KTH ROYAL INSTITUTE
OF TECHNOLOGY

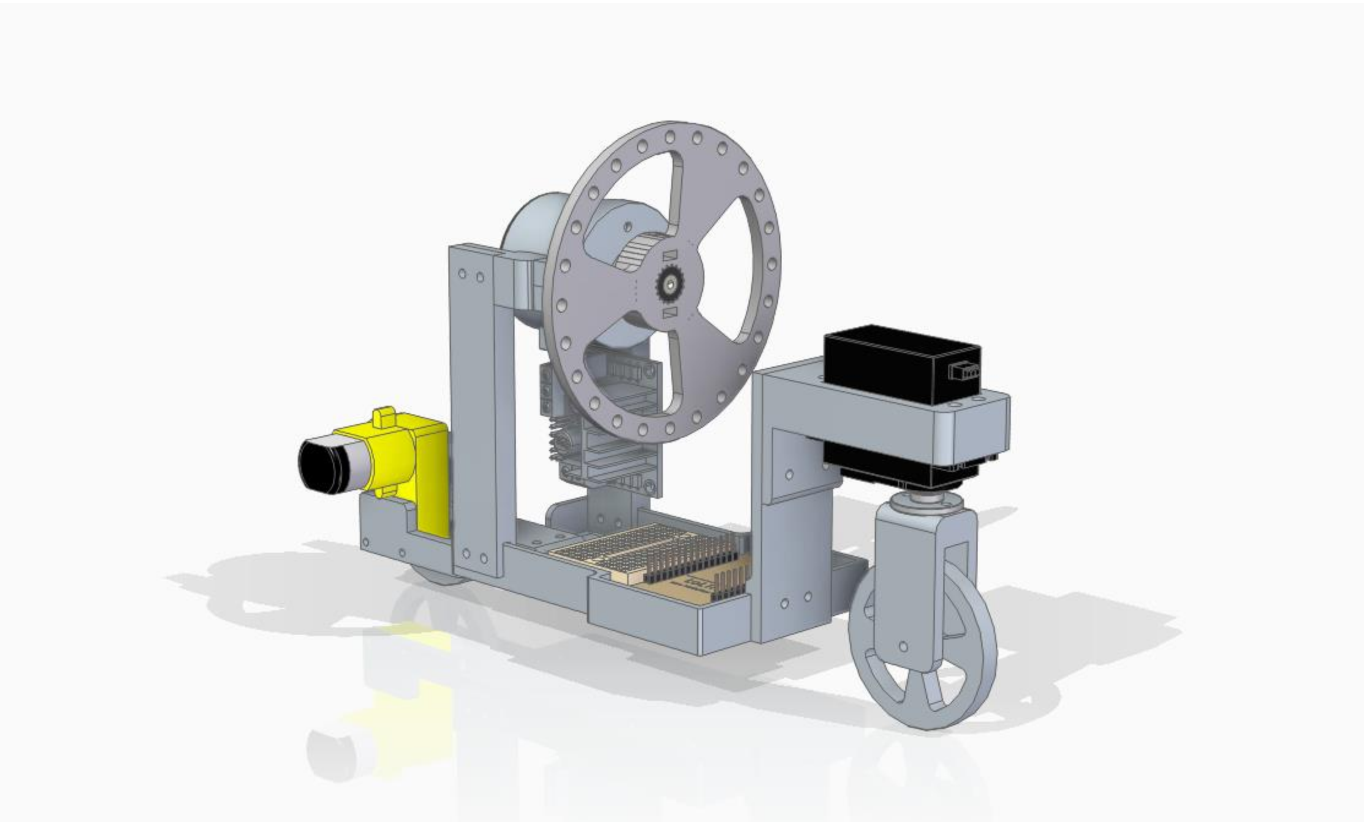
Degree Project in Technology

First cycle 15 credits

Reaction Wheel Control System for a Self-balancing Bike

Reaction wheel based control mechanism for a small scale prototype bike

REMAN SORYANI



Stockholm, Sweden 2023

TRITA – ITM – EX – 2023:87

Reaction Wheel Control System for a Self-balancing Bike

Reaction wheel based control of a small scale prototype bike

REMAN SORYANI

Degree Programme in Mechanical Engineering
Date: June 5, 2023

Supervisor: Daniel Frede
Examiner: Martin Edin Grimheden
School of Industrial Engineering and Management
Swedish title: Design och reglersystem för en självbalanserande cykel
Swedish subtitle: Reaktionshjulsbaserat reglersystem för småskalig prototyp cykel

Abstract

In this thesis, the design of a self balancing prototype bike with a reaction wheel as balancing mechanism will be investigated. Bikes are inherently unstable systems, that dynamically resemble an inverted pendulum at a stand still. Previous work has mostly been focused on keeping the balance through manipulating the steering input, and using state space modelling in order to balance the bike. In this project, a PID controller was implemented as a mean of control for the reaction wheel which was driven by a brushless DC motor. Different variations of Ziegler Nichols tuning methods were compared, and an alternative heuristic tuning method for the PID controller was also tested. The standard Ziegler Nichols and the alternative method yielded the best results.

Keywords

Mechatronics, Control Theory, Reaction Wheel, PID Controller, Robotics

Sammanfattning

I detta kandidatexamensarbete kommer designen av en självbalanserande prototyp cykel med ett reaktionshjul som balanseringsmekanism att undersökas. Cyklar är instabila system, som dynamiskt liknar en inverterad pendel när de är stationära. Tidigare arbeten har huvudsakligen fokuserat på att hålla balansen genom att elektroniskt manipulera styrinmatningen och använda modern reglerteknik för att balansera cykeln. I det här projektet implementerades en PID-regulator som ett medel för att styra reaktionshjulet, driven av en borstlös likströmsmotor. Olika varianter av Ziegler-Nichols inställningsmetoder jämfördes, och även en alternativ heuristisk inställningsmetod för PID-regulatorn testades. Den standard Ziegler Nichols-metoden och den alternativa metoden gav de bästa resultaten.

Nyckelord

Mekatronik, Reglerteknik, Mekanik, PID reglering, Robotik

Acknowledgments

I want to start of by giving thanks to the laboratory assistants Omar Kanbour and Filip Strand for helping along the way, and Daniel Frede for supervising and discussing solutions to problems. Lastly I want to thank my family for their persistent support.

Stockholm, June 2023
Reman Soryani

Contents

1 Introduction **1**
- 1.1 Background . . . 1
- 1.2 Problem . . . 2
- 1.3 Purpose . . . 2
- 1.4 Goals . . . 2
- 1.5 Research Methodology . . . 2
- 1.6 Delimitations . . . 3

2 Background **4**
- 2.1 Mechanics . . . 4
  - 2.1.1 Bike dynamics . . . 4
  - 2.1.2 Reaction wheels . . . 4
  - 2.1.3 System dynamics . . . 5
  - 2.1.4 Free body diagrams . . . 6
- 2.2 Control theory . . . 7
  - 2.2.1 PID controller . . . 7
  - 2.2.2 Ziegler-Nichols . . . 7
  - 2.2.3 Linear Quadratic Regulator . . . 8
  - 2.2.4 Kalman filter . . . 8
  - 2.2.5 High pass filter . . . 8
  - 2.2.6 Low pass filter . . . 9
- 2.3 Electronics . . . 9
  - 2.3.1 Microcontroller . . . 9
  - 2.3.2 IMU . . . 9
  - 2.3.3 DC motors . . . 9
  - 2.3.4 Motor drivers . . . 10

3 Method and Prototype **11**
- 3.1 Mechanical design . . . 11
  - 3.1.1 Bike design . . . 11
  - 3.1.2 Reaction wheel . . . 12

3.2 Internal Measurement Unit 12
3.2.1 Sensor fusion and filtration 13
3.3 Reaction wheel motor selection 13
3.4 Control system 13
3.4.1 Implementation 13
3.4.2 PID Tuning 14

4 Results 16
4.1 Final Design 16
4.2 IMU Performance 17
4.3 PID performance 18
4.3.1 Ziegler Nichols 18
4.3.2 Alternative tuning method 18

5 Discussion 20
5.1 Controller performance when balancing 20
5.2 Mechanical design of a balancing bike 20
5.3 Non linear system dynamics 21

