PID controller capable to control the nonlinear inverted pendulum system angular and linear position in Matlab Simulink.

However, PID controller should be improved so that the maximum overshoot for the linear and angular positions do not have high range as required by the design. W.An [14] claim that Matlab can be used to compare the performance of PID Controller and Linear-quadratic regulator (LQR) in controlling two-wheeled self-balancing robot. It is concluded that LQR has a better performance than PID in self-balancing control in term of the time to achieve the steady state of robot.

Meanwhile, according to J. Wu [1] that the fuzzy logic algorithm can realize the self-balance control of the two-wheeled robot and restrain the robot from falling down. In their paper, a fuzzy controller is applied to the dynamic model of the two-wheeled self-balancing robot.

The simulation shows that the control methods have good performance in maintaining stability which resulting in short settling time and low overshoot. The linear controllers are more popular among researcher designing similar balancing robots like JOE [7].

Linear state space controllers like the Pole placement controller and the Linear Quadratic Regulators (LQR) are the two most popular control system implemented. In the paper of Two Wheels Mobile Robot Using Optimal Regulator Control, N. M. Abdul Ghani [15] mentioned that Pole placement gives best performance in term of settling time and magnitude for position, speed, angle and angle rate of two wheels' mobile robot.

Chapter 1: Background Information and Literature Review

A comparison of the results has demonstrated that Pole placement control provide higher level of disturbance reduction as compared to LQR technique. In other paper the author [16] mentioned that LQR design techniques can stabilize the balancing robot in the presence of large deviation angles with a better performance than PD design techniques based on simulation result. W. An [14] clarifies 12 that LQR technique has a better efficiency than PID controller in self-balancing control based on the simulation result. But this technique is only work on simulation and for future research is to use LQR method to control a physical two-wheeled robot to verify the controller's performance. The comparison of the performance of the controller is summarizing as shown in Table2.2 another method of controlling two wheel balancing robot is through SMC. Sliding mode control (SMC) is a robust technique to control nonlinear systems operating under uncertainty conditions [17]. SMC provides an effective alternative to deal with uncertain dynamic systems, and has been successfully applied in many engineering fields [18].

Based on the model proposed by Grasser et al. [7], Nawawi et al. [19] designed a proportional integral robust controller to achieve the robust stabilization and disturbance rejection of a two wheel self-balancing robot.

The simulation result is successfully shown that two-wheel balancing robot using SMC has a good response to achieve the desired characteristic compare to pole-placement. J. Ha et al.[6] stated that the SMC is able to work with the system without linearization and the controller can guarantee robustness and performance even at the point far from equilibrium. Nasir et al.[20] use Sliding Mode Controller for a nonlinear inverted pendulum system and compare the performance with PID controller. In this article they show that simulation results show that SMC controller has better performance compared to PID controller in controlling the nonlinear inverted pendulum system.

Previous Technique	Journal	Comment
LQR	Simulation and Control of a Two wheeled Self balancing Robot [14]	This technique works very well on the simulation and has a better performance over

Chapter 1: Background Information and Literature Review

		PID in stabilizing but yet not implemented in the real object.
FUZZY	Design of fuzzy logic controller for two-wheeled self-balancing robot [1]	The simulation show that fuzzy logic controller has a good performance in maintaining stability, yielding short settling time and also low overshoot Fuzzy logic controller provides higher robustness compare with using the method pole placement method. However, this method is not yet implemented on the hardware.
POLE PLACEMENTS	Two wheels mobile robot using optimal regulator control [15]	Pole placement gives best performance in term of settling time and magnitude for position, speed, angle and angle rate of two wheels' mobile robot as compared to the LQR technique. But the author still not implemented this technique on the hardware.
SLIDING MODE CONTROL SMC	Controller Design for Two-wheels Inverted	The simulation result is successfully shown that

Chapter 1: Background Information and Literature Review

	Pendulum Mobile Robot Using PISMC [19]	two-wheel balancing robot using SMC has a good response to achieve the desired characteristic compare to pole-placement. Again, this technique only feasible for simulation and not been tested for real hardware.
--	---	--

Tableau 1 Previous technique to control two-wheel robots

1.8 PID Controller

Feedback control is a control mechanism that uses information from measurements. In a feedback control system, the output is sensed. There are two main types of feedback control systems: 1) positive feedback 2) negative feedback. The positive feedback is used to increase the size of the input but in a negative feedback, the feedback is used to decrease the size of the input. The negative systems are usually stable.

A PID is widely used in feedback control of industrial processes on the market in 1939 and has remained the most widely used controller in process control until today.

Thus, the PID controller can be understood as a controller that takes the present, the past, and the future of the error into consideration. After digital implementation was introduced, a certain change of the structure of the control system was proposed and has been adopted in many applications. But that change does not influence the essential part of the analysis and design of PID controllers.

A proportional–integral–derivative controller (PID controller) is a method of the control loop feedback. This method is composing of three controllers [1]:

- a) Proportional controller (PC)
- b) Integral controller (IC)

Chapter 1: Background Information and Literature Review

c) Derivative controller (DC)

a Role of a Proportional Controller (PC)

The role of a proportional depends on the present error, I on the accumulation of past error and D on prediction of future error.

The weighted sum of these three actions is used to adjust Proportional control is a simple and widely used method of control for many kinds of systems. In a proportional controller, steady state error tends to depend inversely upon the proportional gain (ie: if the gain is made larger the error goes down). The proportional response can be adjusted by multiplying the error by a constant K_p , called the proportional gain.

The proportional term is given by:

$$P = K_p \cdot \text{Error}(t) \quad (7)$$

A high proportional gain results in a large change in the output for a given change in the error. If the proportional gain is very high, the system can become unstable. In contrast, a small gain results in a small output response to a large input error. If the proportional gain is very low, the control action may be too small when responding to system disturbances. Consequently, a proportional controller (K_p) will have the effect of reducing the rise time and will reduce, but never eliminate, the steady-state error. In practice the proportional band (PB) is expressed as a percentage so:

$$PB\% = \frac{100}{K_p} \quad (8)$$

Thus a PB of

$$10\% \Leftrightarrow K_p = 10 \quad (9)$$

b Role of an Integral Controller (IC)

An Integral controller (IC) 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. Consequently, an integral control (K_i) will have

Chapter 1: Background Information and Literature Review

the effect of eliminating the steady-state error, but it may make the transient response worse. The integral term is given by:

$$I = K_I \int_0^t error(t) dt \quad (10)$$

c Role of a Derivative Controller (DC)

The derivative of the process error is calculated by determining the slope of the error over time and multiplying this rate of change by the derivative gain K_d . The derivative term slows the rate of change of the controller output. A derivative control (K_d) will have the effect of increasing the stability of the system, reducing the overshoot, and improving the transient response. The derivative term is given by:

$$D = K_D \frac{derro (t)}{dt} \quad (11)$$

Effects of each of controllers K_p , K_d , and K_i on a closed-loop system are summarized in the table shown below in tableau 1.

Parameter	Rise Time	Overshoot	Settling Time	Steady-State Error
K_p	Decrease	Increase	Small change	Decrease
K_i	Decrease	Increase	Increase	Decrease significantly
K_d	Minor Decrease	Minor Decrease	Minor Decrease	No effect in theory

Tableau 2 : A PID controller in a closed-loop system

d PID controller (PIDC)

A typical structure of a PID control system is shown in Fig.1. Fig.2 shows a structure of a PID control system. The error signal $e(t)$ is used to generate the proportional, integral, and derivative actions, with the resulting signals weighted and summed to form the control signal $u(t)$ applied to the plant model.

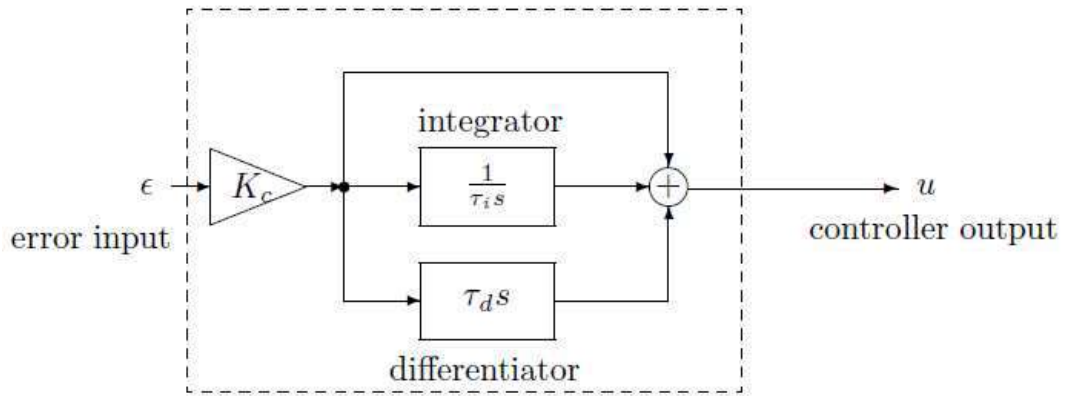


Figure 5 A PID control system

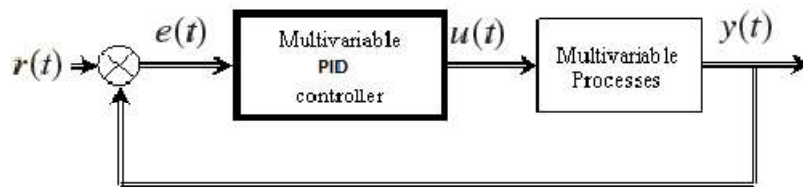


Figure 6 A structure of a PID control system

Where $u(t)$ is the input signal to the multivariable processes, the error signal $e(t)$ is defined as $e(t) = r(t) - y(t)$, $r(t)$ and is the reference input signal. A standard PID controller structure is also known as the “three-term” controller. This principle mode of action of the PID controller can be explained by the parallel connection of the P, I and D elements shown in Figure 3. Block diagram of the PID controller

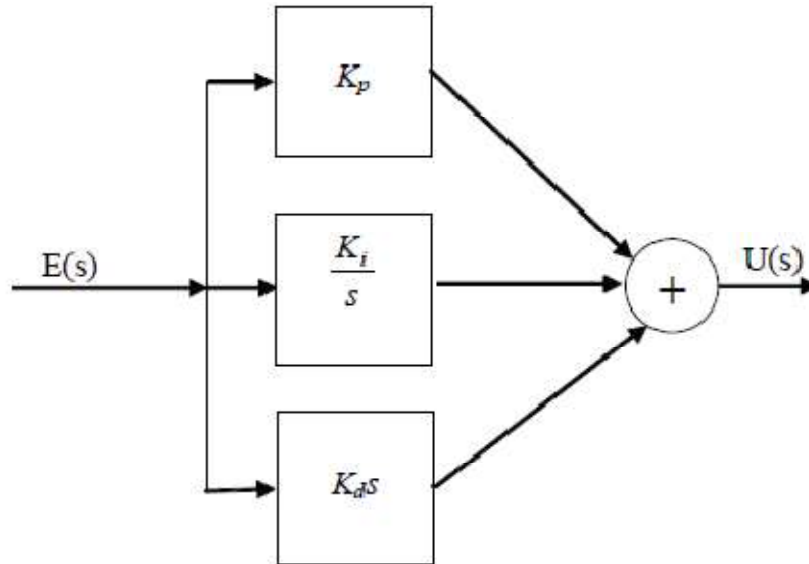
$$G(s) = K_p \left(1 + \frac{1+T_I \cdot T_D \cdot s^2}{T_I \cdot s} \right) = K_p \left(1 + \frac{1}{T_I \cdot s} + T_D \cdot s \right) \quad (12)$$

where K_p is the proportional gain, T_I is the integral time constant, T_D is the derivative time constant, $K_i = \frac{K_p}{T_I}$ is the integral gain and $K_D = K_p T_D$ is the derivative gain. The “threeterm” functionalities are highlighted below. The terms K_p , T_I and T_D definitions are:

- ✓ The proportional term: providing an overall control action proportional to the error signal through the all pass gain factor.
- ✓ The integral term: reducing steady state errors through low frequency compensation by an integrator.

Chapter 1: Background Information and Literature Review

- ✓ The derivative term: improving transient response through high frequency compensation by a differentiator.



Or

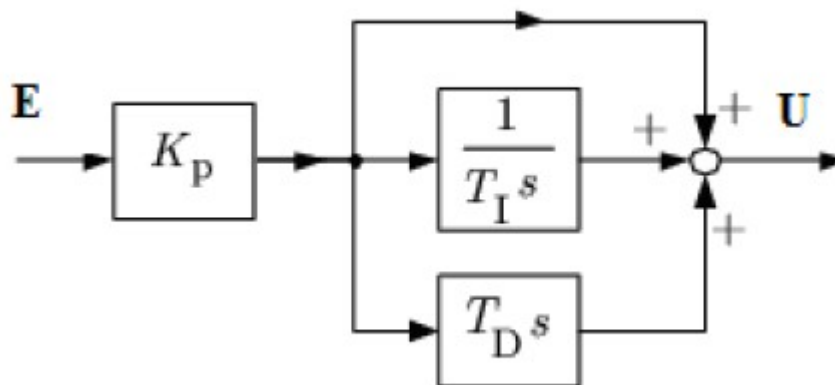


Figure 7 Parallel Form of the PID Compensator

These three variables K_p , T_I and T_D are usually tuned within given ranges. Therefore, they are often called the *tuning parameters* of the controller. By proper choice of these tuning parameters a controller can be adapted for a specific plant to obtain a good behaviour of the controlled system.

The time response of the controller output is

$$U(t) = K_p(e(t) + \frac{\int_0^t e(t)dt}{T_I} + T_d \frac{de(t)}{dt}) \quad (13)$$

Chapter 1: Background Information and Literature Review

Using this relationship for a step input of $e(t)$, i.e. $e(t) = \square(t)$, the step response $r(t)$ of the PID controller can be easily determined. The result is shown in below. One has to observe that the length of the arrow K_{PTD} of the D action is only a measure of the weight of the \square impulse.

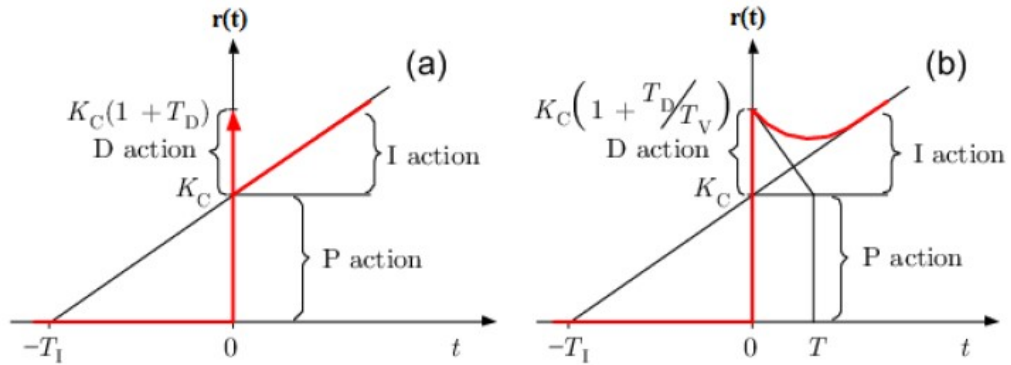


Figure 8 . a) Step response of PID ideal form b) Step response of PID real form

1.9 The transfer function of the PID controller

The transfer function of the PID controller is

$$G(s) = \frac{U(s)}{E(s)} \quad (14)$$

$$G(s) = K_p + \frac{K_I}{s} + K_D s = \frac{K_D s^2 + K_P s + K_I}{s} \quad (15)$$

1.10 PID pole zero cancellation

The PID equation can be written in this form:

$$G(s) = \frac{K_d(s^2 + \frac{K_P}{K_d}s + \frac{K_I}{K_d})}{s} \quad (16)$$

When this form is used it is easy to determine the closed loop transfer function.

