CHALMERS
UNIVERSITY OF TECHNOLOGY

Balancing a Stationary Bicycle with a Reaction Wheel

Bachelor's thesis in Electrical Engineering

Adel Oubaid, Esmail Obed, Isak Rodung, Johanna Lindgren, Lukas Bloom

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026
www.chalmers.se

Bachelor's thesis 2026

Balancing a Stationary Bicycle With a Reaction Wheel

Adel Oubaid
Esmail Obed
Isak Rodung
Johanna Lindgren
Lukas Bloom

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2026

Balancing a Stationary Bicycle With a Reaction Wheel
Adel Oubaid, Esmail Obed, Isak Rodung, Johanna Lindgren, Lukas Bloom
Department of Electrical Engineering
Chalmers University of Technology

Supervisor: Ali Azarbahram, Qianlei Jia. Department of Electrical Engineering
Examiner: Jonas Sjöberg, Department of Electrical Engineering

Bachelor's Thesis 2026
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: The bicycle used in the project.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2026

Balancing a Stationary Bicycle With a Reaction Wheel
Adel Oubaid, Esmail Obed, Isak Rodung, Johanna Lindgren, Lukas Bloom
Department of Electrical Engineering
Chalmers University of Technology

Abstract

This thesis presents the design, modeling, and experimental validation of a stabilization system for a stationary bicycle using a reaction wheel actuator. The bicycle is modeled as an inverted pendulum, and a state-space representation is derived by combining mechanical roll dynamics with the electrical dynamics of a DC motor. The model serves as the foundation for controller design and simulation.

Control strategies are developed and evaluated in MATLAB, where system behavior is analyzed under varying initial conditions, disturbances, and time delays. The cascade controlled architecture was proven as the most optimal. Furthermore, the pole placement method is applied to obtain desired closed-loop dynamics, and the influence of key parameters on system stability is investigated.

The control algorithm is implemented on an embedded platform and tested on a physical prototype. Sensor data from an inertial measurement unit is processed using a complementary filter to estimate the roll angle in real time. Experimental results show that the system can generate stabilizing torque through the reaction wheel, but performance is limited by actuator response and system delays. Modifications, including compensation for back electromotive force, improve responsiveness and stability.

The results demonstrate the feasibility of stabilizing a stationary bicycle using a reaction wheel system, while highlighting practical limitations related to hardware constraints and control implementation.

Keywords: Stationary bicycle, reaction wheel, inverted pendulum, PID control, cascade control

Acknowledgements

We would like to thank our examiner Jonas Sjöberg, for the opportunity to be a part of the Autobike project. We also would like to thank our supervisors Ali Azarbahram & Qianlei Jia, for their constant support and guidance during the project.

Adel Oubaid, Esmail Obed, Isak Rodung, Johanna Lindgren, Lukas Bloom, Gothenburg, May 2026

List of Acronyms

ADAS Advanced Driver Assistance Systems. v, 1, 2
AEB Automatic Emergency Braking. v, 1

DC Direct Current. v, ix, 5, 7, 17

EMF Electromotive Force. v, 7

ICR Industrial Cellular Router. v, 22
IMU Inertial Measurement Unit. v, 12, 20, 21

LabVIEW Laboratory Virtual Instrument Engineering Workbench. v, 23

MyRIO My Reconfigurable Input Output. v, 21, 22, 26, 27

PD Proportional Derivative. v, 12, 32, 33, 42
PI Proportional Integral. v, 12, 32, 33, 42, 43
PID Proportional Integral Derivative. v, 9, 32, 33, 38, 39, 42

RMSE Root Mean Square Error. v, 39
RPM Revolutions Per Minute. v, 17, 21

VESC Vedder Electronic Speed Controller. v, 21, 37, 43
VRU Vulnerable Road Users. v, 1

Contents

List of Acronyms **v**

List of Figures **ix**

List of Tables **xi**

1 Background **1**
1.1 Context of Project . . . 1
1.2 Previous Work on the Autobike Project . . . 2
1.3 Problem Formulation and Purpose . . . 2
1.4 Delimitations . . . 3
1.4.1 Scope Delimitations . . . 3
1.4.2 System Assumptions . . . 3
1.5 AI-usage . . . 4