6 Conclusions and Future work 22
6.1 Conclusions 22
6.2 Future work and recommendations 22

References 24

A Arduino Code 26

B Motor specification sheet 31

C ESP8266 specification sheet 33

List of Figures

2.1 Free body diagram of bike with reaction wheel, as seen from behind. Drawn by author. 6
2.2 Free body diagram of bike with reaction wheel, as seen from the side. Drawn by author. 6
2.3 The Ziegler-Nichols method, as described in Control System Design Guide, by George Ellis (2012). 7

3.1 Model of bike assembly, made in Solid Edge 2022. 11
3.2 Model of the reaction wheel, created in Fusion 360. 12
3.3 Flowchart explaining the code implemented on the microcontroller, made with draw.io. 14
3.4 Effects of varying PID constants, as outlined in PID control system analysis and design, by Yun Li et al. 15

4.1 Final prototype bike 16
4.2 IMU bike lean angle readouts while stationary 17
4.3 IMU bike lean angle readouts while balancing 18

List of Tables

4.1 Ziegler Nichols PID constants 18
4.2 Alternative tuning PID constants 19

List of acronyms and abbreviations

BLDC	Brushless Direct Current
CMG	Control Moment Gyroscope
CPU	Central Processing Unit
DC	Direct Current
DoF	Degrees of freedom
ESC	Electronic Speed Controller
I2C	Inter-Integrated Circuit
IGBT	Insulated-Gate Bipolar Transistor
IMU	Internal Measurement Unit
LiPo	Lithium polymer
LQR	Linear Quadratic Regulator
MEMS	Micro-electromechanical systems
MOSFET	Metal–Oxide–Semiconductor Field-Effect Transistor
PID	Proportional-Integral-Derivative
PWM	Pulse Width Modulation

Chapter 1

Introduction

This thesis will discuss the design and optimization of a self balancing small scale prototype electric bike. Both a stationary bike and one in motion are naturally unstable systems. This thesis examines the use of control algorithms in combination with electronics and a reaction wheel to stabilize this unstable system.

1.1 Background

The mechanics that describes the motion of a bike explain why a bike without proper steering input and rider movement is inherently an unstable system. This is why we find it hard to ride bike until we figure out the concept of counter steering, more often than not from experience, without further understanding of the physics.

In the case of a functioning riderless or balance assisted bike, there has to be a functioning control mechanism that hinders the bike from falling over, keeping the unstable system stable.

This has been researched previously in [1], [2] and [3] where use of a linear actuator to counter-steer, CMG (control moment gyro) and direct input into a servo motor has been used as stability control of a bike with various degrees of success.

This thesis will investigate the use of a reaction wheel for stability control. Reaction wheels are a common method of attitude control for spacecraft. The principle behind it stems from the conservation of angular momentum. The same principle can be applied to various applications where you need to counteract unwanted momentum in order to stay stable, in this case a bike.

1.2 Problem

In order to enable the functionality of a riderless bike or a balance assisting bike, stability control with a reaction wheel will be investigated. Proper evaluation of an assortment of control algorithms will be done in order to decide upon the most fit algorithm. Further more, the robustness of the control system application will be evaluated. Summarized, the following questions will be investigated:

1. Is it possible to stabilize a bike with the help of a reaction wheel mechanism?
2. What control algorithm is most fit for the application?
3. How stable is the system performance?

1.3 Purpose

The purpose of this report is to investigate the possibility of designing a control system in order to stabilize the unstable nature of a bike. Furthermore, there will be an examination of system behaviour and stability in order to understand the boundaries of aforementioned design. As a result, there will be a chance to draw a conclusion about the mechatronic design of the system and take note of strengths, weaknesses and potential improvements.

1.4 Goals

The goal of this project is to build a prototype electric bike that has self balancing capabilities. In order to approach this methodically, this has been divided into the following sub-goals:

1. Stabilize the bike in stationary position on a flat plane.
2. Stabilize the bike in movement on a flat plane.

For both of these sub-goals, there is an option to further investigate the research questions at hand about the mechatronic and control systems design.

1.5 Research Methodology

In order to answer the research questions there will be measurement of sensor data on the final product and analysis of live experiments in order to gather enough information to draw a conclusion.

1.6 Delimitations

Due to budget and time constraints the research will be conducted on a small scale prototype. The experiments will initially be conducted on a non moving bike. If time allows, small angles of steering input ($\pm 10°$) for a moving bike will be tested. This report will not dive deep into bike vehicular dynamics which is a rich and deeply intricate field, the focus is on the design of a mechatronic system and its accompanied control system.

Chapter 2

Background

2.1 Mechanics

2.1.1 Bike dynamics

Bike dynamics are the physical description of the forces acting upon a bike in motion.

Bikes are inherently unstable. This is one of the reasons why it is hard to steer and ride a bike. Bikes tend to turn with the lean of a rider, making it easier to turn the steering wheel in the direction of the lean. At the same time, this makes it more difficult to steer in the opposite direction.

At low speeds, bikes are less stable and require more effort to maintain balance and control. This is why riders need to constantly make small steering adjustments to keep the bike upright.

At high speeds, bikes become more stable due to the gyroscopic effect and construction geometry, which helps keep them upright. However, they also become more sensitive to inputs from the rider, such as steering, acceleration, and braking. This means that small movements or changes in speed can have a significant impact on the bike's stability and handling.

At a standstill, bikes are at their most unstable state, since there is zero gyroscopic effect to help them keep upright.

To summarize, bike dynamics is a complex topic that involves various factors, including geometry, weight distribution, tire deformation, and rider input. [4]

2.1.2 Reaction wheels

Reaction wheels, also known as momentum wheels are a device commonly used in satellites and other smaller spacecraft for attitude control. The construction for the device is rather simple, a disc mounted on an axis, commonly powered by some kind of electric motor. The output torque of the reaction wheel is related to the motor torque, and the moment of inertia, this relation is displayed in the next subsection. The principle of a reaction wheel is to apply torque in the opposite direction of the wanted rotation, as the torque from the reaction wheel will be the cause of an equal reaction torque, that is directed antagonistic to the reaction wheel torque. Reaction wheels are good in the use case where you would want stable and precise rotation in an axis. [5]

2.1.3 System dynamics

The system at hand that is to be investigate and controlled by a feedback loop is the instability of a bike. Assuming a steering input angle of $0°$, lean angle of Θ and ignoring the slip angles at the wheel leads to the free body diagram displayed in figure 2.1. The free body diagram demonstrates that the dynamics of the system have been simplified to a inverted pendulum with a reaction wheel mounted to it. From the free body diagram previously mentioned, Newtons II law can be used to describe the equations for the dynamical system as following:

$$e_Z{:}I_T\ddot{\Theta} = m_T g l_C \sin\Theta - M_m \tag{2.1}$$

Where I_T is the total moment of inertia of the system, m_T is the total mass of the system, and M_m is the torque generated from the electric motor attached to the reaction wheel. There is also a possibility to describe the relation between the electric machine torque and the rotational speed of the flywheel with the same approach:

$$e_Z{:}I_S\dot{\Phi} = M_m \tag{2.2}$$

2.1.4 Free body diagrams

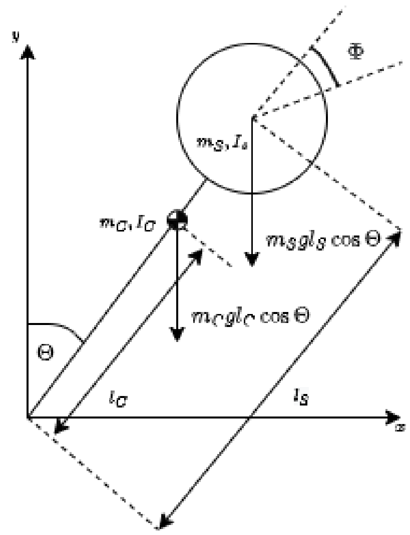


Figure 2.1: Free body diagram of bike with reaction wheel, as seen from behind. Drawn by author.