Chapter 1: Background Information and Literature Review

$$H(s) = \frac{1}{s^2 + 2\varepsilon\omega_0 s + \omega_0^2} \quad (17)$$

If

$$\frac{K_i}{K_d} = \omega_0^2 \quad (18)$$

$$\frac{K_i}{K_d} = 2\varepsilon\omega_0 \quad (19)$$

Then

$$G(s)H(s) = \frac{K_d}{s} \quad (20)$$

This can be very useful to remove unstable poles.

There are several prescriptive rules used in PID tuning. The most effective methods generally involve the development of some form of process model, and then choosing P, I, and D based on the dynamic model parameters.

Method	Advantages	Disadvantages
Manual	Online method No math expression	Requires experienced personnel
Ziegler-Nichols	Online method Proven method	Some trial and error, process upset and very aggressive tuning
Cohen-Coon	Good process models	Offline method Some math

Chapter 1: Background Information and Literature Review

		Good only for first order processes
Software tools	Online or offline method, consistent tuning, Support Non-Steady State tuning	Some cost and training involved
Algorithmic	Online or offline method, Consistent tuning, Support Non-Steady State tuning, Very precise	Very slow

Tableau 3 tuning methods of PID parameters

1.11 The Ziegler–Nichols tuning method

The Ziegler–Nichols tuning method is an empirical method of tuning a PID controller. It was proposed by John G. Ziegler and Nichols in the 1940's. It is performed by setting I (integral) and D (derivative) gains to zero. The P (proportional) gain, K_p is then increased (from zero) until it reaches the ultimate gain K_u , at which the output of the control loop oscillates with a constant amplitude. K_u and the oscillation period T_u are used to set the P, I, and D gains depending on the type of controller used [6,7]:

Chapter 1: Background Information and Literature Review

Ziegler–Nichols method			
Control Type	K_p	K_i	K_d
P	$K_u / 2$	-	-
PI	$K_u / 2.2$	$1.2K_p / T_u$	-
PID	$0.60K_u$	$2K_p / T_u$	$K_p T_u / 8$
Some overshoot	$0.33K_u$	$2K_p / T_u$	$K_p T_u / 3$
No overshoot	$0.2K_u$	$2K_p / T_u$	$K_p T_u / 3$

Tableau 4 Ziegler-Nicols method

We can realize a PID controller by two methods:

First, an analog PID controller

Second, a digital PID controller

1. Circuit diagram below (figure.5) shows an analog PID controller. In this figure, we present an analog PID controller with three simple op amp amplifier, integrator and differentiator circuits.

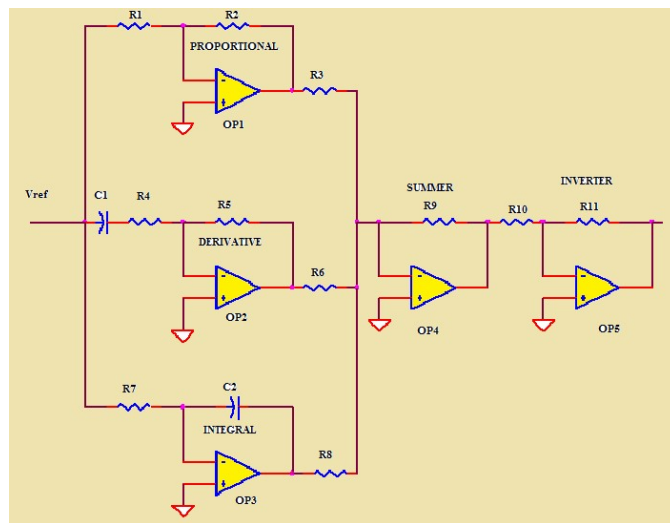


Figure 9 Electronic circuit implementation of an analog PID controller

TERM	DEFINITION	Op Amp Circuit Function
P	Proportional/ Amplifier	$V_o = (R_2/R_1) \cdot V_{ref}$
D	Differentiator	$V_o = R_5 \cdot C_1 \cdot dV_{ref}/dt$
I	Integrator	$V_o = 1/(R_7 \cdot C_2) \cdot \int V_{ref} dt$

Chapter 1: Background Information and Literature Review

Finally, we need to add the three PID terms together. Again the summing amplifier OP4 serves us well. Because the error amp, PID and summing circuits are inverting types, we need to add a final op amp inverter OP5 to make the final output positive.

2. Today, digital controllers are being used in many large and small-scale control systems, replacing the analog controllers. It is now a common practice to implement PID controllers in its digital version, which means that they operate in discrete time domain and deal with analog signals quantized in a limited number of levels. Moreover, in such controller we do not need much space and they are not expensive. A digital version of the PID controller is shown in figure 6 [8,9].

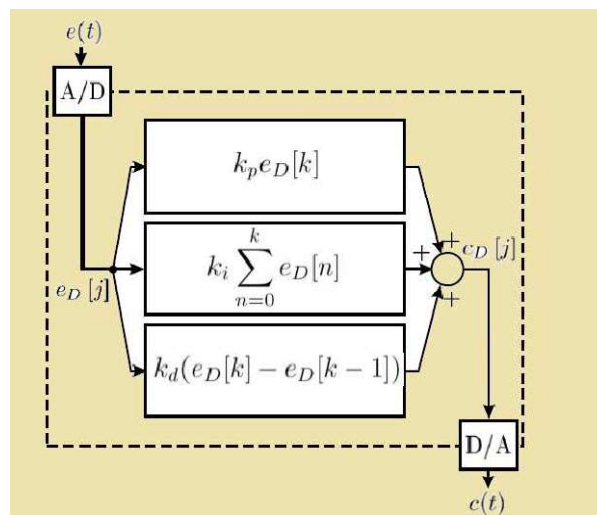


Figure 10 Digital PID Controller

In its digital version, the integral becomes a sum and the differential a difference. The continuous time signal $e(t)$ is sampled in fixed time intervals equals a determined sample period, here called T_c (in figure 10 $T_c = 1$). An A/D (analog to digital) converter interfaces the input and a D/A (digital to analog) converter interfaces the output. This sampled and digitalized input, called $e_D[j]$, exists only in time instants $t = kT_c$ for all $k \geq 0 \in \mathbb{Z}$. A lower bound for the sample period is the computing time of a whole cycle of the digital PID (which includes the A/D and D/A conversion).

While PID controllers are applicable to many control problems, and often perform satisfactorily without any improvements or even tuning, they can

Chapter 1: Background Information and Literature Review

perform poorly in some applications, and do not in general provide optimal control.

I.12 Sensors:

I.13 Overview of Gyroscopes

a Definition

Gyroscopes are physical sensors that detect and measure the angular motion of an object relative to an inertial frame of reference. The term "Gyroscope" is attributed to the mid-19th century French physicist Leon Foucault who named his experimental apparatus for Earth's rotation observation by joining two Greek roots: gyros - rotation and skopeein - to see. Unlike rotary encoders or other sensors of relative angular motion, the unique feature of gyroscopes is the ability to measure the absolute motion of an object without any external infrastructure or reference signals. Gyroscopes allow untethered tracking of an object's angular motion and orientation and enable standalone Heading Reference Systems (AHRS). Combining 3 gyroscopes with 3 accelerometers in a complete 6-axis Inertial Measurement Unit (IMU) enables self-contained Inertial Navigation Systems (INS) for navigation, guidance, and dead reckoning. All gyroscopes can be divided into two main categories, depending on whether the angular velocity or orientation is being measured [1]. Rate gyroscopes measure the angular velocity, or the rate of rotation of an object. Angle gyroscopes, also called Whole Angle or Rate Integrating gyroscopes, measure the angular position, or orientation of an object directly. While devices sensitive to the angular acceleration are used in some applications, these sensors are typically not referred to as gyroscopes, but rather as angular accelerometers. Essentially all existing Micro-Electro-Mechanical-Systems (MEMS) gyroscopes are of the rate measuring type and are typically employed for motion detection (for example, in consumer electronics and automotive safety devices) and motion stabilization and control (for example, in smart automotive steering and antenna/camera stabilization systems).

True INS (inertial navigation system) and AHRS (Attitude and Heading Reference System) applications rely on the continuous tracking of the object's

Chapter 1: Background Information and Literature Review

orientation. Measurement of the angular position can be accomplished either by numerical integration of a rate gyroscope's output, or by using an angle gyroscope which effectively integrates the rotation rate by virtue of its internal dynamics and outputs the angle information directly. When a rate gyroscope is used to track the orientation, its output signal is integrated over time together with the associated errors and noise, leading to fast buildup of the orientation angle drifts (for example, white noise in the angular rate signal results in $1/f^2$ drift, or random walk, of angle).

Successful realization of standalone gyroscope-based INS and AHRS thus requires either angle gyroscopes or rate gyroscopes with extremely stable output and very low noise.

b Historical Overview of Vibratory Gyroscope Technologies

Early work by Leon Foucault during the mid-19th century explored two different design paradigms for angle measuring mechanical gyroscope based on either a spinning or vibrating mass. While the spinning mass approach was the dominant method of mechanical gyroscope construction from its inception well into the second half of the 20th century, it is not well suited for MEMS implementation due to the technological limitations in the manufacturing of precision, low friction bearings. Few designs of spinning mass MEMS gyroscopes using electrostatic levitation have been reported in the literature [2, 3] without yet achieving commercial success due to the inherent instability of the mechanical system and necessity for a sophisticated control system.

The vibrating mass approach, illustrated by the popular Foucault pendulum experiment, exploits the exchange of energy between different axis of vibration due to the Coriolis effect. This architecture, at present referred to as the Coriolis Vibratory Gyroscope (CVG) remained largely a scientific curiosity for almost a century until the introduction of a functional vibratory gyroscope by Sperry in the mid-20th century [5] followed by successful commercialization of quartz tuning fork gyroscopes by BEI Technologies in the late-20th century [6], and very high volume deployment of silicon MEMS CVGs in the early 21st century. Today, silicon vibratory rate gyroscopes with capacitive transduction comprise the majority of MEMS gyroscopes in development and production, with some

Chapter 1: Background Information and Literature Review

research groups and manufacturers pursuing quartz devices with piezoelectric transduction or silicon devices with alternative transduction mechanisms such as inductive or electromagnetic [7]. **Figure 1** shows photographs of a wafer-level batch fabricated silicon-on-insulator (SOI) gyroscope with capacitive actuation and detection designed, fabricated, and packaged at the University of California, Irvine [8].

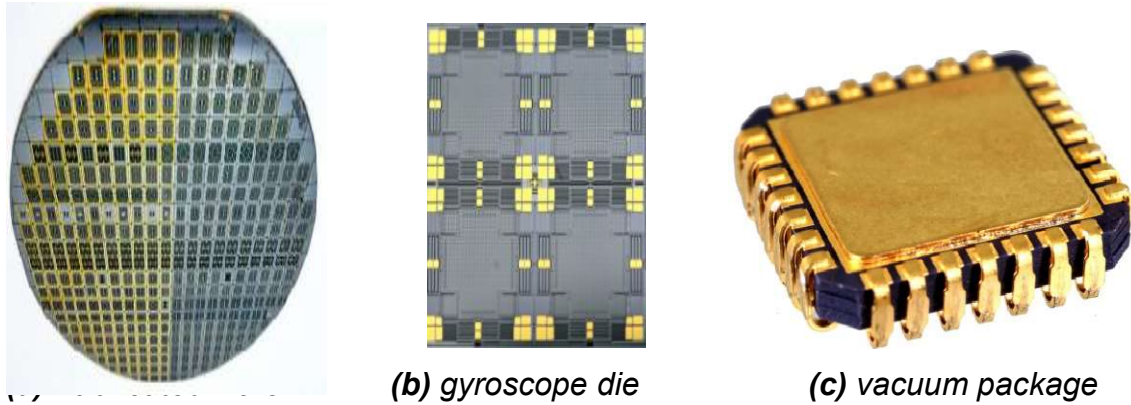


Figure 11 Gyroscope

c Vibratory Gyroscope Dynamics

In this section, the principles of operation of vibratory gyroscopes are derived from the basic concepts of classical mechanics. While a planar, z-axis vibratory gyroscope is the focus of the explanation, the discussions are generic in nature and equally apply to devices with other architectures, including torsional, out-of-plane, and sensors with multiple modes of vibration for simultaneous detection of rotation in several axes. Let $S\xi\eta\zeta$ denote an inertial reference frame, in which a non-inertial frame $Oxyz$ is moving with a linear acceleration $A_0 = (A_{0x}, A_{0y}, A_{0z})$ and an angular velocity $\Omega = (\Omega_x, \Omega_y, \Omega_z)$ where the components of the vectors are given with respect to the moving frame. Coordinates of a point mass in the inertial and non-inertial reference frames are given by vectors $\rho = (\xi, \eta, \zeta)$ and $r = (x, y, z)$, respectively. Newton's equation of motion of the point mass m under the action of force vector $F = (F_x, F_y, F_z)$ with respect to the inertial reference frame $S\xi\eta\zeta$ are given by $\rho'' = A_0 + \Omega[\Omega r] + [\Omega r'] = F/m$, where the prime denotes derivative with respect to time. According to the rules of differentiation in a moving frame,

$$r'' = A_{rel} = F/m - 2[\Omega r'] - (A_0 + [\Omega[\Omega r]] + [\Omega r']) \quad (21)$$

Chapter 1: Background Information and Literature Review

This differential vector equation provides the strict mathematical foundation for the Coriolis vibratory gyroscopes. The Coriolis cross-product of the angular velocity Ω and the coordinate vector $r=(x, y, z)$ governs the coupling and exchange of energy between x, y, z non-inertial axes. This effect allows measuring the input angular velocity Ω by observing the vibration pattern of the proof mass m relative to the non-inertial device reference frame $Oxyz$.