2 Theory **5**
2.1 Dynamics and System Modeling . . . 5
2.1.1 Inverted Pendulum . . . 5
2.1.2 Total Roll Dynamics of the System . . . 7
2.1.3 Electrical Motor . . . 7
2.1.4 State Space Model . . . 8
2.1.5 Rotational Stabilization Principles . . . 9
2.2 Control Theory . . . 9
2.2.1 PID Controller . . . 9
2.2.2 Feedback Control System . . . 10
2.2.3 Pole Placement Method . . . 11
2.2.4 Cascade Control . . . 11
2.2.5 Complementary Filter . . . 12

3 System Overview **14**
3.1 Hardware . . . 14
3.1.1 Hardware Table . . . 15
3.1.2 Bicycle . . . 15
3.1.3 Stabilization system . . . 16
3.1.3.1 Reaction Wheel . . . 17
3.1.3.2 Motor . . . 17
3.1.3.3 Motor holder and actuator mount . . . 18

3.1.3.4 Mounting . . . 19
3.1.4 Electronics and Control . . . 19
3.1.4.1 Battery . . . 19
3.1.4.2 Inertial Measurement Unit . . . 20
3.1.5 ESP32 LOLIN32 Lite . . . 20
3.1.5.1 VESC . . . 21
3.1.6 Discarded components . . . 22
3.1.6.1 MyRIO . . . 22
3.2 Software . . . 22
3.2.1 MATLAB and Simulink . . . 23
3.2.2 LabVIEW . . . 23
3.2.3 VESC-Tool . . . 23
3.2.4 Arduino IDE . . . 23

4 Method **24**
4.1 Simulating in MATLAB . . . 24
4.1.1 Pole Placement Method . . . 24
4.1.2 Non-zero Reference Value . . . 25
4.1.3 Nonlinear State Comparison . . . 25
4.1.4 Disturbances on a Stable System . . . 25
4.1.5 Time Delay . . . 25
4.1.6 Regulators and Cascade control . . . 26
4.2 Validation of Maximum Roll Angle . . . 26
4.3 Hardware Testing . . . 26
4.3.1 Controller Platform Selection . . . 26
4.3.2 IMU Test . . . 27
4.3.3 myRIO Test . . . 27
4.3.4 VESC With The Motor Test . . . 28
4.3.5 Actuator Mount Design and Fabrication . . . 28
4.4 Physical Bicycle Test . . . 30
4.4.1 System Hardware Integration . . . 30
4.4.2 Control System Implementation . . . 30
4.4.3 Experimental Testing and Initial System Performance . . . 30
4.4.4 Motor Current Correction Using Back EMF . . . 31
4.4.5 PD Controller Gain Tuning . . . 31

5 Results **32**
5.1 Maximum Recoverable Angle Calculation . . . 32
5.2 Comparative Analysis of Control Systems in MATLAB . . . 33
5.2.1 PI Controller Evaluation . . . 33
5.2.2 PD Controller Evaluation . . . 34
5.2.3 PID Controller Evaluation . . . 35
5.2.4 Cascade Control Evaluation . . . 36
5.3 MATLAB Parameter Testing . . . 38
5.3.1 Non-zero Reference Value . . . 38
5.3.2 Nonlinear State Comparison . . . 39
5.3.3 Disturbance on a Stable System . . . 39

5.3.4 Time Delay 40
5.4 Evaluation of Best Controller 42
5.5 Physical Test 43
5.5.1 IMU 43
5.5.2 myRIO 44
5.5.3 Effect of Back-EMF Compensation on Reaction-Wheel Response 45
5.5.4 Experimental Evaluation of PD Controller Tuning 47
5.5.4.1 System Limitations and Proposed Improvements . . 48
5.5.4.2 Comparison Between Simulation and Hardware . . . 51

6 Conclusion **52**

List of Figures

2.1 Inverted pendulum shown with relevant variables for the project. . . . 6
2.2 Electrical model of a Direct Current (DC) motor. 7
2.3 Block diagram of a feedback control system. 11
2.4 Cascade block diagram for the system. 12

3.1 Bicycle platform with the control box and reaction-wheel actuator indicated. 14
3.2 Unequipped Bicycle Platform 16
3.3 Approximate actuator placement on the bicycle frame, showing the intended lokation between the top tube, seat tube, and down tube. . 16
3.4 Reaction wheel weights. 17
3.5 Motor 17
3.6 Motor holder SolidWorks model 18
3.7 Actuator mount Catia model 18
3.8 Turnigy battery 14000 mAh 19
3.9 IMU Pmod NAV 20
3.10 Lolin32 lite with esp32 21
3.11 VESC 6 MK 21
3.12 myRIO controller 22

4.1 Testing myRIO controller throught LabVIEW CODE 28
4.2 Prototype pieces to evaluate fit 29
4.3 Finished Grey Actuator Mount 29