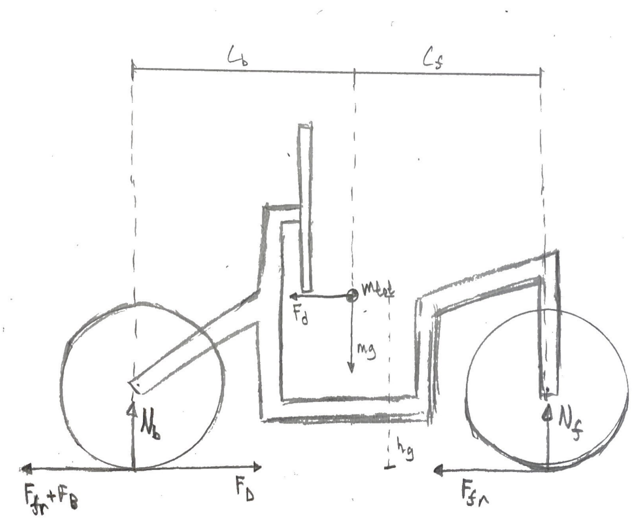


Figure 2.2: Free body diagram of bike with reaction wheel, as seen from the side. Drawn by author.

2.2 Control theory

2.2.1 PID controller

PID control is the single undisputed most common control method in the industry. PID control stems from building on a simple control method, called proportional control which on its own can increase the speed of the feedback system. The integrating effect (I), allows the system to keep the output signal steady and constant when approaching the sought after value. The derivative effect (D), notices the change over time in system performance and adjusts accordingly. This derivative effect keeps systems from going unstable when increasing the proportional and integral gain.

The full PID controller can be described mathematically as following:

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau)\, d\tau + K_D \frac{d}{dt} e(t)$$

Where $u(t)$ is the input signal to the system, which depends on the error term $e(t) \ = r(t) - y(t)$, where $r(t)$ is the reference value and $y(t)$ is the measured value. The constant values K_P, K_I, K_D are the values meant to be tuned to the system in order to find satisfying performance. [6]

2.2.2 Ziegler-Nichols

Ziegler-Nichols is a common approach to tuning PID controllers. Its based on finding the peak K_P value at initial instability, referred to as K_{MAX} and also the frequency f_0 for the oscillation at that point of instability. Based on those values the Ziegler-Nichols method defines your K_P, K_I and K_D for the controller. [7] [6]

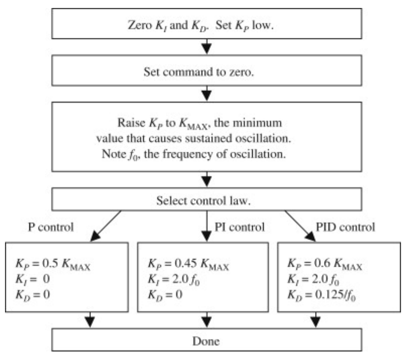


Figure 2.3: The Ziegler-Nichols method, as described in Control System Design Guide, by George Ellis (2012).

2.2.3 Linear Quadratic Regulator

Linear quadratic regulators, commonly referred to as LQR, is another control method based on state space modelling. A common state space model could be defined as following:

$$\dot{x} = Ax + Bu$$

$$y = Cx$$

Where A, B, C are matrices. Together, this model describes the system dynamics that you want to control. In order to control this system we use the input signal u as following:

$$u = -Lx + \overrightarrow{r}$$

Which gives the closed system equation as following:

$$\dot{x} = (A - BL)x + B\overrightarrow{r}$$

$$y = Cx$$

For a state space model, there usually is tuning of the parameter L, which is the gain for the system feedback. The vector $\overrightarrow{r}$ is the reference vector for the system. The idea of a linear quadratic regulator is to use a state space model and optimize the value of L by minimizing a quadratic cost equation with the help of the algebraic Riccati-equation. LQR control is a powerful tool, but requires good knowledge and modelling of the system. [6]

2.2.4 Kalman filter

A Kalman filter is a recursive estimator that is commonly used when working with noisy data. Using statistical methods, the Kalman filter estimates the optimal (in the case of Gaussian data) value of a certain noisy data point, given that it has seen the previous data point and its accompanied noise. If the data that the Kalman filter is applied to is not normally distributed, the Kalman filter is no longer an optimal estimator, but it is the best linear estimator. [8] [9]

2.2.5 High pass filter

A high pass filter is a common tool used in signal processing. There are specific cases in which there is a need to transmit higher frequency signals and block out the signals at a lower frequency. For these cases a high pass filter is highly suitable. A high pass filter allows the wanted frequencies, called the passband through and filters out the lower frequencies of the signals, also known as the stopband. The intersection of these frequency ranges is called the cutoff frequency. [10]

2.2.6 Low pass filter

Low pass filtration is a common tool within signal processing. For the case when lower frequency signals are wanted and high frequency signals are unwanted, a low pass filter is desirable. The low pass filter allows the passband, which is the allowed range of frequencies to pass through. The stopband, which in this case is the range of high frequency signals, are filtered out. Shortly put, the low pass filter has its passband below the cutoff frequency and stopband above the cutoff. [10]

2.3 Electronics

2.3.1 Microcontroller

Microcontrollers are small computers on a single integrated circuit that are designed to control specific tasks such as data logging and signalling output. It is a versatile device that consists of a CPU, memory, and input/output peripherals, all integrated onto a single chip. They are programmable and can be configured to control specific functions, making them very adaptable to a wide range of applications. For prototyping purposes, microcontrollers on development boards have easy to use interfaces which makes it easy to flash code onto the chip and iteratively develop a project. [11]

2.3.2 IMU

Internal measurement units (IMU) are devices that combines the use of accelerometers, gyroscopes and occasionally magnetometers to give readouts of the acceleration and angular rates of the device it is attached to. In modern day technology there is a widespread use of electric MEMS IMUs, that in a small package can deliver this combination of accelerometer and gyroscope in order to give sensor readouts to a system. [12] [13]

2.3.3 DC motors

There are two distinct different types of DC (direct current) motors.

Brushed motors use brushes and a commutator to transfer electrical power to the rotating armature. The armature, wound with wire coils, rotates within a magnetic field generated by permanent- or electromagnets. When the armature rotates, the brushes make contact with the commutator, this switches the direction of current flow in the armature coils and keeps the motor spinning continuously. Brushed motors are due to their construction a cause for electrical noise and arcing, which can lead to worse motor performance and electromagnetic disturbances for other electrical components.

Brushless motors, also known as BLDC motors, do not use brushes or a commutator to transfer electrical power to the armature. They use electronic controllers and sensors to control the timing and direction of the current flow through the motor windings. Permanent magnets or electromagnets are mounted on the rotor, while the stator contains the windings that generate the rotating magnetic field. They are less prone to generating electrical noise and arcing, which as mentioned can degrade the performance of brushed motors and other surrounding electronics. [14] [11]

2.3.4 Motor drivers

Motor drivers are electronic devices that control speed, direction, and torque of electric motors. Most commonly used are H-bridge drivers for brushed motors and three-phase motor drivers for BLDC motors.

H-bridge drivers for brushed motors are electronic circuits that use four transistors to control the direction and speed of the motor. By switching the transistors in a specific order, the motor driver will regulate the voltage and current delivered to the motor, allowing it to rotate at different speeds and in different directions.

Three-phase motor drivers for BLDC motors are more complex than H-bridge drivers and require additional electronic components such as MOSFETs or IGBTs. A very common type of driver for this application is called an ESC, short for electronic speed driver, which simplifies the control of BLDC motors with a basic microcontroller. [11]

Chapter 3

Method and Prototype

3.1 Mechanical design

3.1.1 Bike design

The construction of the bike was initially designed with CAD software (as seen in figure 3.1) to ensure both functionality and fit of components and the overall package. This was done to reduce iterative construction of mechanical components and instead focus testing on the interplay between the different parts of the mechatronic system. The bike was designed to be relatively modular, in the case of any design or component changes based on observations during testing. All of the parts of the assembly were designed with 3D-printing in mind, as this was the most suitable manufacturing process for the prototype.

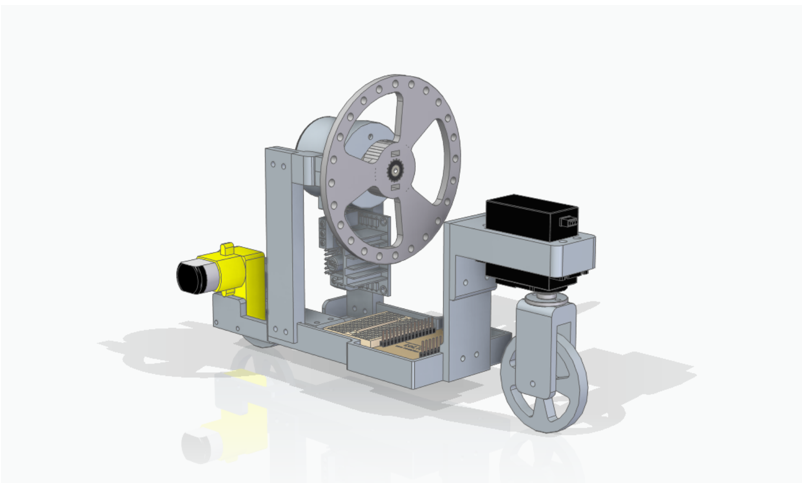


Figure 3.1: Model of bike assembly, made in Solid Edge 2022.