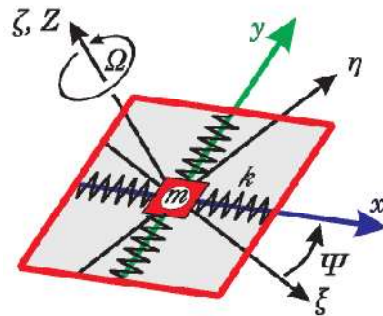
A theoretical model of a single axis Coriolis vibratory gyroscopes is shown in **Figure 2**. A point mass m forms the Coriolis sensitive proof mass and is constrained to motion in the Oxy plane under the influence of elastic forces of the suspension and inertial forces, caused by the motion of the $Oxyz$ non-inertial reference frame. To simplify the conceptual discussion, it is assumed that the proof mass suspension produces a linear field of elastic forces $F_s = (kxx, kyy)$. Also neglected are the projection of the gravity field on the Oxy plane, out-of-plane Oz axis dynamics, and acceleration of the origin O . These assumptions are justified by high out-of-plane stiffness of typical bulk micro machined structures and frequency separation between the external and Coriolis accelerations. After linearization with respect to the input angular velocity vector Ω components $\Omega_x, \Omega_y, \Omega_z$,

Equation 2 can be simplified to the following:

$$x'' + \omega_x^2 x - y\Omega_z' - 2\Omega_z y' = F_x \quad (22)$$

$$y'' + \omega_y^2 y + x\Omega_z' + 2\Omega_z x' = F_y \quad (23)$$

where ω_x and ω_y are the natural frequencies of the x - and y -mode, respectively. The following discussion will consider two different modes of operation of vibratory gyroscopes - rate measuring and angle measuring.



Chapter 1: Background Information and Literature Review

Figure 2: Theoretical model of a single z-axis vibratory gyroscope consisting of a proof mass m suspended in the x - y plane. The xyz non-inertial frame of reference associated with the sensor is moving relative to the inertial frame $\xi\eta\zeta$ with an angular velocity $\Omega=(0, 0, \Omega_z)$. Coriolis force coupling between the x and y coordinates causes energy exchange which is used to detect the input rate Ω_z .

d Rate Gyroscope Operation

Rate gyroscopes are operated with inherent non-symmetry between the Ox axis, designated as the drive-mode, and the Oy axis, designated as the sense-mode. The drive-mode is operated in the forced vibrations mode, where the excitation force F_x is a sinusoidal waveform with amplitude f and angular frequency ωd . The sense-mode of a rate gyroscope can be operated either open-loop or in a force-to-rebalance closed loop, where a feedback force is generated to suppress the sense-mode vibrations and is simultaneously used as the measure of the input rate. Dynamics of a rate gyroscope with an open-loop sense-mode can be derived from **Equation 2** as

Equation 3:

$$x'' + x\omega x_2 - y\Omega_z' - 2\Omega_z y' = f/m \sin(\omega dt) \quad (24)$$

$$y'' + y\omega y_2 + x\Omega_z' + 2\Omega_z x' = 0 \quad (25)$$

In practical implementations of vibratory rate gyroscopes stabilization of the drive-mode velocity is desired to reduce the effect of manufacturing tolerances and operational conditions on the scale factor of the sensor. This is accomplished by an electronic feedback system which regulates the amplitude f of the driving force by means of an *Automatic Gain Control (AGC)* to maintain constant amplitude of the drive-mode motion. The frequency ωd of the drive-mode feedback force is generated by a *Phase-Lock Loop (PLL)* which locks to the drive-mode mechanical resonance frequency ωx . An alternative approach for drive-mode operation is through positive feedback of the velocity signal x' , which effectively de-stabilizes the drive-mode and maintains resonance of the drive-mode. In the velocity feedback configuration, the need for a separate frequency source is eliminated, however often at the cost of increased jitter and

Chapter 1: Background Information and Literature Review

noise in the Coriolis signal demodulation block. Simplified equations for a vibratory rate gyroscope with closed-loop drive-mode and open-loop sense-mode, see **Figure 3**, are derived from **Equation 3** as

Equation 4:

$$x = \sin(\omega x t) \quad (26)$$

$$y'' + y' \left(\frac{\omega y}{Q_y} \right) + y \omega y_2 = -2\Omega z x' \quad (27)$$

Here, an energy dissipation term, defined by the sense-mode quality factor Q_y , is now included in the sense-mode dynamics, while the term $x\Omega z'$ is omitted assuming slow-varying input Ωz .

Figure 4 illustrates the motion of the proof mass relative to the device reference frame in the presence of input rotation. The drive-mode is a resonator driven to a constant amplitude of vibrations at a fixed frequency by means of a feedback system. During rotation of the device, Coriolis effect causes coupling of energy from the drive-mode to the sense-mode. The Coriolis force applied to the sense-mode is given by the cross-product of and is proportional to the drive-mode velocity x' and the input angular rate Ωz . Since the drive-mode velocity is a sinusoidal signal with a fixed frequency, the Coriolis effect results in the *Amplitude Modulation (AM)* of the input angular rate by the drive-mode velocity. To measure the input angular rate, displacement (or, equivalently, velocity) of the Coriolis force induced sense-mode vibrations is measured by means of an amplitude.

e Gyroscope Fundamentals

The gyroscope measures angular velocity, in radians per second or degrees per second. Intuitively, by integrating the angular velocity the tilt angle can be calculated.

Since the gyroscope readings are taken at discrete time intervals, dt , numerical integration is performed using the Euler method. This is shown in the equation below: $gyroanglet = gyroanglet-1 + angularvelocity$ (28)

Chapter 1: Background Information and Literature Review

The equation above assumes the sensor initial position is 0 degrees. If the starting position of the sensor is non-zero, the angle has to be initialized to the value, possibly from another sensor i.e. inclinometer or accelerometer.

Using the approach mentioned the following graph was obtained:

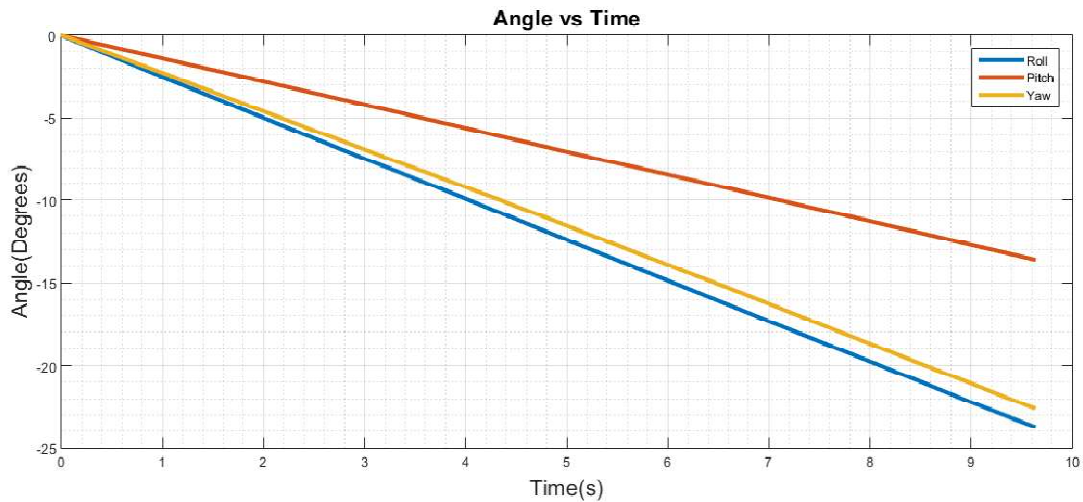


Figure 12 Graph of Angle (calculated using equation 1) against Time

A key element to observe was that the measurements were taken while the IMU was at rest; this highlights the effect of the bias in the measurements. Due to the integration, this systematic error is summed in every loop; thus the angle appears to be increasing even though the gyroscope was not moving. Furthermore, the size of the bias also increases over time [26]. In cases where the object can rotate in 3D, the yaw angle needs to be coupled with the roll and pitch to reduce the chances of Gimbal Lock [27]. The robot is assumed to have only one axis of rotation thus coupling does not need to be considered

1.14 Accelerometer

a Theory & Design

An accelerometer is a sensor that measures the physical acceleration experienced by an object due to inertial forces or due to mechanical excitation. In aerospace applications accelerometers are used along with gyroscopes for navigation guidance and flight control. Conceptually, an accelerometer behaves as a damped mass on a spring. When the accelerometer experiences acceleration, the mass is displaced and the displacement is then measured to give the acceleration

Chapter 1: Background Information and Literature Review

In these devices, piezoelectric, piezoresistive and capacitive techniques are commonly used to convert the mechanical motion into an electrical signal. Piezoelectric accelerometers rely on piezoceramics (e.g. lead zirconate titanate) or single crystals (e.g. quartz, tourmaline). They are unmatched in terms of their upper frequency range, low packaged weight and high temperature range. Piezoresistive accelerometers are preferred in high shock applications. Capacitive accelerometers performance is superior in low frequency range and they can be operated in servo mode to achieve high stability and linearity.

Modern accelerometers are often small micro electromechanical systems (MEMS), consisting of little more than a cantilever beam with a proof-mass (also known as seismic-mass) realized in single crystal silicon using surface micromachining or bulk micromachining processes.

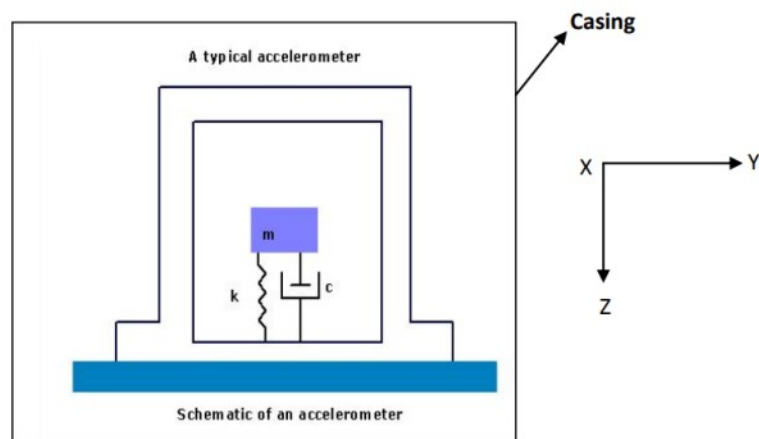


Figure 13 Schematic of an accelerometer

The principle of working of an accelerometer can be explained by a simple mass (m) attached to a spring of stiffness (k) that in turn is attached to a casing, as illustrated in fig 2.1. The mass used in accelerometers is often called the seismic-mass or proof-mass. In most cases the system also includes a dashpot to provide a desirable damping effect. The principle of working of an accelerometer can be explained by a simple mass (m) attached to a spring of stiffness (k) that in turn is attached to a casing, as illustrated in fig 2.1. The mass used in accelerometers is often called the seismic-mass or proof-mass. In most cases the system also includes a dashpot to provide a desirable damping effect.

Chapter 1: Background Information and Literature Review

The dashpot with damping coefficient (c) is normally attached to the mass in parallel with the spring. When the spring mass system is subjected to linear acceleration, a force equal to mass times acceleration acts on the proof-mass, causing it to deflect. This deflection is sensed by a suitable means and converted into an equivalent electrical signal. Some form of damping is required, otherwise the system would not stabilize quickly under applied acceleration. To derive the motion equation of the system Newton's second law is used, where all real forces acting on the proof-mass are equal to the inertia force on the proof-mass. Accordingly a dynamic problem can be treated as a problem of static equilibrium and the equation of motion can be obtained by direct formulation of the equations of equilibrium. This damped mass-spring system with applied force constitutes a classical second order mechanical system. From the stationary observer's point of view, the sum of all forces in the z direction is,

$$F_{applied} - F_{damping} - F_{spring} = m\ddot{x} \quad (29)$$

$$m\ddot{x} + F_{damping} + F_{spring} = F_{applied} \quad (30)$$

$$m\ddot{x} + kx + \dot{x}x = F \quad (31)$$

Where m = mass of the proof-mass x = relative movement of the proof-mass with respect to frame c = damping coefficient k = spring constant F = force applied The equation of motion is a second order linear differential equation with constant coefficients. The general solution $x(t)$ is the sum of the complementary function $X_c(t)$ and the particular integral $X_p(t)$

$$X(t) = X_c(t) + X_p(t) \quad (32)$$

The complementary function satisfies the homogeneous equation

$$m\ddot{x} + kx + c\dot{x} = 0 \quad (33)$$

The solution to $X_c(t)$ be

$$X_c(t) = Ce^{st} \quad (34)$$

Substituting

$$(ms^2 + cs + k)Ce^{st} = 0 \quad (35)$$

Chapter 1: Background Information and Literature Review

As Ce^{st} cannot be zero for all values of t , then $(ms^2 + cs + k) = 0$ called as the auxiliary or characteristic equation of the system. The solution to this equation for values of S is

$$S_{1,2} = \frac{-c \pm \sqrt{c^2 - 4mk}}{2m} \quad (36)$$

From the above equation 36, the following useful formulae are derived

$$\omega_n = \sqrt{mk}, \quad c/m = 2\varepsilon\omega_n \quad \text{and} \quad \varepsilon = c/2\sqrt{km} \quad (37)$$

Where ω_n : undamped resonance frequency, k : spring constant m : mass of proof-mass, c : damping coefficient and ε : damping factor Steady state performance In the steady state condition, that is, with excitation acceleration amplitude „ a “ and frequency „ ω “, the amplitude of the response is constant and is a function of excitation amplitude and frequency „ ω “. Thus for static response „ ω “=0, the deflection amplitude

$$X = X_0 = \frac{F}{k} \cdot X = \frac{m a}{k} \quad (38)$$

Here the sensitivity „ S “ of an accelerometer is defined by, $S = X / a = m/k$
Dynamic performance For the dynamic performance it is easier to consider the Laplace transform of equation (37) It can be seen by comparing equation (26) and (21) that the bandwidth of an accelerometer sensing element has to be traded off with its sensitivity since $S = 1/\omega_n^2$ (this trade off can be partly overcome by applying feedback, i.e closed loop scheme). The sensor response is determined by damping present in the system. A damping factor (ζ) between 0.6 to 1.2 results in high response time, fast settling time, good bandwidth and linearity

b Accelerometer Basic Principles

The accelerometer measures the acceleration relative to free fall. The acceleration is often measured in g s, which is based on earth's gravitational pull (9.81m/s). To determine the orientation of the accelerometer, it is assumed that the only force acting on the object is earth's gravitation pull. Gravity always acts 'down', thus when the object is tilted; the force is divided into components in the x , y , and z directions of the object. Since the axes are orthogonal to one

Chapter 1: Background Information and Literature Review

another, Pythagoras theorem can be used to show the relationship between the forces as shown in the diagram below:

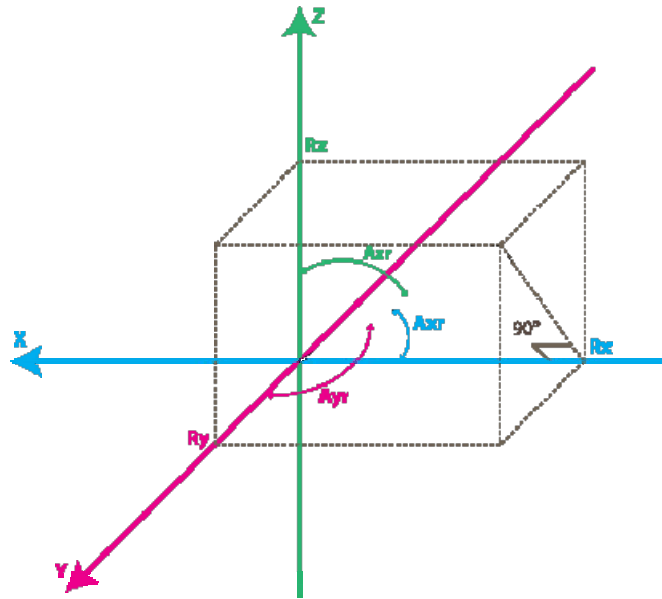


Figure 14 Diagram that shows the components of gravity in each direction [26]

From the diagram, the following equation can be obtained:

$$A_{xr} = \theta_x = \cos^{-1}(x\sqrt{x_2^2 + y_2 + z_2}) = \cos^{-1}(xR) \quad (39)$$

Alternatively, the following equation is more useful as it calculates the angle relative to the z-axis (Roll) [26]:

$$Roll = \phi_{xyz} = \tan^{-1}(yz) \quad (40)$$

With a similar approach, Pitch can also be calculated. However, Yaw cannot be determined accurately, especially when the force in the z direction = 1g (assuming the accelerometer can only rotate and not translate in any direction). In this case changing the yaw, will have no impact on the components of x and y, making yaw constant. This however does not affect the implementation in the robot as only the tilt angle is required [26]. Using equation 6, the following graph was obtained:

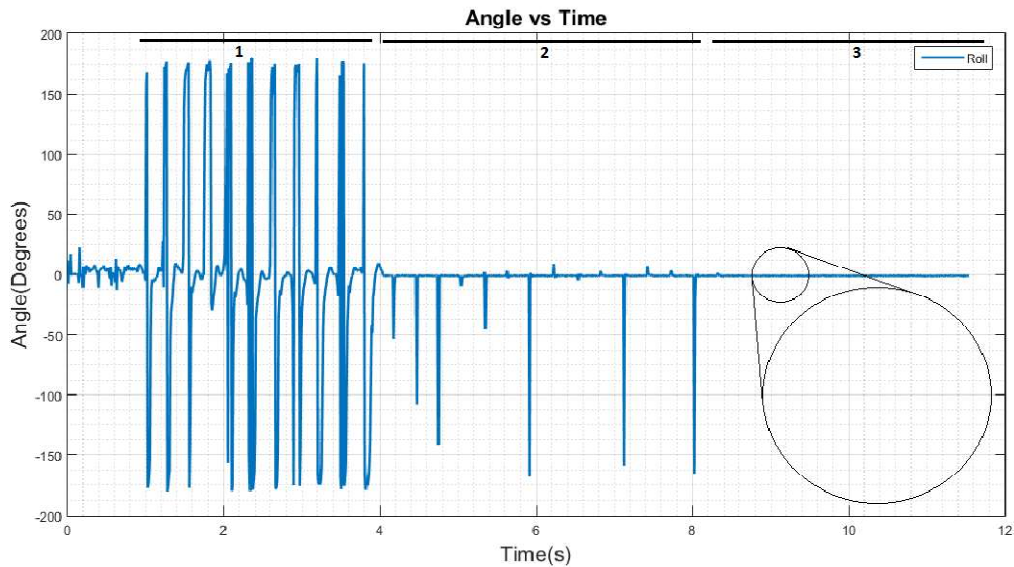


Figure 15 Graph of Angle (Accelerometer) against Time

The graph above is divided into 3 sections. In all cases, the IMU orientation was kept constant (as well as humanly possible). In section 1, the IMU was moved up and down. In section 2, the table was tapped lightly and in section 3, the accelerometer was left without external disturbances. The diagram shows the problems with using accelerometer for measuring angles, even in section 3, when zoomed in it can be see that the line is still not perfectly smooth, emphasizing the noisy nature of the accelerometer. The root of these issues is the assumption that gravity is the only force acting upon the object [29].

I.15 Interfacing the Arduino MPU 6050

a Introduction

The MPU 6050 communicates with the Arduino through the I2C protocol. The MPU 6050 is connected to Arduino as shown in the following diagram. If your MPU 6050 module has a 5V pin, then you can connect it to your Arduino's 5V pin. If not, you will have to connect it to the 3.3V pin. Next, the GND of the Arduino is connected to the GND of the MPU 6050.

Inter-Integrated Circuit, abbreviated as I2C is a serial bus short distance protocol developed by Philips Semiconductor about two decades ago to enhance communication between the core on the board and various other ICs involved around the core. This application note intends to describe the functionality of various serial buses with emphasis on I2C, and how I2C is

Chapter 1: Background Information and Literature Review

different from other serial buses. This application note also intends to explain the functionality and working of I2C, as well as some sample code that explains how I2C is implemented.

The general concept of serial bus communication is shown in Figure 1. The most popular serial bus communication protocols available today in the market are, SPI, UART, I2C, CAN, USB, IEE1394, and so on.

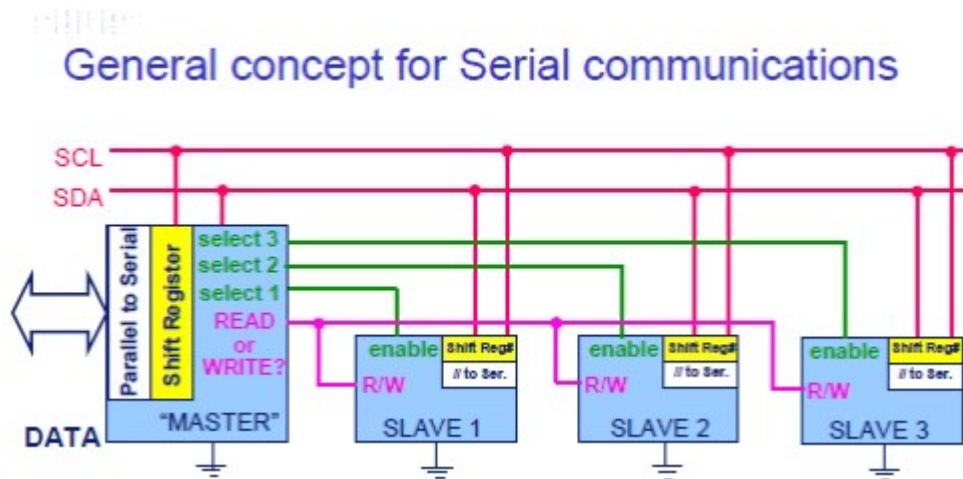


Figure 16 Serial Communication Overview

Philips originally developed I2C for communication between devices inside of a TV set. Examples of simple I2C-compatible devices found in embedded systems include EEPROMs, thermal sensors, and real-time clocks. I2C is also used as a control interface to signal processing devices that have separate, application-specific data interfaces. Philips, National Semiconductor, Xicor, Siemens, and other manufacturers offer hundreds of I2C-compatible devices. I2C buses can typically reach speeds up to 400 Kbps.

b Structure of I2C:

I2C is appropriate for interfacing to devices on a single board, and can be stretched across multiple boards inside a closed system. An example is a host CPU on a main embedded board using I2C to communicate with user interface devices located on a separate front panel board. I2C is a two-wire serial bus, as shown in Figure 1. There's no need for chip select or arbitration logic, making it cheap and simple to implement in hardware. The two I2C signals are serial data and serial clock. Together, these signals make it possible to support serial transmission of 8-bit bytes of data-7-bit device addresses plus control bits-over

Chapter 1: Background Information and Literature Review

the two-wire serial bus. In a bind, an I2C slave can hold off the master in the middle of a transaction using what's called clock stretching (the slave keeps SCL pulled low until it's ready to continue). Most The I2C protocol can also support multiple masters. There may be one or more slaves on the bus. Both masters and slaves can receive and transmit data bytes. Each I2C-compatible hardware slave device comes with a predefined device address, the lower bits of which may be configurable at the board level. The master transmits the device address of the intended slave at the beginning of every transaction. Each slave is responsible for monitoring the bus and responding only to its own address. This addressing scheme limits the number of identical slave devices that can exist on an I2C bus without contention, with the limit set by the number of user-configurable address bits.

Communication in I2C:

Figure 17 shows the communication in I2C.

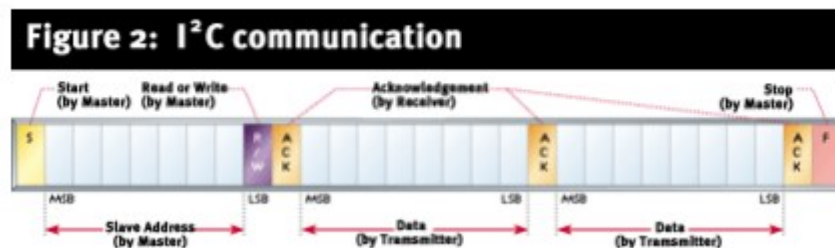


Figure 17 Communication in I2C

The I2C signalling protocol provides device addressing, a read/write flag, and a simple acknowledgement mechanism. Other elements of I2C protocol are general call (broadcast) and 10-bit extended addressing. Standard I2C devices operate up to 100Kbps, while fast-mode devices operate at up to 400Kbps. Most often, the I2C master is the CPU or microcontroller in the system. Some microcontrollers even feature hardware to implement the I2C protocol. You can also build an all-software implementation using a pair of general-purpose I/O pins. Since the I2C master controls transaction timing, the bus protocol doesn't impose any real-time constraints on the CPU beyond those of the application. For a fixed I2C, the high and low logics are defined at 3.0 V and 1.5 V. For dependant I2C, these are defined at $0.7 \cdot V_{dd}$ and $0.3 \cdot V_{dd}$ respectively. The pull-up resistor values required for I2C are typically at 1K for 3.0V of V_{dd} and

Chapter 1: Background Information and Literature Review

1.6K for 5V of Vdd. Typical operating temperatures are between -40 degrees and +85 degrees Centigrade.

c Addressing in I2C:

Figure 3 shows the SDA and SCL for I2C

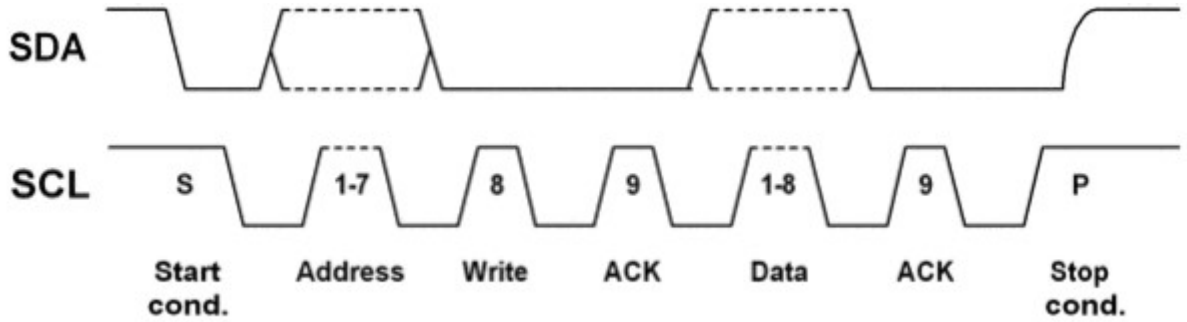


Figure 18 I2C addressing

The following table shows I2C addresses reserved for special purposes:

10 bit addresses, binary noted, MSB is left	Purpose
0000000 0	General Call
0000000 1	Start Byte
0000001 X	CBUS Addresses
0000010 X	Reserved for Different Bus Formats
0000011 X	Reserved for future purposes
00001XX X	High-Speed Master Code
11110XX X	10-bit Slave Addressing
11111XX X	Reserved for future purposes

Tableau 5 I2C Address

Chapter 1: Background Information and Literature Review

d Kalman Filter :

As microcontrollers work in discrete time, the discrete time Kalman Filter will be used to estimate the tilt angle. To implement the algorithm, the process has to be described by the linear stochastic difference equation [21] $x_k = Ax_{k-1} + Buk + w_k$ (8) where, x_k is the state vector containing the variables to be estimated. A is the state transition matrix that is applied to x_{k-1} . u_k is the control (input) vector and B (control input matrix) maps the inputs to the state vector. Finally, w_k is vector that contains the process noise for each of the variables in x_k . The noise is assumed to be normally distributed with the a mean value of zero and the covariance is given by Q (w) $\sim N(0, Q)$ (9) The measurements are modelled by $z_k = Hx_k + v_k$ (10) Where, z_k is contains the measured values of x_k . H transforms the state vector into measurements. v_k is the measurement noise also Gaussian distributed but with a variance of R (v) $\sim N(0, R)$ (11) w_k and v_k are white noises and independent of each other [30].

I.16 DC MOTORS

Almost every mechanical movement that we see around us is accomplished by an electric motor. Electric machines are a means of converting energy. Motors take electrical energy and produce mechanical energy. Electric motors are used to power hundreds of devices we use in everyday life. Motors come in various sizes. Huge motors that can take loads of 1000's of Horsepower is typically used in the industry. Some examples of large motor applications include elevators, electric trains, hoists, and heavy metal rolling mills. Examples of small motor applications include motors used in automobiles, robots, hand power tools and food blenders. Micro-machines are electric machines with parts the size of red blood cells, and find many applications in medicine.

Electric motors are broadly classified into two different categories: DC (Direct Current) and AC (Alternating Current). Within these categories are numerous types, each offering unique abilities that suit them well for specific applications. In most cases, regardless of type, electric motors consist of a stator (stationary field) and a rotor (the rotating field or armature) and operate through the interaction of magnetic flux and electric current to produce rotational speed and

Chapter 1: Background Information and Literature Review

torque. DC motors are distinguished by their ability to operate from direct current.

There are different kinds of D.C. motors, but they all work on the same principles. In this chapter, we will study their basic principle of operation and their characteristics. It's important to understand motor characteristics so we can choose the right one for our application requirement. The learning objectives for this chapter are listed below:

- Understand the basic principles of operation of a DC motor.
- Understand the operation and basic characteristics of simple DC motors.
- Compute electrical and mechanical quantities using the equivalent circuit.
- Use motor nameplate data.
- Study some applications of DC motors.