5.1 PI controller simulation with an initial angle of $\theta = 0.00005$ radians. . 34
5.2 PD controller simulation with an initial angle of $\theta = 0.041$ radians. . 35
5.3 PID controller simulation with an initial angle of $\theta = 0.039$ radians. . 36
5.4 Cascade controller simulation with a maximum initial angle of $\theta =$ 0.04 radians. 37
5.5 PID controller with a non-zero reference value. 38
5.6 PID controller comparison between a nonlinear system and a linear system. 39
5.7 Cascade controller response to a 4 Nm external torque disturbance applied to the roll angle for 0.1 seconds. 40
5.8 PD controller with a time delay of 50 milliseconds. 41
5.9 Response of the roll angle evaluated across varying sensor time delays. 42
5.10 IMU Readings 44

5.11 Coordinate Axes 44
5.12 myRIO accelerometer measurements along the three axes 45
5.13 Experimental response of the reaction-wheel system without back-EMF current compensation over time. 47
5.14 Experimental response of the reaction-wheel system with back-EMF current compensation over time. 47
5.15 Experimental response of the PD controller to the bicycle roll angle, with $K_p = 200$ and $K_d = 25$. 48
5.16 PD controller current response. 50
5.17 Melted motor attachment. 51

List of Tables

3.1 Mechanical components and specifications. 15
3.2 Electrical/control components and specifications. 15

5.1 Values for parameters that were used in the calculation and simulations. 33
5.2 Effects of varying time delays on PD controller stability. 41
5.3 Objective performance comparison of evaluated control systems. . . . 42

1
Background

This chapter establishes the foundational context for the Autobike project. It outlines the motivation behind developing an autonomous bicycle for automotive safety testing and reviews previous iterations of the system. Finally, it defines the specific objectives of this current research and establishes clear delimitations to define the scope and boundaries of the project.

1.1 Context of Project

Urban transportation systems are undergoing rapid transformation due to increased urbanization and a growing demand for sustainable mobility solutions. Cycling has become an increasingly important mode of transport in urban environments, contributing to reduced emissions and congestion. However, cyclists are classified as Vulnerable Road Users (VRU), a term used within European transport policy to describe non-motorized road users, such as pedestrians and cyclists, as well as other road users lacking physical protection. As a result, cyclists remain disproportionately exposed to traffic-related injuries [1].

In parallel, the automotive industry has, in recent years, intensified the development of Advanced Driver Assistance Systems (ADAS) incorporating autonomous sensing technologies aimed at improving traffic safety. Volvo Cars introduced Automatic Emergency Braking (AEB) capable of detecting pedestrians and cyclists as early as 2013 [2]. Reliable evaluation of such systems requires realistic and reproducible test scenarios involving stationary and dynamic obstacles. The obstacles should simulate different car and bicycle encounters.

The Autobike project originated from the realization that there is potential in developing an autonomous alternative to the non-autonomous obstacles used in crash tests today. Current test setups often rely on specialized mechanical rigs designed for individual scenarios, limiting flexibility and increasing costs due to the need for multiple dedicated test systems [3, 4]. An autonomous self-balancing bicycle instead enables more adaptable and realistic test scenarios, where both stationary and dynamic cyclist behavior can be reproduced within the same platform.

1.2 Previous Work on the Autobike Project

Early work within the Autobike project focused primarily on stabilizing the bicycle through forward motion, where a minimum velocity enabled trajectory tracking and self-balancing behavior [5, 6]. During this period, system development emphasized path planning, steering-based stabilization, and sensor fusion using complementary filtering.

Later work expanded the system modeling and state estimation capabilities by introducing more advanced estimation methods, including Kalman filtering [7]. These developments improved the accuracy of state estimation and enabled more robust control-oriented modeling of the bicycle system.

More recent work shifted toward stationary stabilization using reaction-wheel-based balancing methods [8]. Instead of relying on forward motion, stabilization torque could be generated directly through angular momentum transfer from a rotating mass, enabling balancing at standstill.

1.3 Problem Formulation and Purpose

The automotive industry increasingly relies on ADAS to protect vulnerable road users, such as cyclists. Rigorously evaluating these safety systems requires realistic and reproducible test scenarios. However, current crash test targets are often relatively simple, and previous iterations of the autonomous Autobike were limited by their dependence on forward motion to maintain balance. This lack of stationary stability restricts the system's ability to accurately simulate authentic, low-speed, or standstill traffic encounters.