3.1.2 Reaction wheel

In order for the bike to balance, the reaction wheel has to have enough moment of inertia to be able to counteract the torque generated from gravity acting upon the bike, as seen in equation (2.1).

The equation for the moment of inertia for a body that lies two dimensionally in the X-Y plane formulated as follows:

$$I_Z = \sum_i m_i r_i^2$$

As the moment of inertia for a body is proportional to the square of the radial distance to the mass, the reaction wheel is designed in a way where majority of the mass is placed on the extremities of the wheel, in order to increase I_S. In order to achieve such a weight distribution and to have the ability to vary the moment of inertia easily, a 3D printed design where the majority of the mass would come from a combination of M4 screws and nuts was chosen.

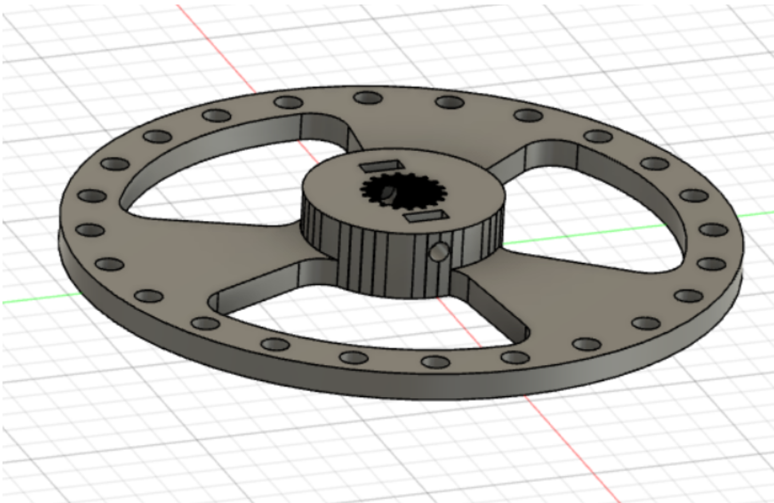


Figure 3.2: Model of the reaction wheel, created in Fusion 360.

3.2 Internal Measurement Unit

The IMU is an integral device for this project. A MPU6050 GY-521 6-DoF accelerometer and gyroscopic sensor was selected for this project due to it being readily available and low cost. The communication between the IMU and the ESP8266 is handled through I^2C protocol at 400kHz. The yaw angle was measured in order to decide the lean angle of the bike.

3.2.1 Sensor fusion and filtration

In order to accommodate for gyroscopic drift and accelerometer readout noise, a complementary filter was used in order to create a simple implementation of sensor fusion. The aim of the complementary filter was to eliminate majority of the drift and decrease the noise. The data from the accelerometer readouts were low pass filtered and the gyroscopic readouts were high pass filtered.

3.3 Reaction wheel motor selection

The motor that drives the reaction wheel is integral to the project and needed to fulfill a certain list of criteria: High torque to weight ratio, good amount of motor speed while still providing high torque, no backlash, minimize electromagnetic interference.

Initially tests were conducted with a DC-motor, which produced a lot of electromagnetic noise, resulting in disturbance of the I^2C communication between the ESP8266 and the IMU. DC-motors also have the issue of either lacking speed or torque. Gearboxes that are mounted to DC-motors within the price range for the project often come with a lot of backlash, which is not desirable.

The chosen motor for the project was a Nidec BLDC-motor with a built in ESC, enabling easy control through PWM from the microcontroller. This motor had adequate speed and torque output, while being relatively light with small packaging.

The IMU froze less frequently when using this motor, but the I^2C communication was not deemed satisfactory until the motor cables and female connectors of the DuPont cables for the IMU were shielded with conductive aluminium foil.

3.4 Control system

3.4.1 Implementation

PID Control was chosen as the method of control for this project. The reasoning behind the chosen control method are three-fold. Majority of the previous similar work had been done with control methods that are based on state space modelling, which makes use of idealisation and linearizing that may not always relate well to the reality when the system is very idealized. Many systems do not have models at all, and the model at hand for this project is partially unknown, since a generously idealised model has been derived in the background. For these kind of systems where there is limited knowledge about the physical model, PID controllers can be preferable. PID controllers are the most commonly used industry control method, which makes it interesting to study the system behaviour under PID control.

The PID controller was implemented within the code that runs on the ESP8266 microcontroller. The flowchart on display in figure 3.3 explains the logic of the controller. The given input is the lean angle of the bike, the reference point is initially 0, but the target set point is moved with time in order to keep the bike more stable. The output of the PID controller sets the PWM signal which controls motor speed through voltage duty cycle control.

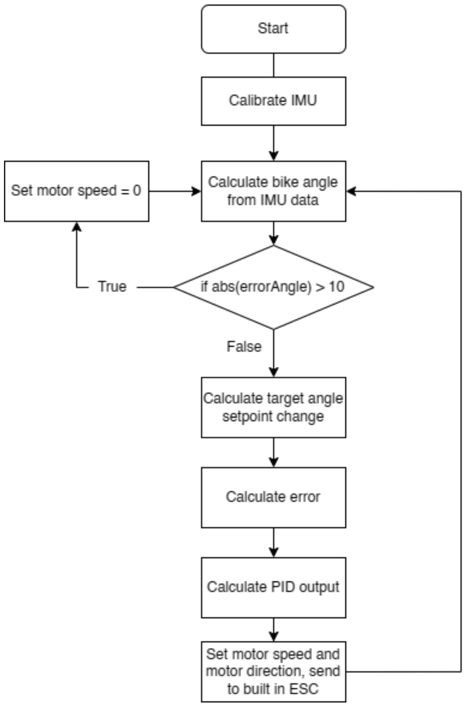


Figure 3.3: Flowchart explaining the code implemented on the microcontroller, made with draw.io.

3.4.2 PID Tuning

Attempts were made to use the Ziegler-Nichols method to find satisfactory PID constants. Since the system is naturally unstable, an observation during tests was that it is hard to find an appropriate K_{MAX} value, since constant sustained oscillation would not occur. An appropriate K_{MAX} was found at $K_P = 75$. The oscillation had an approximate frequency $f_0 = 4.27Hz$. Several different Ziegler-Nichols parameters were tested, outlined by Microstar Laboratories [15]. The classic ZN, less overshoot and no overshoot approach were used.

Another tuning approach was tested, based on using the K_{MAX} value, starting with setting $K_P \approx 0.7 K_{MAX}$. Afterwards the system behaviour was observed and K_I, K_D carefully adjusted according to the system response from tuning these constants as outlined by the chart in figure 3.4.

TABLE 1 Effects of independent P, I, and D tuning on closed-loop response. For example, while K_I and K_D are fixed, increasing K_P alone can decrease rise time, increase overshoot, slightly increase settling time, decrease the steady-state error, and decrease stability margins.

	Rise Time	Overshoot	Settling Time	Steady-State Error	Stability
Increasing K_P	Decrease	Increase	Small Increase	Decrease	Degrade
Increasing K_I	Small Decrease	Increase	Increase	Large Decrease	Degrade
Increasing K_D	Small Decrease	Decrease	Decrease	Minor Change	Improve

Figure 3.4: Effects of varying PID constants, as outlined in PID control system analysis and design, by Yun Li et al.

This approach was better suited to the system and yielded better results. To reduce the risk of saturation due to integral windup, the integral output was clamped to 50 percent of the total PWM value.

Chapter 4

Results

In this chapter the results from the project will be presented. The design of the prototype, IMU performance, and different PID parameter values will be on display. Notable system behaviours will be presented.

4.1 Final Design

Figure 4.1 shows the final design of the prototype. While it is a compact design, some issues have appeared during testing. The main issues with the design seem to directly affect the overall system behaviour.