I.17 Electromechanical Energy Conversion

An electromechanical energy conversion device is essentially a medium of transfer between an input side and an output side. Three electrical machines (DC, induction and synchronous) are used extensively for electromechanical energy conversion. Electromechanical energy conversion occurs when there is a change in magnetic flux linking a coil, associated with mechanical motion.

a Electric Motor

The input is electrical energy (from the supply source), and the output is mechanical energy (to the load). Electrical Electromechanical Mechanical energy conversion device energy Source Motor Load

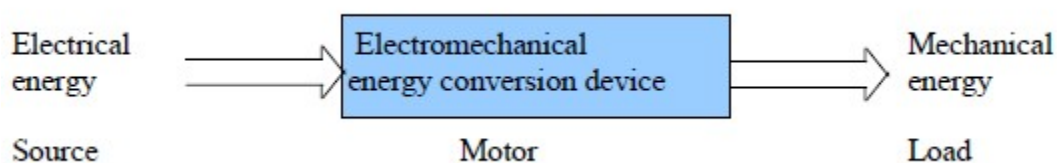


Figure 19

Chapter 1: Background Information and Literature Review

b Electric Generator

The Input is mechanical energy (from the prime mover), and the output is electrical energy.

Mechanical energy Source → Electromechanical energy conversion device Generator → Electrical energy Load

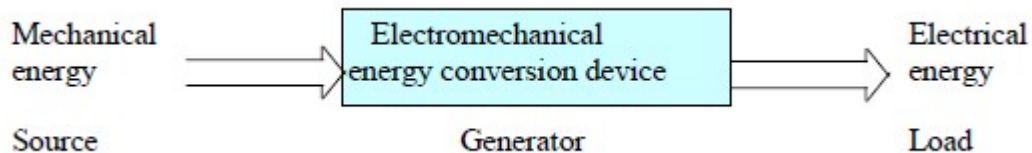


Figure 20

I.18 Construction

DC motors consist of one set of coils, called armature winding, inside another set of coils or a set of permanent magnets, called the stator. Applying a voltage to the coils produces a torque in the armature, resulting in motion.

a Stator

- The *stator* is the stationary outside part of a motor.
- The stator of a permanent magnet dc motor is composed of two or more permanent magnet pole pieces.
- The magnetic field can alternatively be created by an *electromagnet*. In this case, a DC coil (field winding) is wound around a magnetic material that forms part of the stator.

b Rotor

- The *rotor* is the inner part which rotates.
- The rotor is composed of windings (called armature windings) which are connected to the external circuit through a mechanical commutator.
- Both stator and rotor are made of ferromagnetic materials. The two are separated by air-gap.

Chapter 1: Background Information and Literature Review

I.19 Winding

A winding is made up of series or parallel connection of coils.

- Armature winding - The winding through which the voltage is applied or induced.
- Field winding - The winding through which a current is passed to produce flux (for the electromagnet)

I.20 DC Motor Basic Principles

a Energy Conversion

If electrical energy is supplied to a conductor lying perpendicular to a magnetic field, the interaction of current flowing in the conductor and the magnetic field will produce mechanical force (and therefore, mechanical energy).

b Value of Mechanical Force

There are two conditions which are necessary to produce a force on the conductor. The conductor must be carrying current, and must be within a magnetic field. When these two conditions exist, a force will be applied to the conductor, which will attempt to move the conductor in a direction perpendicular to the magnetic field. This is the basic theory by which all DC motors operate. The force exerted upon the conductor can be expressed as follows.

$$F = B i l \text{ Newton (1)}$$

where B is the density of the magnetic field, l is the length of conductor, and i the value of current flowing in the conductor. The direction of motion can be found using Fleming's

Left-Hand Rule.

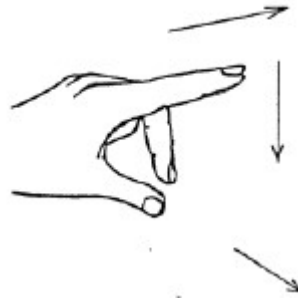


Figure 21 Fleming's Left-Hand Rule

The first finger points in the direction of the magnetic field (first - field), which goes from the North pole to the South pole. The second finger points in the direction of the current in the wire (second - current). The thumb then points in the direction the wire is thrust or pushed while in the magnetic field (thumb - torque or thrust)

c Principle of operation

Consider a coil in a magnetic field of flux density **B** (figure 4). When the two ends of the coil are connected across a DC voltage source, current **I** flows through it. A force is exerted on the coil as a result of the interaction of magnetic field and electric current. The force on the two sides of the coil is such that the coil starts to move in the direction of force.

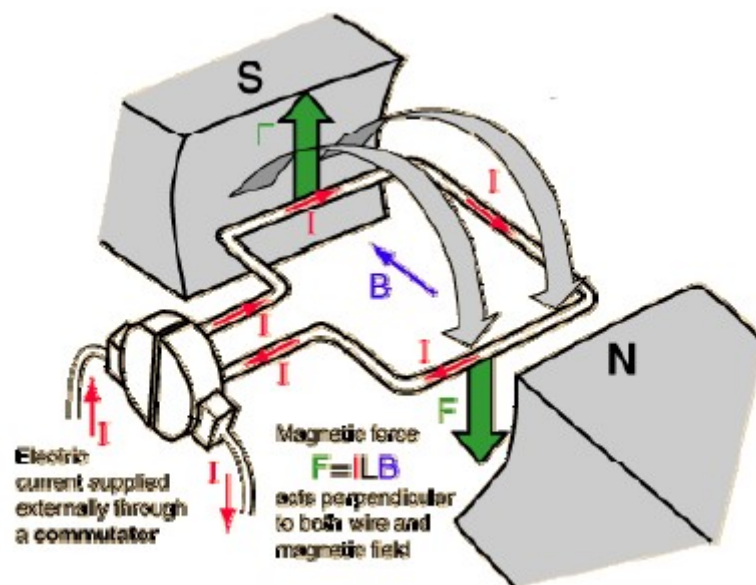


Figure 22 : Torque production in a DC motor

Chapter 1: Background Information and Literature Review

In an actual DC motor, several such coils are wound on the rotor, all of which experience force,

resulting in rotation. The greater the current in the wire, or the greater the magnetic field, the faster the wire moves because of the greater force created. At the same time this torque is being produced, the conductors are moving in a magnetic field. At different positions, the flux linked with it changes, which causes an *emf* to be induced ($e = d\Phi/dt$) as shown in

figure 23. This voltage is in opposition to the voltage that causes current flow through the conductor and is referred to as a *counter-voltage* or *back emf*.

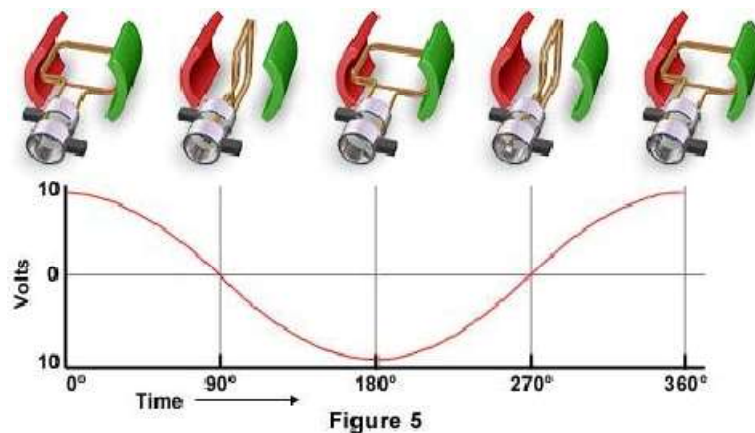


Figure 23

CHAPTER 2

METHODOLOGY

1.1. Introduction

The methodology of this project is divided into two parts namely the mechanical design and the software algorithm. This chapter will describe the method for this subject in order to achieve the desired objective. The project will be developed based on a flowchart that determines all the necessary activities that has to be accomplished. Figure 24 shows the flow chart of the project, which begins with information collection on topics related to the two-wheeled balancing robot through literature review. After reviewing all the resources, next step is to do the modelling for the inverted pendulum as it provides the fundamental concept for this project. Matlab Simulink will be used as a simulation tools to see the performance of the controller. Next, is to design and fabricate the hardware. The process includes building base and body of the robot. This is follow by software algorithm implementation and hardware integration. Finally, the robot will be tested and fine-tuned for performance improvement.

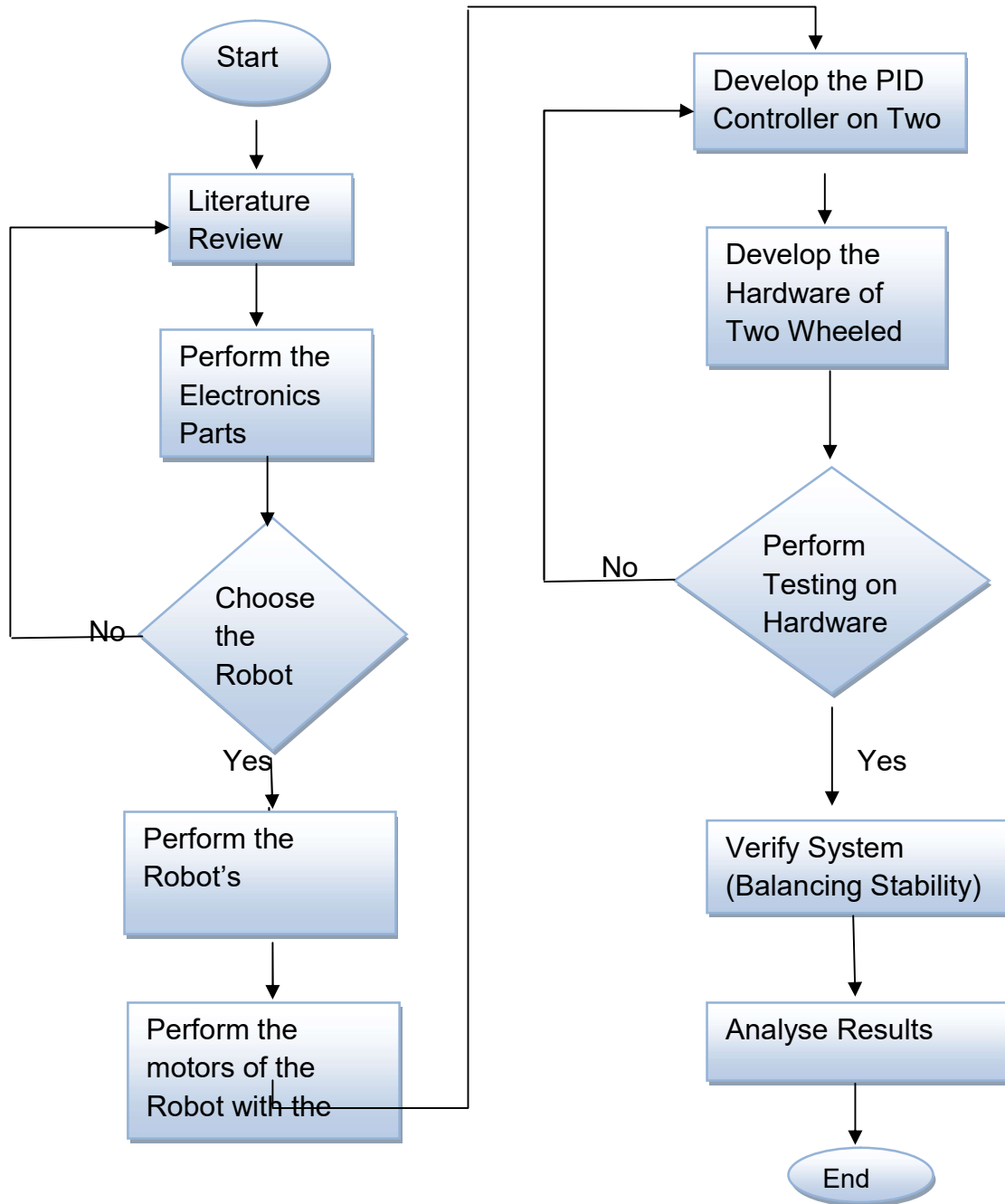


Figure 24 Flow chart of the project

1.2. Hardware development

The design of the hardware system is crucial in bringing the mechanism and software to work together. The main components in the circuit of the balancing robot are the inertial measurement unit (IMU), the Arduino controller and the DC servo motor. Figure 25 shows the overall block diagram of the electronic system

for the balancing robot. The IMU is used to measure the acceleration and the angular rate of the robot and the output is processed into digital form. The raw inputs from the IMU are further processed to obtain the tilt angle of the robot. This tilt angle is then fed into the PID controller algorithm to generate the appropriate speed to the DC motor in order to balance the robot.

IMU

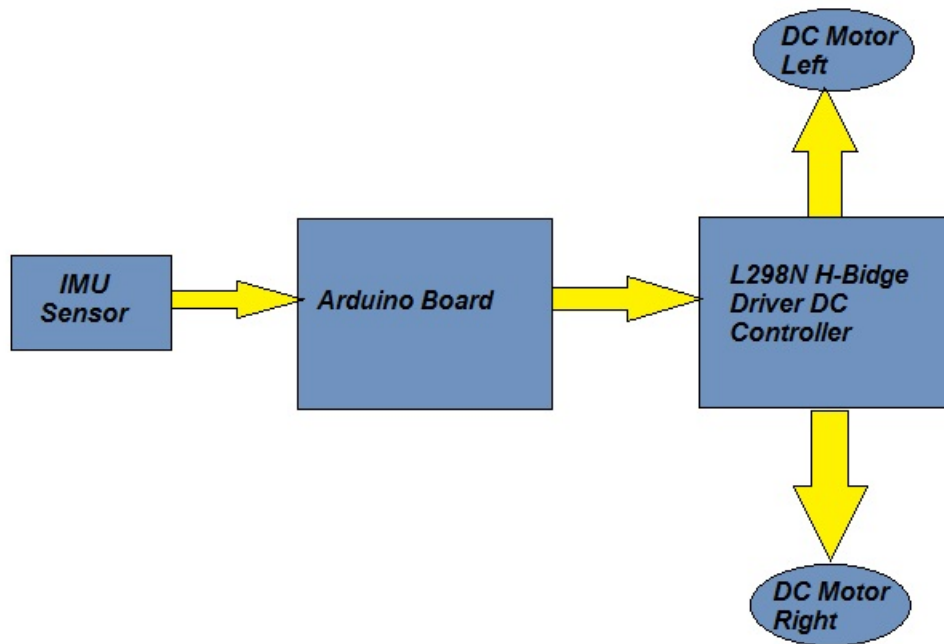


Figure 25 Block diagram of the hardware

I. Arduino board

The main controller chosen for the balancing robot is the Arduino Uno as shown in Figure 3.4. It can be considered as the brain of the balancing robot and is connected to the IMU to process the tilt angle information. After processing, it will communicate with the motor driver in order to adjust the speed and direction of the motor.



Figure 26 Arduino Uno board

II. IMU sensor

In order to obtain the tilt angle of the balancing robot; the Six Degree of Freedom Inertial Measurement Unit (IMU) is used as in Figure 27. The MPU 6050 is a 6 DOF which means that it gives six values as output. The value consists three values from the accelerometer and three from the gyroscope. This chip uses I2C (Inter Integrated Circuit) protocol for communication. The module has on board Digital Motion Processor (DMP) capable of processing complex 9-axis Motion-Fusion algorithms. The SDA and SCL pins are used to establish a connection with the Arduino pins A4 and A5 to receive the accelerometer and gyroscope data. The interrupt pint (INT) is to instruct the Arduino when to read the data from the module and this pin instruct the Arduino only when the values change.