Therefore, the purpose of this project is to address this limitation by further developing and evaluating a reaction wheel-based control system. By enabling the Autobike to maintain upright stability from a completely stationary state without external support, this project aims to significantly increase the flexibility, realism, and applicability of the testing setup for future automotive safety evaluations.

The specific objectives are to:

- derive a control-oriented roll model of the stationary bicycle,
- design and compare candidate controllers in simulation,
- estimate the maximum recoverable roll angle,
- implement the sensing and actuation chain on hardware, and
- evaluate the limitations of the physical prototype.

1.4 Delimitations

To ensure the project remains feasible within the set timeframe and to manage the complexity of the system design, specific boundaries and simplifications have been established. These are divided into scope delimitations and system assumptions.

1.4.1 Scope Delimitations

These points define the physical and operational boundaries of the project:

- **Stationary Operation Only:** The stabilization of the bicycle will exclusively be evaluated from a standstill. Forward or directional movement is not covered in this phase of the Autobike project.
- **Horizontal Ground:** The physical prototype will only be evaluated on a flat, horizontal surface, and stabilization on tilted or uneven ground will not be physically tested. However, non-zero reference tracking is included solely as an exploratory simulation to observe theoretical system behavior under a continuous gravitational disturbance.
- **No Test Dummy**: The current prototype and control system evaluation will not include the added mass or unpredictable dynamics of a physical test dummy.

1.4.2 System Assumptions

To derive the state-space model and design the control system, several physical simplifications were necessary. These assumptions are justified below:

- **Rigid Body Dynamics:** The bicycle is assumed to have no flexible parts and is modeled as a single rigid body.
  - *Justification:* Modeling frame flexibility, tire deformation, and component vibration would require a higher-order nonlinear model. A rigid body assumption is standard for inverted pendulum control problems and keeps the state-space model computable in real-time by the microcontroller without introducing severe calculation delays.
- **No Wheel Slipping:** It is assumed that there is no slipping between the contact points of the bicycle wheels and the ground.
  - *Justification:* This ensures that the pivot point of the system remains perfectly fixed. If the wheel slipped, the basic geometric relationships used to calculate the maximum recoverable angle and the necessary restorative torque would break down.
- **Negligible Temperature Losses:** Thermal losses from mechanical or electrical components (such as motor heating) are not included in the model.
  - *Justification:* The project focuses on short-term transient stability and control responses. Thermal dynamics occur on a significantly slower timescale and will not substantially affect the immediate balancing capabilities during testing.

- **Linearized Friction Model:** In the mathematical derivations and subsequent simulations, non-linear frictional effects such as static friction and Coulomb friction, as well as aerodynamic drag, are neglected. Viscous friction within the reaction wheel motor is, however, included in the system dynamics.
  - *Justification:* While the physical prototype naturally experiences all forms of friction during hardware testing, incorporating non-linear friction into the theoretical model would significantly complicate the derivation of a standard linear state-space representation. Assuming purely viscous friction allows for standard linear control techniques to be applied while still capturing the primary dampening effect.

1.5 AI-usage

During the preparation of this report, AI tools (specifically ChatGPT and Google Gemini), were used to assist with the writing and development process. The use of these tools was limited to the following areas:

- **Text Refinement:** Identifying spelling or grammatical errors and suggesting vocabulary enhancements to improve readability.
- **Code Comprehension:** Assisting in the understanding and interpretation of MATLAB scripts and functions.
- **To create images or improve images clarity and resolution.**

2
Theory

This chapter presents the theoretical background relevant to the stabilization of the bicycle. The chapter is divided into two main domains: system dynamics and control theory. First, the physical behavior is described by approximating the bicycle as an inverted pendulum. By integrating the mechanical equations of motion with the electrical dynamics of the DC motor, a comprehensive state-space representation is derived to serve as the basis for the controller design. This section also explores the physical principles of rotational stabilization using reaction wheels. Second, the chapter introduces the core control-theory concepts applied in this project.

2.1 Dynamics and System Modeling

This section presents the dynamic modeling of the bicycle and the associated actuator system. The physical behavior of the system is described using fundamental principles of mechanics and electrical dynamics, forming a mathematical representation suitable for control design. The bicycle is modeled as an inverted pendulum, where the roll dynamics are influenced by gravitational torque and the control torque generated by the reaction wheel. In addition, the electrical dynamics of the motor is included to capture the relationship between input voltage and generated torque. These models are combined into a state-space representation, which serves as the foundation for controller design and analysis in later sections. Lastly, the rotational stabilization principles are presented.