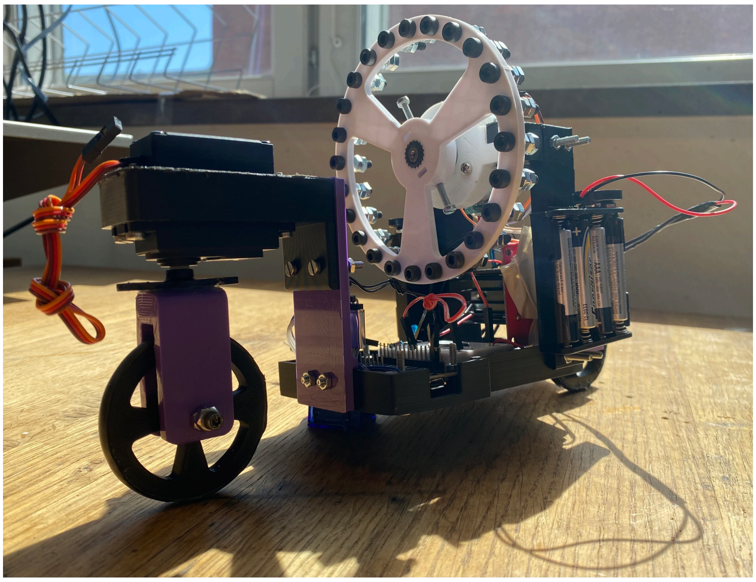


Figure 4.1: Final prototype bike

The center of gravity is positioned rather high, while the actual chassis has a low ride height, majority of the weight comes from the motor, reaction wheel, and battery which all are placed relatively high up the bike. This makes the bike being more prone to tipping.

The bike frame proved to not be the most rigid construction which was made apparent during testing when the angular velocity of the reaction wheel reached towards the highest motor speeds, this led to resonance frequencies that made the whole bike shake about.

4.2 IMU Performance

Implementing sensor fusion with a complementary filter worked well in order to make the IMU reliable for the majority of the time. Figure 4.2 shows the IMU readouts at a standstill, with the bike lifted by the chassis placed on a level surface. Figure 4.3 shows the IMU readouts when attempting balance on a flat surface. The weight of the complementary filter was kept to 98 percent gyroscopic readouts, and 2 percent accelerometer readout.

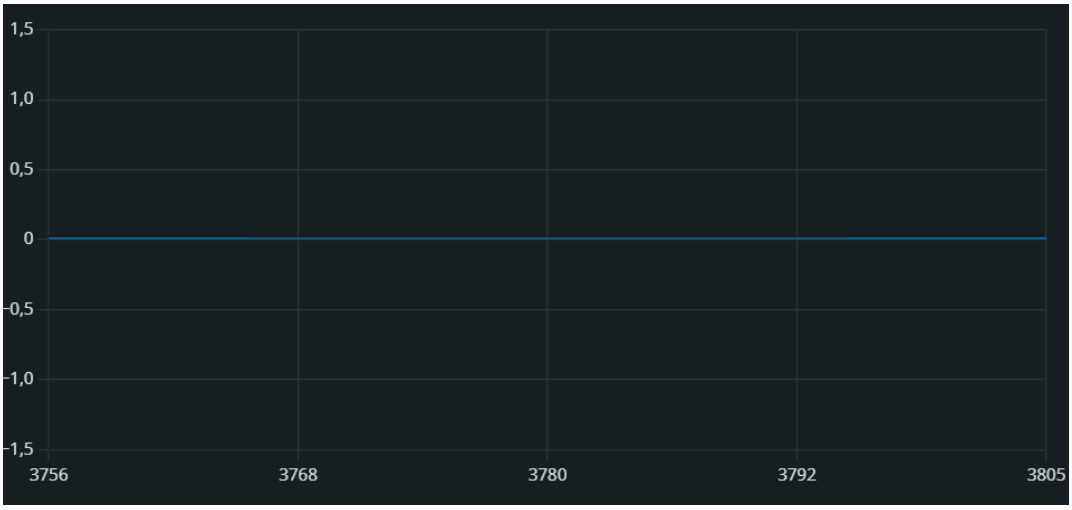


Figure 4.2: IMU bike lean angle readouts while stationary

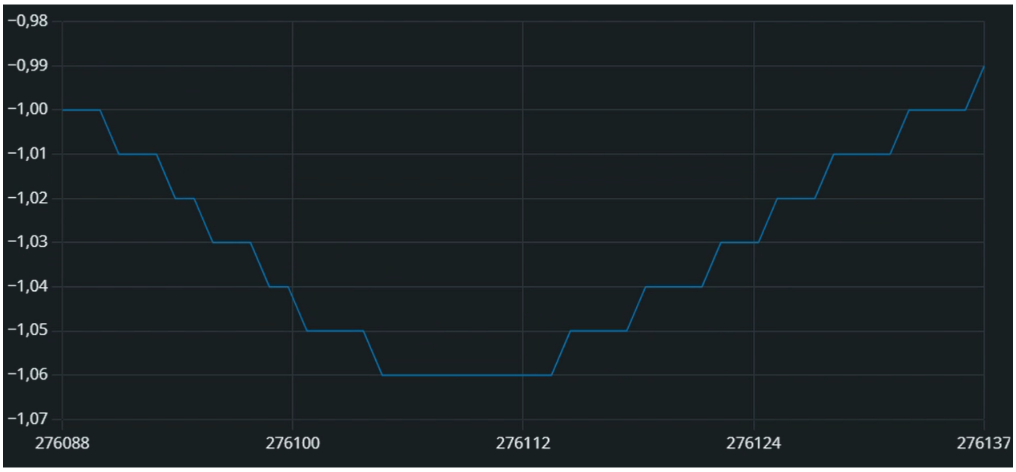


Figure 4.3: IMU bike lean angle readouts while balancing

4.3 PID performance

4.3.1 Ziegler Nichols

All versions of Ziegler Nichols failed to stabilize the system to different extents. The system was observed to have an estimated ultimate gain of $K_{MAX} = 75$. The original Ziegler Nichols settings had a very noticeables advantage in system behaviour compared to the less overshoot and no overshoot versions. However, these PID settings ultimately failed to balance the robot even momentarily. Due to the poor performance from the less overshoot approach and no overshoot approach, no data has been provided from these tests.

Table 4.1: Ziegler Nichols PID constants

	ZN standard	ZN less overshoot	ZN no overshoot
K_P	45	24.8	15
K_I	0.12	0.12	0.12
K_D	0.03	0.08	0.08
Best vertical time (s)	7.1		

4.3.2 Alternative tuning method

Since it was hard to determine both the ultimate gain and especially the ultimate period, another tuning method was tested. Similarly to Ziegler Nichols, after finding the ultimate gain the proportional gain is set to $K_P \approx 0.7K_{MAX}$, then K_I, K_D were heuristically tuned, with the starting point of the values given by the standard Ziegler Nichols method. The proportional gain was then fine tuned slightly. This method worked better than the Ziegler Nichols method and showed more promising performance.

Table 4.2: Alternative tuning PID constants

	ZN + Heuristic tuning
K_P	48
K_I	1.2
K_D	2
Best vertical time (s)	10.7

Chapter 5

Discussion

5.1 Controller performance when balancing

The PID controller did show some promise during testing as a mean to balance the bike. This makes it seem like a control method that could reach acceptable performance when balancing a stationary bike, although more tuning of the PID controller has to be made in order to find parameters which yield good performance results.

A common issue that was found during testing was that the bike found a stable position, but a slight offset made the reaction wheel begin to accelerate in order to counteract the torque acted upon the bike. This led to the motor slowly accelerating, less torque is then applied to the bike as the torque is proportional to the speed of the reaction wheel. A solution to this issue might be to move the set point more aggressively. This way, the motor would have to accelerate the speed at a faster rate, generating more torque from the reaction wheel.

5.2 Mechanical design of a balancing bike

A very important factor in order to make a balancing robot tuneable is the construction of the object. The bike was tough to tune as it could not deviate a lot from its upright position before falling over. Shifting the center of mass lower would most certainly help when tuning the system to perform better, as it would be more resistant to tipping, allowing for a broader range of angles before losing control.