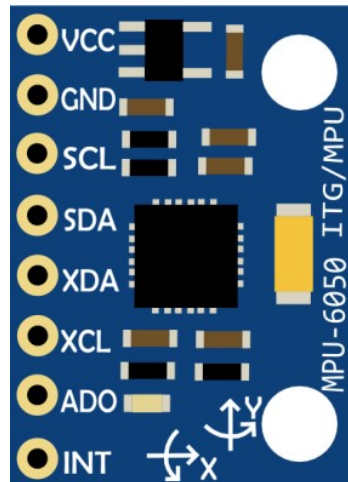


Figure 27 IMU board

IMU	Advantage	Disadvantage
Accelerometer	No bias	Affected by object's acceleration
Gyroscope	Unaffected by object's acceleration	Accumulated bias

Tableau 6 Comparison between Accelerometer and Gyroscope in IMU

The biggest advantage of the DMP is that it eliminates the need to perform complex calculations on the Arduino side. The DMP combines the raw sensor data and performs some calculations onboard to minimize the errors in each sensor. Accelerometers and gyros have different inherent limitations when used on their own as indicate in Table 6

III. I2C interface

As depicted in Figure 28, Inter-Integrated Circuit (I²C pronounced I- Squared- C) is a 2 wire serial bus typically used to communicate with sensors and other small components. The two lines of the I2C bus are SDA (Data) and SCL (clock) which can be run in parallel to communicate with several devices at once. I2C allows up to 112 "slave" (such as a sensor) devices to be controlled by a single "master" (such as Arduino, in our case).

Each slave device on the bus must have its own unique address so the master can communicate directly with the intended device. Both master and slave can transfer data over the I2C bus but that transfer is always controlled by the master. These addresses are typically hard-coded into the slave device, but often allow it to be changed by simply pulling one of the pins of the sensor high or low. This allows more than one of the same device to be on the same bus without conflicting addresses.

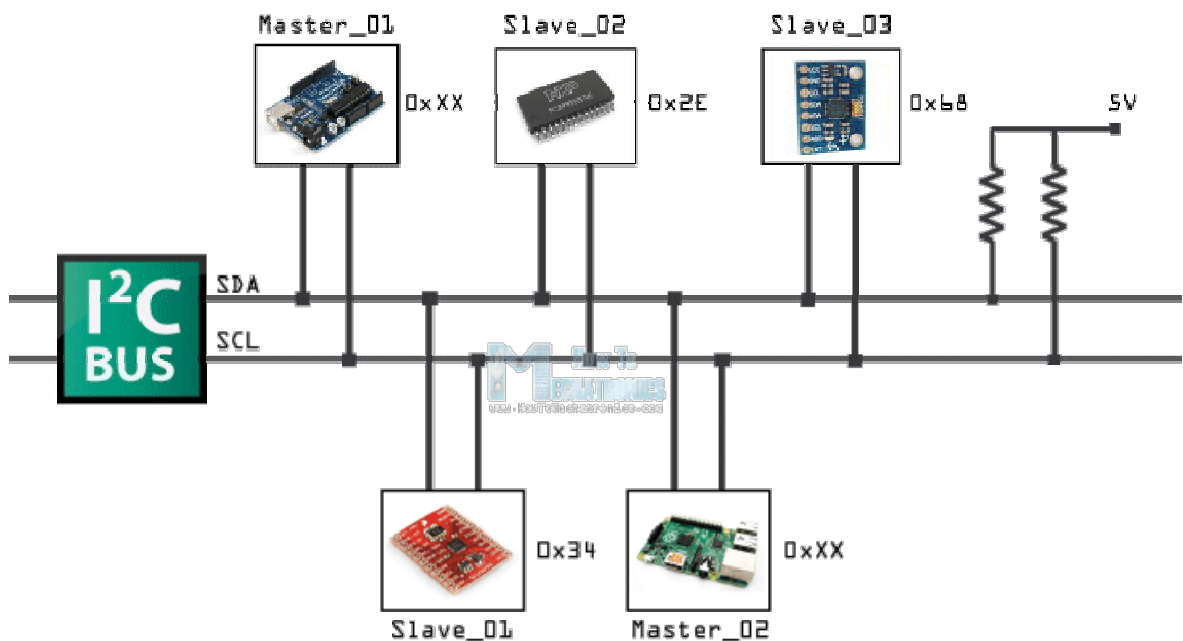


Figure 28 I2C protocol

IV. DC Gear-Motor

Figure 29 shows the DC gear motor used as the actuator of the two wheel balancing robot. The motor is used to generate torque so that the robot could

balance itself and stay in the upright position. For this motor, the 120mA Dc motor driver is used as the motor controller.

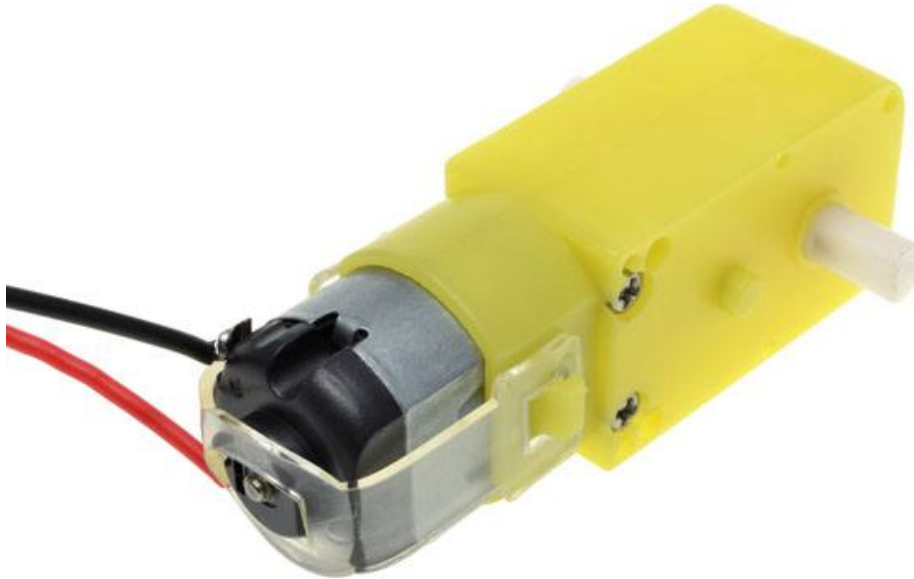


Figure 29 DC geared motor

For the DC motor used, below are the features of the DC motor:

- i. Rated speed is 130rpm. The robot requires an average rpm so that it could counter the balancing error in a suitable speed. Low speed might not be able to balance the robot properly. Therefore, a higher rpm is chosen.
- ii. Rated torque is 127.4 mN.m. The torque of the motors must be carefully chosen because a low torque might not be capable to balance the robot. The torque does not necessarily be too high. The torque required is based on the formula, Torque = Force x Distance

Voltage	DC 3V-6V
Current	100 MA-120MA
Reduction rate	48: 1

RPM (With wgeel)	100-240
Tire Diameter	65mm
Car Speed(M/minute)	20-48
Motor Weight (g)	29/each
Motor Size	70mm X 22mm X 18mm
Noise	<65dB

Tableau 7 Table DC gear-motor Specification

Center hole	5.3MM x 3.66MM
Wheel size	65 x 26mm

Tableau 8 Tires Parameter



Figure 30 DC Gear-Motor and Wheel Parameters

V. L298 Dual H-Bridge DC Motor Controller

This module is based on the very popular **L298 Dual H-Bridge Motor Driver** Integrated Circuit. The circuit will allow to us easily and independently control two motors of up to 2A each in both directions.

It is ideal for robotic applications and well suited for connection to a microcontroller requiring just a couple of control lines per motor

Here is the schematic diagram for the circuit

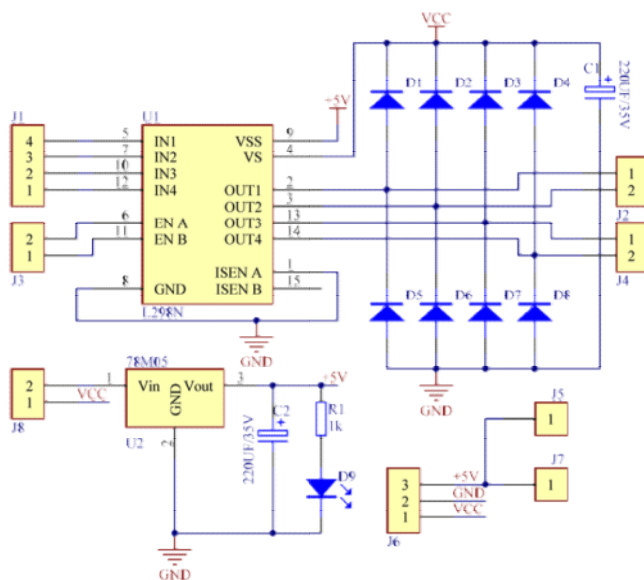


Figure 31 Schematic diagram for motor drive module



Figure 32 L298 Dual H-Bridge Board DC Motor Controller

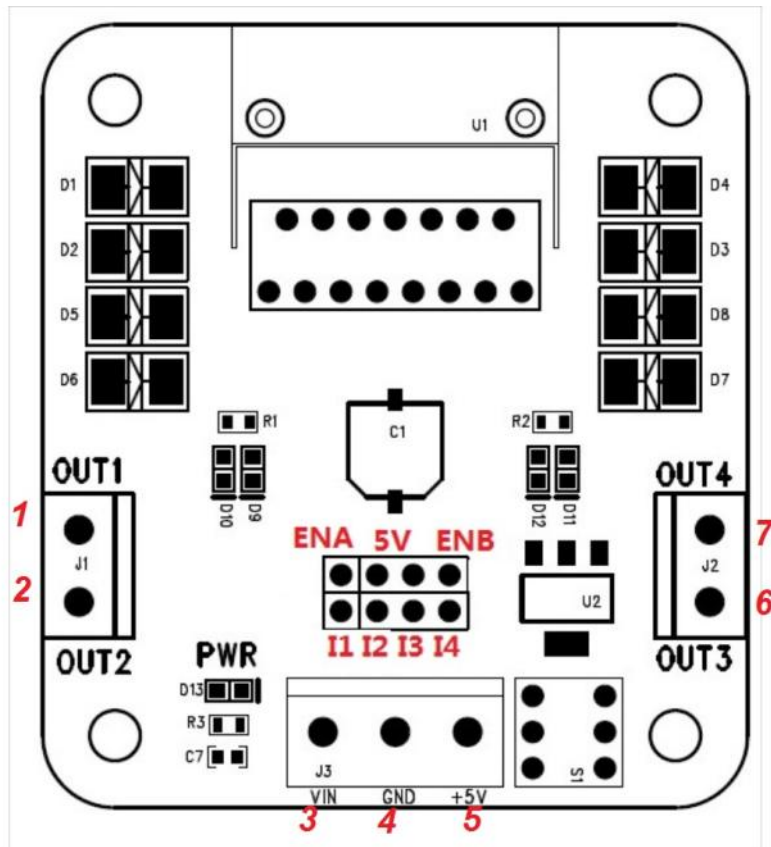


Figure 33 Input Output OF The L298 Dual H-Bridge Board

Consider the image – match the numbers against the list below the image:

- 1 ----- DC motor 1 “+” OUT1
- 2 ----- DC motor 1 “-” OUT2
- 3 ----- VIN Connect motor supply voltage here, maximum of 35V DC.
- 4 ----- GND
- 5 ----- 5V output if 12V jumper in place, ideal for powering your Arduino (etc)
- 6 ----- DC motor 2 “+” OUT3
- 7 ----- DC motor 2 “-” OUT4
- IN1 ----- To control Motor 1 direction
- IN2 ----- To control Motor 1 direction
- IN3 ----- To control Motor 2 direction

IN4 ----- To control Motor 2 direction

ENA ----- DC motor 1 enable jumper. Connect to PWM output for DC motor speed control.

ENB -----DC motor 2 enable jumper. Connect to PWM output for DC motor speed control

To control one or two DC motors is quite easy with the [L298N H-bridge module](#). First we connected each motor to the A and B connections on the L298N module.

We should verify the polarity of the motors used for the robot and ensure that is the same on both inputs. Otherwise we may need to swap them over when you set both motors to forward and one goes backwards!

Next, we connect our power supply – the positive to pin 4 on the module and negative/GND to pin 5.

This can be fed to your Arduino's 5V pin to power it from the motors' power supply. And we must connect Arduino GND to pin 5 on the module as well to complete the circuit.

Now we will need six digital output pins on Arduino, two of which need to be PWM (pulse-width modulation) pins. PWM pins are denoted by the tilde (“~”) next to the pin number, for example in the image of the Arduino Uno's digital pins.

Finally, we connect the Arduino digital output pins to the driver module. In our project we have two DC motors, so digital pins D4, D5, D7 and D6 will be connected to pins IN1, IN2, IN3 and IN4 respectively. Then connect D10 to module pin 7 (remove the jumper first) and D11 to module pin 12 (again, remove the jumper).

The motor direction is controlled by sending a HIGH or LOW signal to the drive for each motor (or channel). For example for motor one, a HIGH to IN1 and a LOW to IN2 will cause it to turn in one direction, and a LOW and HIGH will cause it to turn in the other direction.

However the motors will not turn until a HIGH is set to the enable pin (7 for motor one, 12 for motor two). And they can be turned off with a LOW to the same pin(s). However if we need to control the speed of the motors, the PWM signal from the digital pin connected to the enable pin can take care of it.

This is what we've done with the DC motor demonstration EXAMPLE 1.

So what's happening in that sketch? In the function `demoOne()` we turn the motors on and run them at a PWM value of 200. This is not a speed value, instead power is applied for 200/255 of an amount of time at once.

Then after a moment the motors operate in the reverse direction (see how we changed the HIGHS and LOWs in the `digitalWrite()` functions?). To get an idea of the range of speed possible of your hardware, we run through the entire PWM range in the function `demoTwo()` which turns the motors on and then runs through PWM values zero to 255 and back to zero with the two for loops.

Finally this is demonstrated in the video on this page – using our well-worn tank chassis with two DC motors.

I.3. Software development

This section describes the balancing of the two wheeled robot and the designing of the optimum PID controller. The software development is the hardest the most time consuming part of the project. It involves the tuning of the PID controller to get the optimum value of K_p , K_i and K_d . Then, the controller will be integrated into the final project code.