During tests where the motor reached peak speeds or switched directions at a high rate, resonance frequencies were making the whole construction shake. A more rigid construction may have been less susceptible to these oscillations that affected system performance by making the tires lose grip with the ground.

Scaling up the experiment to a full size bike could potentially lead to better performance. A full scale bike would be constructed with the proper strength and stiffness characteristics of a bike frame together with precise bike construction geometry. This would probably decrease the vibrations and increase the balance of the overall system through having more precise and centered placement of center of mass and the wheel axles.

5.3 Non linear system dynamics

The balancing of a bike can be described as an inverted pendulum problem, as shown before in figure 2.1. For small angles we can assume this is a linear problem by formulating it with the help of linearization. However, at larger angles, small angle approximation no longer holds up. This leads to the approximation becoming less accurate at larger angles, and as PID controllers are linear controllers, it can be extremely tricky to find a set of parameters that hold up for an increasingly non linear system. In order to make a PID work more reliably in a non linear system, modifications to the controller such as gain scheduling which modifies the controller parameters based on the current operation condition. Methods of this nature could adjust the performance of the PID controller to be more fitting to non linear systems, ultimately leading to better control.

Chapter 6

Conclusions and Future work

6.1 Conclusions

Testing has made it clear that there needs to be further evaluation regarding how stable the system can become with the help of a PID controller. The results hint at the possibility that PID control is an adequate control method for a self balancing bike.

The construction of the bike is a contributor to the poor system performance as the center of mass is placed too high which makes tuning of the PID tough, since the bike has a very narrow range of angles from which it can recover to its already unstable equilibrium point.

Previous research and the findings during this project suggests that designing a self balancing bike is possible, but the selected control method and chosen implementation of balance mechanism is fundamental to system performance.

As most previous successful work has been carried out with control methods that make use of state space models, there is an argument to be made that these more modern control methods might be advantageous to use over the more classical PID control, due to the somewhat underwhelming performance presented in the results.

Further research has to be conducted to evaluate the control system under vehicular motion, as this particular trial is left to be tested due to time constraints.

6.2 Future work and recommendations

A comparison made specifically between a modern state space control method like LQR and the traditional PID controller has to be conducted in order to decide what control method is best to pursue for a reaction wheel balanced bike.

Further research has to be done in order to find out how the reaction wheel bike behaves under motion, and what control method works best for this application.

The MPU6050 IMU did not have a built in magnetometer. A direct consequence of this was the disturbance of gyroscopic data, as the IMU constantly was susceptible to gyroscopic drift. Future work could be conducted with a higher precision 9-DoF IMU with higher precision data collection. Combined with a well implemented Kalman filter, there would be a possibility to examine how precise and consistent the angle readouts for the lean angle of the bike can become and how this affects system performance.

It would be advisable to carry out similar projects with a stable power source, where the voltage in the system can be kept constant. Rechargeable Li-Po batteries would be suitable, as their low weight and ability to keep the voltage constant between tests will increase system consistency which would be good for tuning purposes.

A more rigid chassis construction with a lower center of mass would be a good idea to implement, in order to make it easier for the system to balance, as it currently is too prone to tipping over due to the high center of mass and losing contact with the ground due to resonance frequencies making the entire construction shake.

More research has to be conducted on full scale bikes in order to investigate the control mechanisms potential for real life applications.

References

[1] Z. Fawaz, "_USDesign and Control of a Self-balancing Bicycle Using an Electric Linear Actuator," Thesis, Apr. 2019, accepted: 2019-05-03T14:37:58Z. [Online]. Available: http://deepblue.lib.umich.edu/handle/2027.42/148871 [Page 1.]

[2] F. G. Li, "Research on self-balancing unmanned bicycle based on cascade PID control," vol. 2, Dec. 2021. doi: 10.1109/ICIBA52610.2021.9687999 pp. 442–446. [Page 1.]

[3] P. Yuan Lam, "Design and Development of a Self-Balancing Bicycle Using Control Moment Gyro - CORE," Ph.D. dissertation. [Online]. Available: https://core.ac.uk/display/48657545?utm_source=pdf&utm_medium=banner&utm_campaign=pdf-decoration-v1 [Page 1.]

[4] J. Fajans, "Steering in bicycles and motorcycles," *American Journal of Physics*, vol. 68, no. 7, pp. 654–659, Jul. 2000. doi: 10.1119/1.19504 Publisher: American Association of Physics Teachers. [Online]. Available: https://aapt.scitation.org/doi/10.1119/1.19504 [Page 4.]

[5] D. Bender, "Chapter 10 - Flywheels," in *Storing Energy*, T. M. Letcher, Ed. Oxford: Elsevier, Jan. 2016, pp. 183–201. ISBN 978-0-12-803440-8. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128034408000105 [Page 5.]

[6] T. Glad and L. Ljung, *Reglerteknik - Grundläggande teori*, 4th ed. Studentlitteratur, 2006. ISBN 978-91-44-02275-8 [Pages 7 and 8.]

[7] G. Ellis, "Chapter 6 - Four Types of Controllers," in *Control System Design Guide (Fourth Edition)*. Boston: Butterworth-Heinemann, Jan. 2012, pp. 97–119. ISBN 978-0-12-385920-4. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780123859204000060 [Page 7.]

[8] L. Kleeman, "Understanding and Applying Kalman Filtering." [Page 8.]

[9] "What is an Inertial Measurement Unit?" [Online]. Available: https://www.vectornav.com/resources/inertial-navigation-articles/what-is-an-inertial-measurement-unit-imu [Page 8.]

[10] L. P. Huelsman, "Analog Electrical Filters," in *Encyclopedia of Physical Science and Technology (Third Edition)*, R. A. Meyers, Ed. New York: Academic Press, Jan. 2003, pp. 519–530. ISBN 978-0-12-227410-7. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B0122274105000235 [Page 9.]

[11] H. Johansson, *Elektroteknik*. Stockholm: Institutionen för maskinkonstruktion, Tekniska högsk., 2006. [Pages 9 and 10.]

[12] L. Kleeman, "Understanding and Applying Kalman Filtering," Tech. Rep., Jan. 1996. [Online]. Available: https://www.cs.cmu.edu/afs/.cs.cmu.edu/Web/People/motionplanning/papers/sbp_papers/kalman/kleeman_understanding_kalman.pdf;https://scholar.google.com/citations?view_op=view_citation&hl=en&user=Q590I6YAAAAJ&citation_for_view=Q590I6YAAAAJ:qxL8FJ1GzNcC [Page 9.]

[13] B. Or, PhD, "What is IMU?" Nov. 2022. [Online]. Available: https://towardsdatascience.com/what-is-imu-9565e55b44c [Page 9.]

[14] "Electric motor," Feb. 2023, page Version ID: 1141785239. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Electric_motor&oldid=1141785239 [Page 10.]

[15] "Ziegler-Nichols Tuning Rules for PID." [Online]. Available: https://www.mstarlabs.com/control/znrule.html [Page 14.]

Appendix A

Arduino Code

Degree project

Reman Soryani

```
#include
float yawRate;
float accX, accY;
float accAngle = 0, gyroAngle = 0;
float degConv = 57.2957;
float gyroPart = 0.98;
float accPart = 0.02;
float errorAngle = 0;
float earthRotSpeed = 0.004167;

int16_t gyX, gyY, gyZ;
int16_t accXraw, accYraw, accZraw;
float calibrationYawRate;
int calibrationCounter;
float deltaT = 0, currentTime = 0, previousTime = 0;

int32_t freq = 20000;
const byte nidecPWM = 14; //D5
int PWM_CH = 1;
int dir = 2; //D4 always HIGH at boot
int brake = 0; //D3 Constant HIGH

float KP = 48;  //
float KI = 1.2; //.14;  //
float KD = 2;  //
float error;
float previousError;
float targetAngle = 0;
float integral;
float integralPart;
float derivative;
float PID;
int pwmCalc;
int pwmOutput;
float angleBounce = 0.1; //0.1;
int i = 0;

const byte dcPWM = 12;
const byte servoPWM = 13;
const byte led = 16;


void setup() {
  //Motor setup
  pinMode(nidecPWM, OUTPUT);
  analogWriteFreq(20000);
  analogWrite(nidecPWM, 255);

  Serial.begin(9600);

  pinMode(led, OUTPUT);
  delay(500);               //Calibration warning
  digitalWrite(led, HIGH);
  delay(100);
  digitalWrite(led, LOW);
  delay(5000);

  //MPU6050 setup
  Wire.setClock(400000);
  Wire.begin();
  delay(250);
  Wire.beginTransmission(0x68);
```

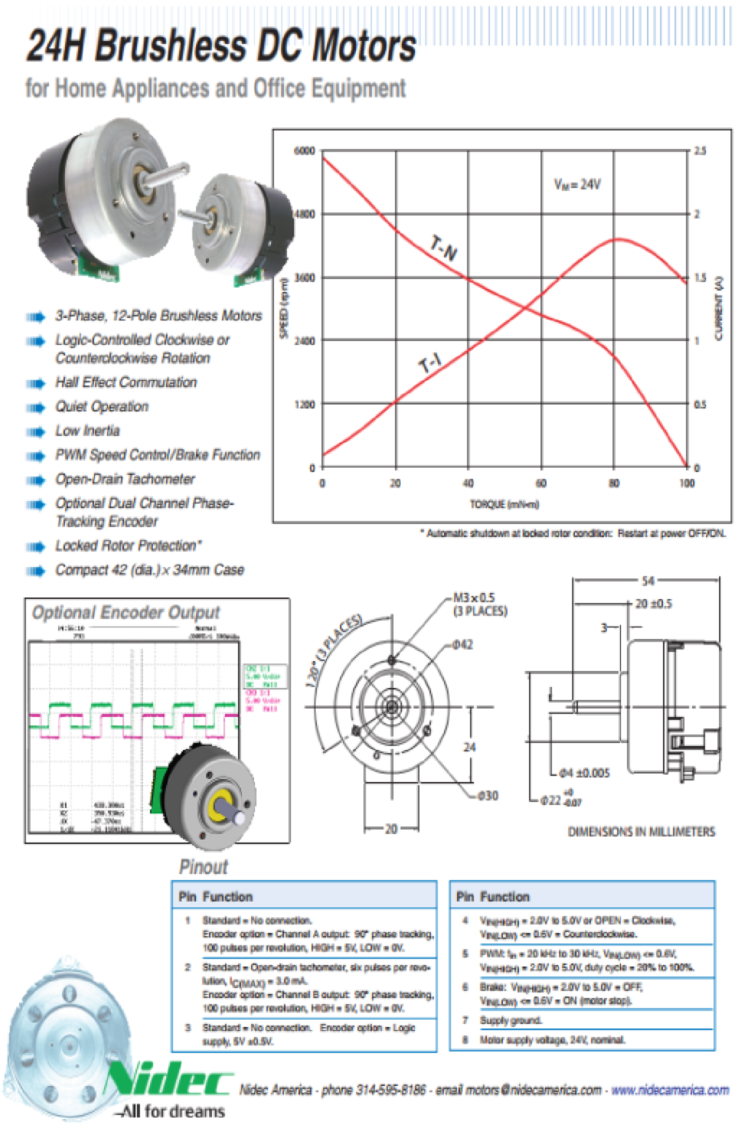
```
  Wire.write(0x6B);
  Wire.write(0x00);          //PWR management
  Wire.endTransmission();
  //Calibration step
  for (calibrationCounter = 0; calibrationCounter < 1000; calibrationCounter ++) {  //Calibrate on each startup
    gyroRead();
    calibrationYawRate += yawRate;
    delay(1);
  }
  calibrationYawRate /= 1000;  //Divide by # measurements

  pinMode(brake, OUTPUT);
  //digitalWrite(brake, HIGH);

  pinMode(dir, OUTPUT);
  //digitalWrite(dir, HIGH);

  delay(1000);              //Calibration over, 3 second warning
  digitalWrite(led, HIGH);
  delay(1000);
  digitalWrite(led, LOW);
  delay(3000);
}

void gyroRead(void) {
  Wire.beginTransmission(0x68);
  Wire.write(0x1A);
  Wire.write(0x05);          //Enable low pass  //prev 0x05
  Wire.endTransmission();
  Wire.beginTransmission(0x68);
  Wire.write(0x1B);          //Gyro data
  Wire.write(0x8);           //LSB 65.5, +-500 deg/s
  Wire.endTransmission();
  Wire.beginTransmission(0x68);
  Wire.write(0x43);        //Gyro Register
  Wire.endTransmission();
  Wire.requestFrom(0x68,6);

  gyX = Wire.read()<<8 | Wire.read();   //Ignore X, Y
  gyY = Wire.read()<<8 | Wire.read();
  gyZ = Wire.read()<<8 | Wire.read();

  yawRate = (float)gyZ/65.5;    //LSB -> deg/s

}

void accRead(void){

  Wire.beginTransmission(0x68);
  Wire.write(0x1A);
  Wire.write(0x05);          //Enable low pass
  Wire.endTransmission();
  Wire.beginTransmission(0x68);
  Wire.write(0x1C);          //Acc data
  Wire.write(0x10);          //LSB 4096, +-8g
  Wire.endTransmission();
  Wire.beginTransmission(0x68);
  Wire.write(0x3B);          //Acc Register
  Wire.endTransmission();
  Wire.requestFrom(0x68,6);

  accXraw = Wire.read() << 8 | Wire.read(); //Ignore Z
  accYraw = Wire.read() << 8 | Wire.read();
  accZraw = Wire.read() << 8 | Wire.read();

  accX = ((float)accXraw/4096) - 0.05;  //LSB -> g, Adjust for sensorerror
  accY = ((float)accYraw/4096);

}
```

```

void loop() {

  //Loop timer
  previousTime = currentTime;        // Previous time is stored before the actual time read
  currentTime = millis();            // Current time actual time read
  deltaT = ((currentTime - previousTime) / 1000); // Divide by 1000 to get seconds
  gyroRead();
  accRead();

  yawRate -= calibrationYawRate;     //Adjust for sensor error
  yawRate -= earthRotSpeed;          //Adjust for constant rotation of earth
  accAngle = -atan2(accY, accX)*degConv;

  //Serial.print(yawRate);
  //Serial.print(",");

  if(i > 0){
    gyroAngle = gyroAngle + (yawRate * deltaT);
  }
  //else{
  //  gyroAngle = 0;
  //}

  //Complementary filter 0.98 high pass gyro, 0.02 low pass acc
  errorAngle = gyroAngle * gyroPart + accAngle * accPart;
  //Serial.print(gyroAngle);
  //Serial.print(",");
  //Serial.print(accAngle);
  //Serial.print(",");
  Serial.print(errorAngle);
  Serial.print(",");

  if(i > 0){
    if(errorAngle < targetAngle){     //Enable target setpoint move
      targetAngle -= angleBounce * deltaT;
    }
    else if(errorAngle > targetAngle){
      targetAngle += angleBounce * deltaT;
    }
    else{
      targetAngle = 0;
    }
  }

  Serial.print(targetAngle);
  Serial.print(",");

  //PID Controller
  error = targetAngle - errorAngle;
  integral += error * deltaT;
  integralPart = constrain(integral, -64, 64);      //Anti windup esque function
  derivative = (error - previousError) / deltaT;
  previousError = error;
  PID = KP * error + KI * integralPart + KD * derivative;

  //Motor control
  pwmCalc = constrain(PID, -255, 255);
  Serial.println(pwmCalc);


  if (abs(errorAngle) > 25){
      pwmOutput = 255; //Nidec reverse duty cycle
      digitalWrite(brake, HIGH);
      digitalWrite(dir, HIGH);
      analogWriteFreq(freq);
      analogWrite(nidecPWM, pwmOutput);
  }
  else if (pwmCalc <= 0) {
```

```
      pwmOutput = abs(pwmCalc);
      pwmOutput = map(pwmOutput, 0, 255, 255, 0); //Adjust duty cycle for Nidec motor
      digitalWrite(brake, HIGH);
      digitalWrite(dir, HIGH);
      analogWriteFreq(freq);
      analogWrite(nidecPWM, pwmOutput);
  } else{
      pwmOutput = abs(pwmCalc);
      pwmOutput = map(pwmOutput, 0, 255, 255, 0); //Adjust duty cycle for Nidec motor
      digitalWrite(brake, HIGH);
      digitalWrite(dir, LOW);
      analogWriteFreq(freq);
      analogWrite(nidecPWM, pwmOutput);
  }
  i += 1;
}
```

Appendix B

Motor specification sheet



24H Brushless DC Motors

for Home Appliances and Office Equipment

- 3-Phase, 12-Pole Brushless Motors
- Logic-Controlled Clockwise or Counterclockwise Rotation
- Hall Effect Commutation
- Quiet Operation
- Low Inertia
- PWM Speed Control/Brake Function
- Open-Drain Tachometer
- Optional Dual Channel Phase-Tracking Encoder
- Locked Rotor Protection*
- Compact 42 (dia.) x 34mm Case

* Automatic shutdown at locked rotor condition: Restart at power OFF/ON.