I. The characteristics of PID controller (Tuning Parameters)

A proportional controller (K_p) will have the effect of reducing the rise time and will reduce but never eliminate the steady-state error. An integral control (K_i) will have the effect of eliminating the steady-state error for a constant or step input,

but it may make the transient response slower. A derivative control (K_d) will have the effect of increasing the stability of the system, reducing the overshoot and improving the transient response. The effects of each of controller parameters, K_p , K_d and K_i on a closed loop system are summarized in the

Response	Rise Time	Overshoot	Settling Time	Steady State Error
K_p	Decrease	Increase	Small Change	Decrease
K_i	Decrease	Increase	Increase	Eliminate
K_d	Small Change	Decrease	Decrease	No change

Tableau 9 Effect of PID Tuning

- RiseTime — Time it takes for the response to rise from 10% to 90% of the steady-state response.
- SettlingTime — Time it takes for the error $|y(t) - y_{final}|$ between the response $y(t)$ and the steady-state response y_{final} to fall to within 2% of y_{final} .
- SettlingMin — Minimum value of $y(t)$ once the response has risen.
- SettlingMax — Maximum value of $y(t)$ once the response has risen.
- Overshoot — Percentage overshoot, relative to y_{final} .
- Undershoot — Percentage undershoot.
- Peak — Peak absolute value of $y(t)$
- PeakTime — Time at which the peak value occurs.

The following figure illustrates some of these quantities on a typical second-order response.

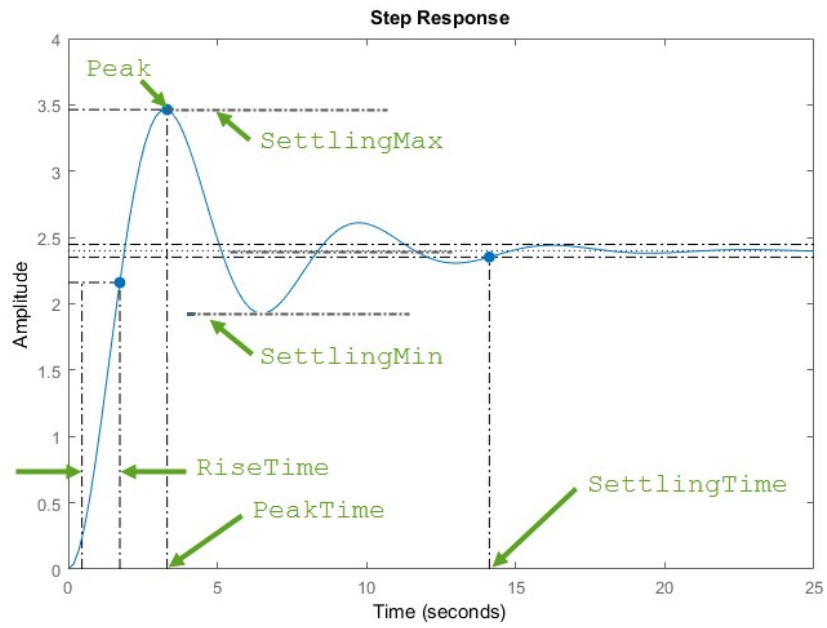


Figure 34 Step Response

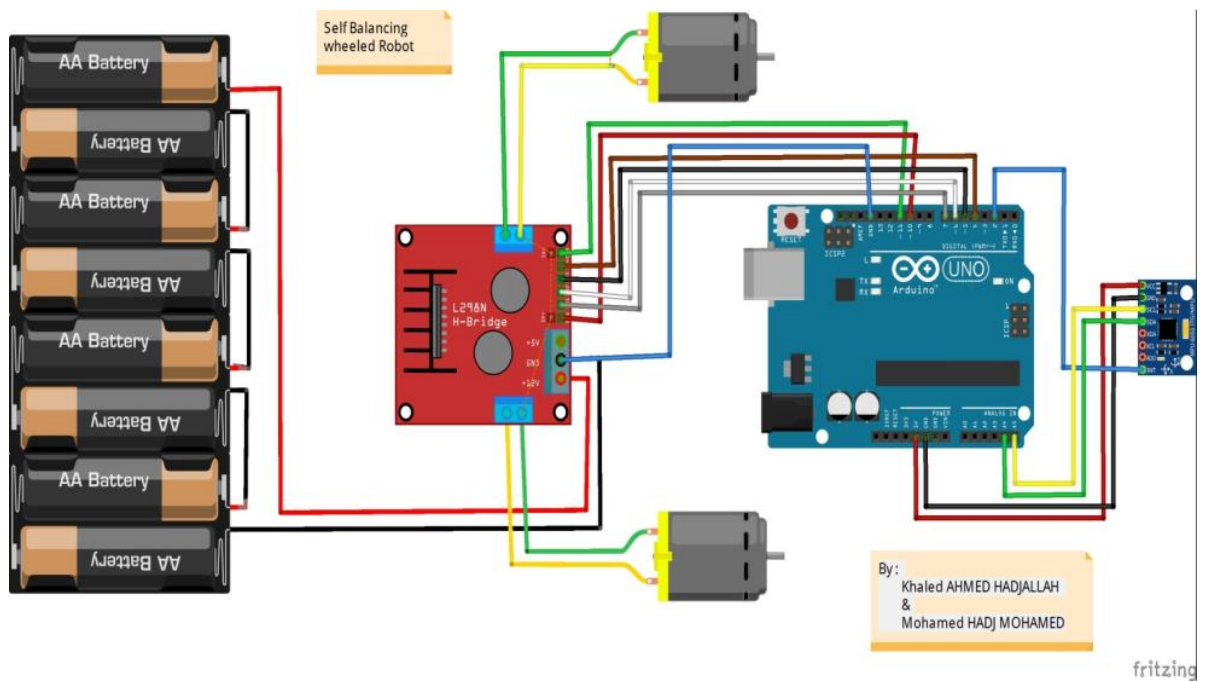


Figure 35 Synoptic diagram

CHAPTER 3

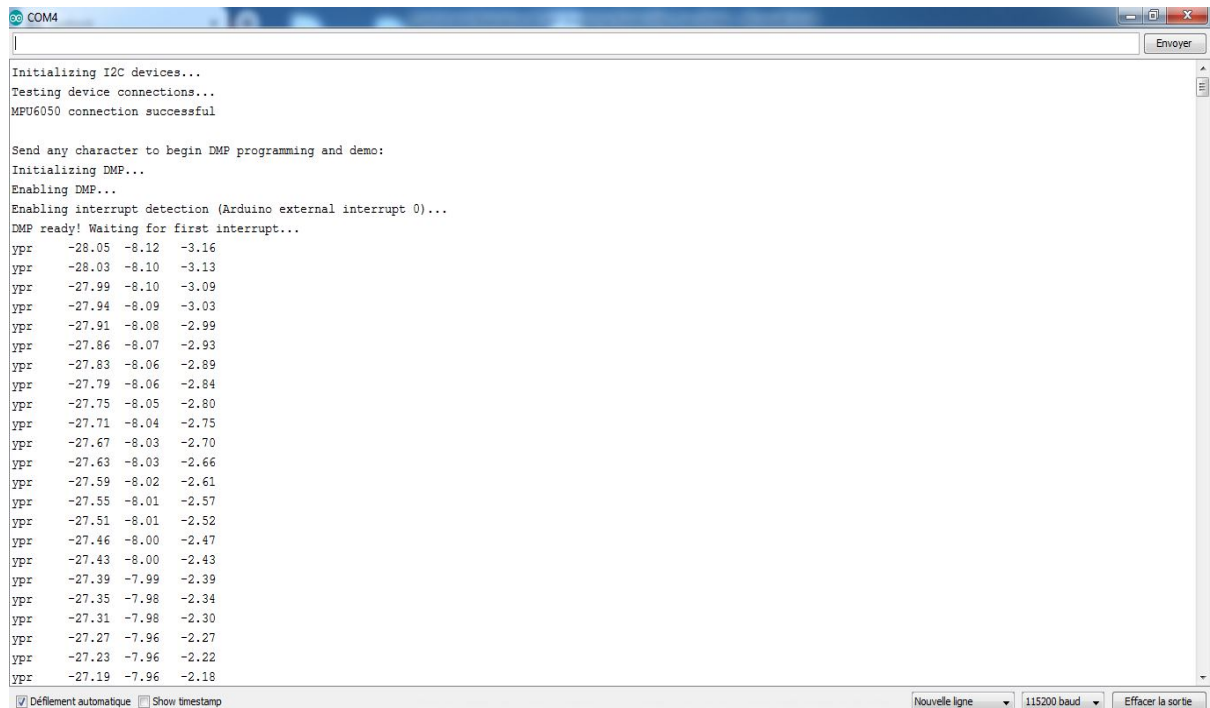
DISCUSSION AND RESULTS

II. Discussion Results

This section describes in detail the results of the project. It provides a qualitative assessment as well as quantitative data to support that assessment.

II.1. Initialising and enabling the DMP

II.1.1. 1st step: reading data from MPU6050



```

COM4
Initializing I2C devices...
Testing device connections...
MPU6050 connection successful

Send any character to begin DMP programming and demo:
Initializing DMP...
Enabling DMP...
Enabling interrupt detection (Arduino external interrupt 0)...
DMP ready! Waiting for first interrupt...
ypr  -28.05  -8.12  -3.16
ypr  -28.03  -8.10  -3.13
ypr  -27.99  -8.10  -3.09
ypr  -27.94  -8.09  -3.03
ypr  -27.91  -8.08  -2.99
ypr  -27.86  -8.07  -2.93
ypr  -27.83  -8.06  -2.89
ypr  -27.79  -8.06  -2.84
ypr  -27.75  -8.05  -2.80
ypr  -27.71  -8.04  -2.75
ypr  -27.67  -8.03  -2.70
ypr  -27.63  -8.03  -2.66
ypr  -27.59  -8.02  -2.61
ypr  -27.55  -8.01  -2.57
ypr  -27.51  -8.01  -2.52
ypr  -27.46  -8.00  -2.47
ypr  -27.43  -8.00  -2.43
ypr  -27.39  -7.99  -2.39
ypr  -27.35  -7.98  -2.34
ypr  -27.31  -7.98  -2.30
ypr  -27.27  -7.96  -2.27
ypr  -27.23  -7.96  -2.22
ypr  -27.19  -7.96  -2.18

 Défilement automatique  Show timestamp
Nouvelle ligne 115200 baud Effacer la sortie

```

Figure 36 DMP Data from MPU6050

II.1.2. 2nd step: Finding the minimum speed:

We carried out several experiments on the robot to figure out the minimum speed of the motor so the robot can move, we did that by emitting a burst of PWM signal on the power board until the robot moves, in our case the minimum speed was 20t/minute.

II.2. Control Loop Performance

The control loop frequency is the largest number of times the control loop can run in one second. The higher this value is, the better the accuracy of the sensor integration and the better the balancing.

Our initial target for the control loop frequency was 100Hz. This target allows 100ms time to receive sensor data, process it, and output the motor duty cycle. Upon implementing the code, we discovered that the algorithm could easily run at over 500Hz. The maximum that we recorded was 530Hz. The control loop frequency varies by ± 20 Hz depending on the interrupt mechanisms used by the software PWM.

This was much better than expected and it allowed us to add even more complexity to the loop. We added in a serial communications interface that allowed us to tune the PID values in real-time via a serial cable. Adding this interface brought the control loop frequency to around 490Hz – which was still much higher than the required target.

II.3. Balancing Performance

We set about tuning the PID controller to achieve the best balance. With only the proportional gain, we managed to get the robot to oscillate about the vertical with a jitter of around $3\text{cm} \pm 1\text{cm}$ measured at the top of the robot. The robot could balance completely on its own for about 15 seconds. After this time, the steady state error and oscillation amplitude grew exponentially which would make the robot fall on one side.

Reducing the proportional gain and adding in the integral gain allowed the oscillation amplitude to be dampened. The robot now would now balance with minimal jitter of less than 1cm. The robot also responded to external disturbances (pushing with a finger) very well. The robot regained balance very quickly (within 1 second). However, an excessive disturbance caused it to oscillate wildly and then fall over. Adding in the derivative gain reduced the overshoot. At this state in the tuning, one effect that was very noticeable was that the robot could follow a finger with minimal force.

Further tuning would allow the robot to balance indefinitely. However, there was a lot of play in the gear motors we were using. Hence further tuning did not affect the balance performance. Moreover, the sponsor had plans to build a proper, large scale version of the robot, with precisely controllable stepper motors. This robot will need a complete retune of the PID parameters anyway; hence we decided not to spend more time on tuning our prototype version.

II.3.1. 3rd step: PID Controller:

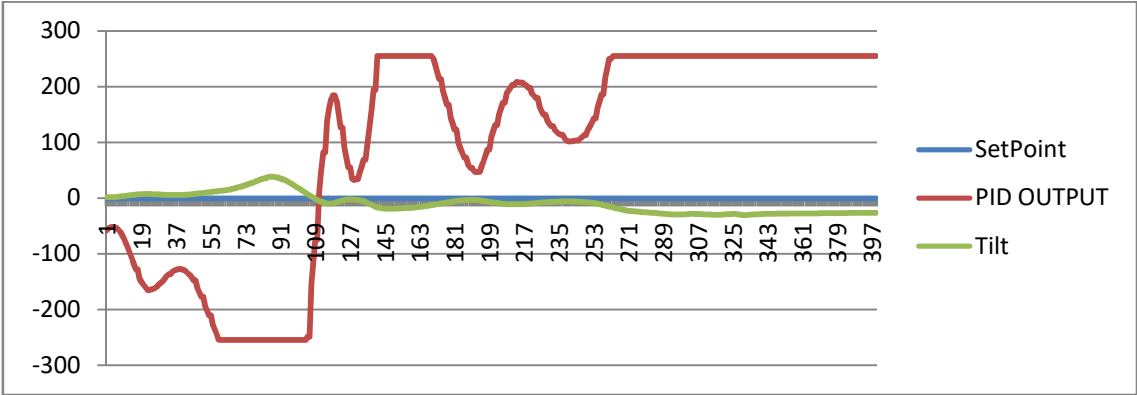


Figure 37 PID = 20_0_0

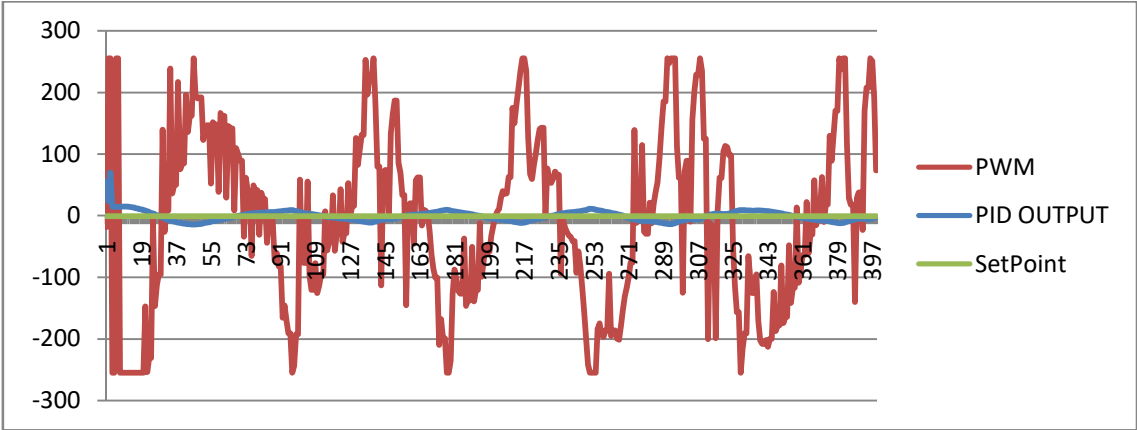


Figure 38 PID = 15_190_1.6

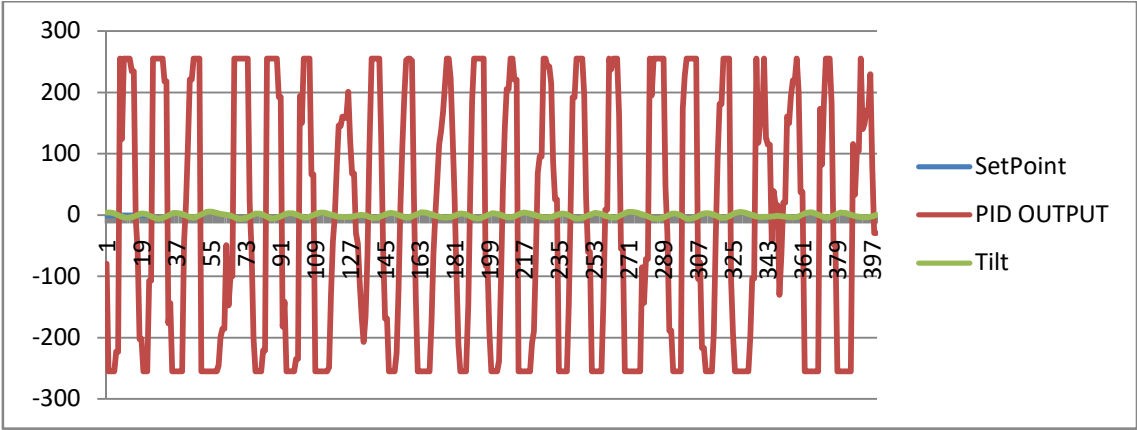


Figure 39 PID = 70_240_1.9

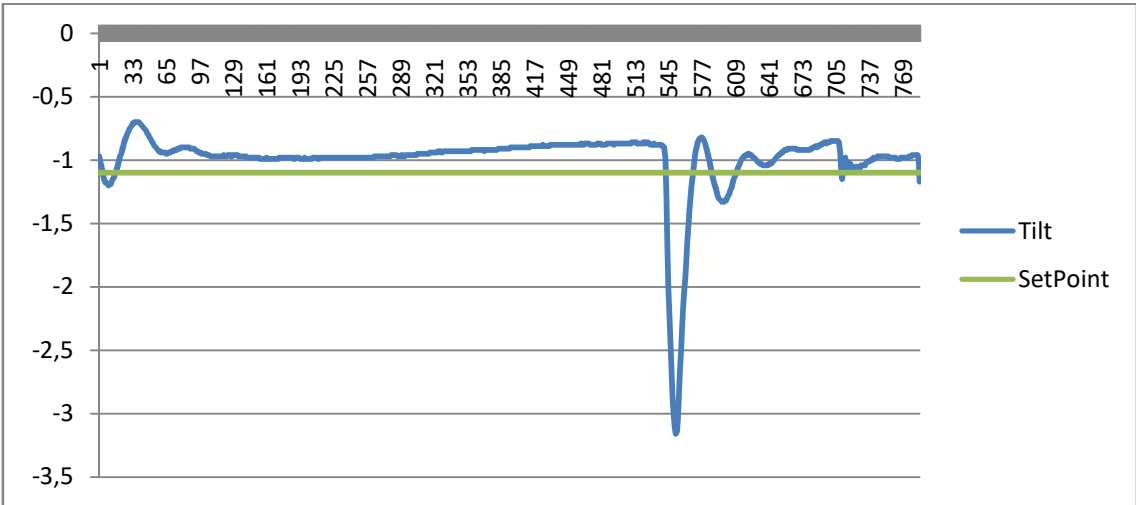


Figure 40 PID = 15_190_1.6

General Conclusion

III. General Conclusion

III.1. Project review

This project was successful in achieving its aims to balance a two-wheeled autonomous robot based on the inverted pendulum model. PID control strategies have been implemented to address the problem of balance control for the system. The PID controller is implemented in our robot for real-time experimentation showed promising results in balancing the robot. During testing, the robot is able to maintain its vertical position by slightly adjusting its wheels.

In one of the previous articles we explained how to obtain the angular position by use of a gyroscope. We saw that it was very easy to obtain an accurate measurement that was not susceptible to external forces. The less good news was that, because of the integration over time, the measurement has the tendency to drift, not returning to zero when the system went back to its original position. The gyroscope data is reliable only on the short term, as it starts to drift on the long term.

The Zeigler-Nichols tuning rules used for the PID controller are simple and intuitive as they require little process knowledge and can be applied with modest effort. However, for an accurate and robust operation, the trajectory control dynamics would be modelled to enable a suitable controller be designed.

More research is needed to investigate the effects of linearising the dynamics of the system mode to improve the stability and robustness of the robot. An attempt to use other controller or some algorithm to achieve the optimal stability of the robot is highly recommended for future research. That way, oscillatory movements of the robot while balancing can be eliminated, thus accurate trajectory control and waypoint navigation can be implemented.

III.2. *Recommendations for future work*

This thesis provides the base for future research on self balancing robot applications and control systems development. More research should be conducted to exploit the self balancing technology and its application for other projects that require sensor fusion technology.

The linear control system developed in this thesis proved to be able to balance the robot under minimal disturbance. But the robustness of the system is not fully tested and is in question. More experiment needs to be performed to evaluate the robustness of the system and fine tuning of the control algorithm is required for better performance.

Future research on implementing non-linear controllers is strongly recommended for the balancing robot system as it will improve the robustness of the system.

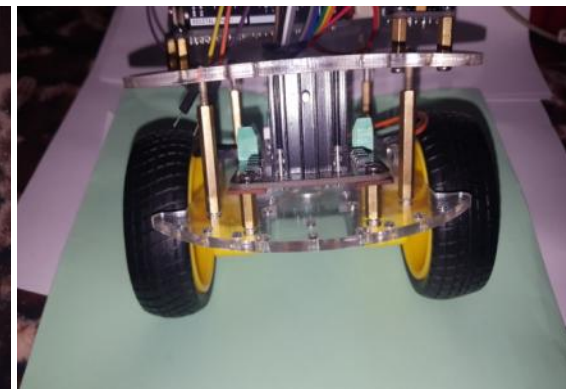
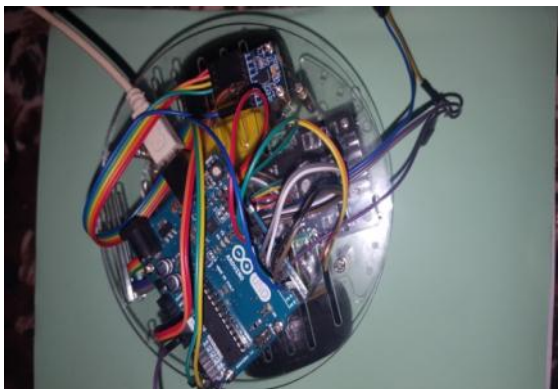
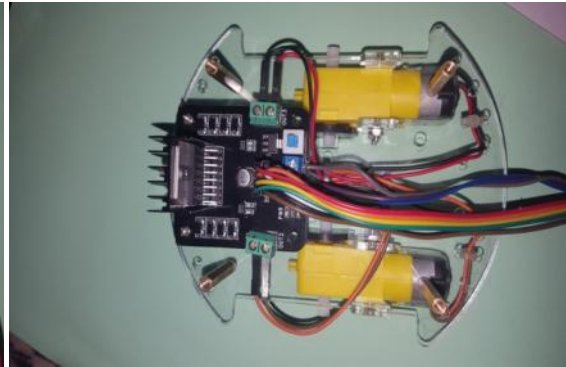
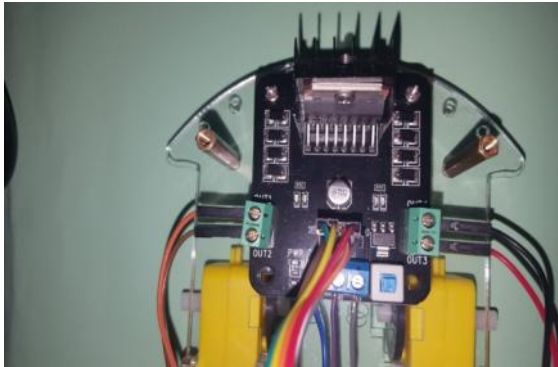
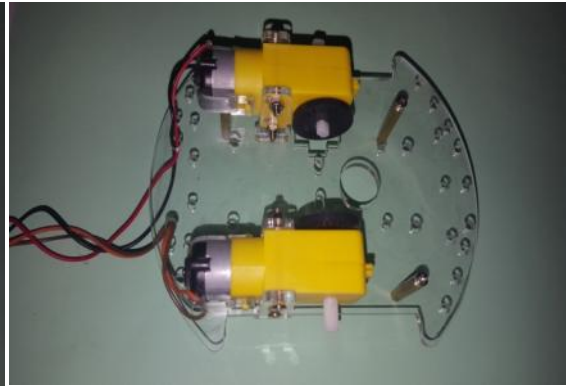
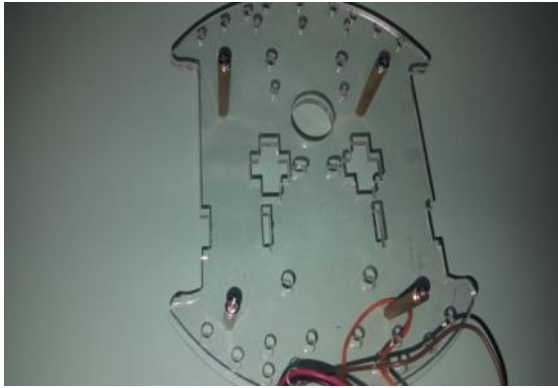
The trajectory control of the robot was delayed mainly because the robot was unable to stay at a fixed spot. When the problem is solved, the present trajectory control system can be expanded to enable waypoints to be set for the robot to follow instead of the limited movement restricted with the remote controller. Combining the two sensors yields the filtered estimate which has none of the problems of the two individual sensors.

References

- [1] O. Boubaker, "The inverted pendulum: A fundamental benchmark in control theory and robotics," IEEE, 2012.
- [2] D. Goldman, N. Gravish, S. Sharpe and H. Li, "Nonlinear Dynamics of Human Locomotion," Georgia Institute of Technology, Shanghai Jiao Tong University, 2012.
- [3] C. J, Interviewee, 5 Lines of Code to Land on the Moon. [Interview]. 2016.
- [4] MathWorks. 2012. *Control Tutorials for MATLAB and SIMULINK, Inverted Pendulum*. Available at: <http://ctms.engin.umich.edu/CTMS/index.php.exampleInverted> Pendulum section - System Modeling. [Accessed 21 April 2015]
- [5] Brooks, R. A. (1986), a robust layered control system for a mobile robot. *Robotics and Automation, IEEE Journal of*, 2(1).
- [6] Barbosa, Ramiro S.; Machado, J. A. Tenreiro; Ferreira, Isabel M, Tuning of PID controllers based on Bode's ideal transfer function. *Nonlinear dynamics*
- [7] D. Maiti, A. Acharya M. Chakraborty, A. Konar, R. Janarthanan, "Tuning PID and Fractional PID Controllers using the Integral Time Absolute Error Criterion" 4th International Conference on Information and Automation for Sustainability, 2008.
- [8] Z. Yongpeng , Sh. Leang-San , M. A.Cajetan , and A. Warsame, " Digital PID controller design for delayed multivariable systems", *Asian Journal of Control*, Vol. 6, No. 4, Dec 2004.
- [9] Pierre, D.A. and J.W. Pierre, "Digital Controller Design-Alternative Emulation Approaches," *ISA Trans.*Vol. 34, No. 3, (1995).
- [10] Esme, B. (2016). Bilgin's Blog | Kalman Filter For Dummies
- [11] Gornicki, K. (2015). Autonomous Self Stabilising Robot, Undergraduate. The University of Manchester
- [12] Baines, G. (2015) 'Embedded Systems Project: Motor Characterisation and Gearbox Ratio Selection'
- [13] Pololu.com. (2016). Pololu - 47:1Metal Gearmotor 25Dx52L mm MP 12V with 48 CPR Encoder

- [14] Energizer.com. (2016). Product Datasheet - L91 Ultimate Lithium
- [15] Learn.sparkfun.com. (2016). Battery Technologies
- [16] Hobbyking. (2016). Turnigy 2200mAh 3S 20C Lipo Pack
- [17] Sparkfun. (2016). L298 H Bridge Datasheet
- [18] Rugged Circuits. (2016). The Motor Driver Myth
- [19] Anon, (2016). L6203 Datasheet
- [20] Invensense.com. (2016). MPU-9250 | InvenSense
- [21] Sainsmart.com. (2016). SainSmart 2-Wheel Arduino Self-Balancing Robot Kit 3D Printing, Arduino, Robotics | Sainsmart

Building the Robots



EXAMPLE 1 (Control DC-Motor PWM Code .ino)

```
// connect motor controller pins to Arduino digital pins
// motor one
int enA = 10;
int in1 = 4;
int in2 = 5;
// motor two
int enB = 11;
int in3 = 7;
int in4 = 6;
void setup()
{
  // set all the motor control pins to outputs
  pinMode(enA, OUTPUT);
  pinMode(enB, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
}
void demoOne()
{
  // this function will run the motors in both directions at a fixed speed
  // turn on motor A
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  // set speed to 200 out of possible range 0~255
  analogWrite(enA, 200);
  // turn on motor B
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
  // set speed to 200 out of possible range 0~255
  analogWrite(enB, 200);
  delay(2000);
  // now change motor directions
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);
  delay(2000);
  // now turn off motors
```

```

digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW); }
void demoTwo() {
    // this function will run the motors across the range of possible speeds
    // note that maximum speed is determined by the motor itself and the operating voltage
    // the PWM values sent by analogWrite() are fractions of the maximum speed possible
    // by your hardware
    // turn on motors
    digitalWrite(in1, LOW); digitalWrite(in2, HIGH);
    digitalWrite(in3, LOW); digitalWrite(in4, HIGH);
    // accelerate from zero to maximum speed
    for (int i = 0; i < 256; i++) {
        analogWrite(enA, i);
        analogWrite(enB, i);
        delay(20); }
    // decelerate from maximum speed to zero
    for (int i = 255; i >= 0; --i)
    {
        analogWrite(enA, i);
        analogWrite(enB, i);
        delay(20);
    }
    // now turn off motors
    digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
}
void loop()
{
    demoOne();
    delay(1000);
    demoTwo();
    delay(1000);
}
}

```

MPU 6050 Diagram

FIG. BLOCK Diagram