Pinout

Pin	Function
1	Standard = No connection. Encoder option = Channel A output: 90° phase tracking, 100 pulses per revolution, HIGH = 5V, LOW = 0V.
2	Standard = Open-drain tachometer, six pulses per revolution, $I_{C(MAX)}$ = 3.0 mA. Encoder option = Channel B output: 90° phase tracking, 100 pulses per revolution, HIGH = 5V, LOW = 0V.
3	Standard = No connection. Encoder option = Logic supply, 5V ±0.5V.

Pin	Function
4	$V_{IN(HIGH)}$ = 2.0V to 5.0V or OPEN = Clockwise, $V_{IN(LOW)}$ <= 0.6V = Counterclockwise.
5	PWM: f_{IN} = 20 kHz to 30 kHz, $V_{IN(LOW)}$ <= 0.6V, $V_{IN(HIGH)}$ = 2.0V to 5.0V, duty cycle = 20% to 100%.
6	Brake: $V_{IN(HIGH)}$ = 2.0V to 5.0V = OFF, $V_{IN(LOW)}$ <= 0.6V = ON (motor stop).
7	Supply ground.
8	Motor supply voltage, 24V, nominal.

Nidec – All for dreams

Nidec America - phone 314-595-8186 - email motors@nidecamerica.com - www.nidecamerica.com

Appendix C

ESP8266 specification sheet

1. General Overview

1.1. Introduction

Espressif Systems' Smart Connectivity Platform (ESCP) is a set of high performance, high integration wireless SOCs, designed for space and power constrained mobile platform designers. It provides unsurpassed ability to embed WiFi capabilities within other systems, or to function as a standalone application, with the lowest cost, and minimal space requirement.

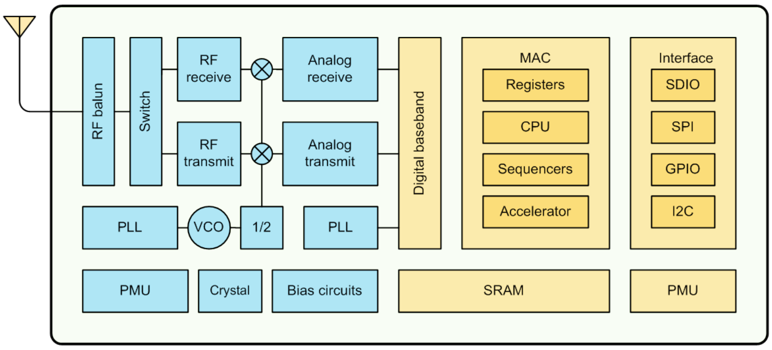


Figure 1 ESP8266EX Block Diagram

ESP8266EX offers a complete and self-contained WiFi networking solution; it can be used to host the application or to offload WiFi networking functions from another application processor.

When ESP8266EX hosts the application, it boots up directly from an external flash. In has integrated cache to improve the performance of the system in such applications.

Alternately, serving as a WiFi adapter, wireless internet access can be added to any micro controller-based design with simple connectivity (SPI/SDIO or I2C/UART interface).

ESP8266EX is among the most integrated WiFi chip in the industry; it integrates the antenna switches, RF balun, power amplifier, low noise receive amplifier, filters, power management modules, it requires minimal external circuitry, and the entire solution, including front-end module, is designed to occupy minimal PCB area.

ESP8266EX also integrates an enhanced version of Tensilica's L106 Diamond series 32-bit processor, with on-chip SRAM, besides the WiFi functionalities. ESP8266EX is often integrated with external sensors and other application specific devices through its GPIOs; sample codes for such applications are provided in the software development kit (SDK).

Espressif Systems' Smart Connectivity Platform (ESCP) demonstrates sophisticated system-level features include fast sleep/wake context switching for energy-efficient VoIP, adaptive radio biasing for low-power operation, advance signal processing, and spur cancellation and radio co-existence features for common cellular, Bluetooth, DDR, LVDS, LCD interference mitigation.

1.2. Features

- 802.11 b/g/n
- Integrated low power 32-bit MCU
- Integrated 10-bit ADC
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLL, regulators, and power management units
- Supports antenna diversity
- WiFi 2.4 GHz, support WPA/WPA2
- Support STA/AP/STA+AP operation modes
- Support Smart Link Function for both Android and iOS devices
- SDIO 2.0, (H) SPI, UART, I2C, I2S, IR Remote Control, PWM, GPIO
- STBC, 1x1 MIMO, 2x1 MIMO
- A-MPDU & A-MSDU aggregation & 0.4s guard interval
- Deep sleep power <10uA, Power down leakage current < 5uA
- Wake up and transmit packets in < 2ms
- Standby power consumption of < 1.0mW (DTIM3)
- +20 dBm output power in 802.11b mode
- Operating temperature range -40C ~ 125C
- FCC, CE, TELEC, WiFi Alliance, and SRRC certified

1.3. Parameters

Table 1 Parameters

Categories	Items	Values
WiFi Paramters	Certificates	FCC/CE/TELEC/SRRC
	WiFi Protocles	802.11 b/g/n
	Frequency Range	2.4G-2.5G (2400M-2483.5M)
	Tx Power	802.11 b: +20 dBm
		802.11 g: +17 dBm
		802.11 n: +14 dBm
	Rx Sensitivity	802.11 b: -91 dbm (11 Mbps)
		802.11 g: -75 dbm (54 Mbps)
		802.11 n: -72 dbm (MCS7)
	Types of Antenna	PCB Trace, External, IPEX Connector, Ceramic Chip
Hardware Paramaters	Peripheral Bus	UART/SDIO/SPI/I2C/I2S/IR Remote Control
		GPIO/PWM
	Operating Voltage	3.0~3.6V
	Operating Current	Average value: 80mA
	Operating Temperature Range	-40°~125°
	Ambient Temperature Range	Normal temperature
	Package Size	5x5mm
	External Interface	N/A
Software Parameters	WiFi mode	station/softAP/SoftAP+station
	Security	WPA/WPA2
	Encryption	WEP/TKIP/AES
	Firmware Upgrade	UART Download / OTA (via network)
	Ssoftware Development	Supports Cloud Server Development / SDK for custom firmware development
	Network Protocols	IPv4, TCP/UDP/HTTP/FTP

	User Configuration	AT Instruction Set, Cloud Server, Android/ iOS App

1.4. Ultra Low Power Technology

ESP8266EX has been designed for mobile, wearable electronics and Internet of Things applications with the aim of achieving the lowest power consumption with a combination of several proprietary techniques. The power saving architecture operates mainly in 3 modes: active mode, sleep mode and deep sleep mode.

By using advance power management techniques and logic to power-down functions not required and to control switching between sleep and active modes, ESP8266EX consumes about than 60uA in deep sleep mode (with RTC clock still running) and less than 1.0mA (DTIM=3) or less than 0.5mA (DTIM=10) to stay connected to the access point.

When in sleep mode, only the calibrated real-time clock and watchdog remains active. The real-time clock can be programmed to wake up the ESP8266EX at any required interval.

The ESP8266EX can be programmed to wake up when a specified condition is detected. This minimal wake-up time feature of the ESP8266EX can be utilized by mobile device SOCs, allowing them to remain in the low-power standby mode until WiFi is needed.

In order to satisfy the power demand of mobile and wearable electronics, ESP8266EX can be programmed to reduce the output power of the PA to fit various application profiles, by trading off range for power consumption.

1.5. Major Applications

Major fields of ESP8266EX applications to Internet-of-Things include:

- Home Appliances
- Home Automation
- Smart Plug and lights
- Mesh Network
- Industrial Wireless Control
- Baby Monitors
- IP Cameras
- Sensor Networks
- Wearable Electronics

- WiFi Location-aware Devices
- Security ID Tags
- WiFi Position System Beacons

TRITA - ITM - EX- 2023:87