2.1.1 Inverted Pendulum

The bicycle can be approximated as an inverted pendulum, since the combined mass of the bicycle and the hardware is above the ground contact point [9]. An inverted pendulum consists of a rigid body with its center of mass positioned above its pivot point, making the upright position inherently unstable [10]. For the purpose of this project, the bicycle is stationary, and the analysis is restricted to small deviations around the upright equilibrium. Consequently, translational motion, steering effects, and higher-order dynamic terms are neglected. Therefore, Figure 2.1 is defined with a positive roll angle to the left of the vertical axis and a negative to the right.

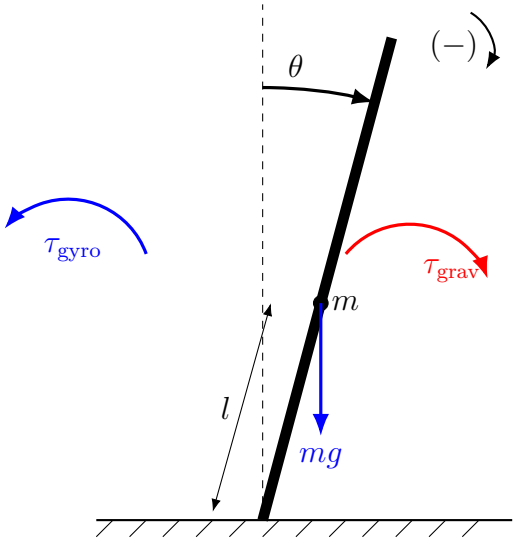


Figure 2.1: Inverted pendulum shown with relevant variables for the project.

Under these assumptions, the model is reduced to its dominant unforced roll dynamics, which can be described using Newton's second law as:

$$I\ddot{\theta} = \tau_{\text{grav}} \tag{2.1}$$

The gravitational torque is given by:

$$\tau_{\text{grav}} = mgl\sin(\theta) \tag{2.2}$$

where:

- m is the total mass of the system.
- g is the gravitational acceleration.
- l is the distance from the wheel-ground contact point to the center of mass.
- θ is the vertical roll angle of the bicycle.
- I is the moment of inertia of the system with respect to the ground contact point.
- $\ddot{\theta}$ is the angular acceleration (the second time derivative of the roll angle).

Note that the stabilizing control torque, τ_{gyro}, shown in Figure 2.1, is excluded from this initial unforced pendulum equation but will be introduced in the following section to complete the system dynamics.

For small roll angles, the approximation $\sin(\theta) \approx \theta$ can be applied, resulting in a linearized model:

$$I\ddot{\theta} = mgl\,\theta \tag{2.3}$$

This simplified expression forms the basis for the control-oriented model used in this project.

2.1.2 Total Roll Dynamics of the System

The total roll dynamics are obtained by including the control torque generated by the real reaction wheel. The equation of motion can therefore be written as:

$$I\ddot{\theta} = \tau_{\text{grav}} - \tau_{\text{gyro}} \tag{2.4}$$

where τ_{gyro} is the torque delivered from the reaction wheel and can be described as

$$\tau_{gyro} = I_f\dot{\omega} - b\omega. \tag{2.5}$$

where:

- I_f is the moment of inertia from the flywheel.
- $\dot{\omega}$ is the angular acceleration of the flywheel.
- $b\omega$ is the friction torque that is created when the motor is rotating.

Substituting the expressions for gravitational and control torque yields:

$$I\ddot{\theta} = mgl\sin(\theta) - I_f\dot{\omega} + b\omega \tag{2.6}$$

Applying the small-angle approximation $\sin(\theta) \approx \theta$ gives the linearized model:

$$I\ddot{\theta} \approx mgl\,\theta - I_f\dot{\omega} + b\omega \tag{2.7}$$

2.1.3 Electrical Motor

To describe the electrical dynamics of the DC motor, Kirchhoff's voltage law [11] is applied to the equivalent circuit. Figure 2.2 illustrates this standard electrical model, consisting of an internal armature resistance, an armature inductance, and an Electromotive Force (EMF) acting against the applied voltage.

Applying Kirchhoff's voltage law to the circuit gives:

$$V_{\text{in}} - V_{R_a} - V_{L_a} - V_{\text{emf}} = 0 \tag{2.8}$$

where V_{in} is the input voltage applied to the motor, V_{R_a} is the voltage drop across the armature resistance, V_{L_a} is the voltage drop across the armature inductance and V_{emf} is the induced back-EMF.

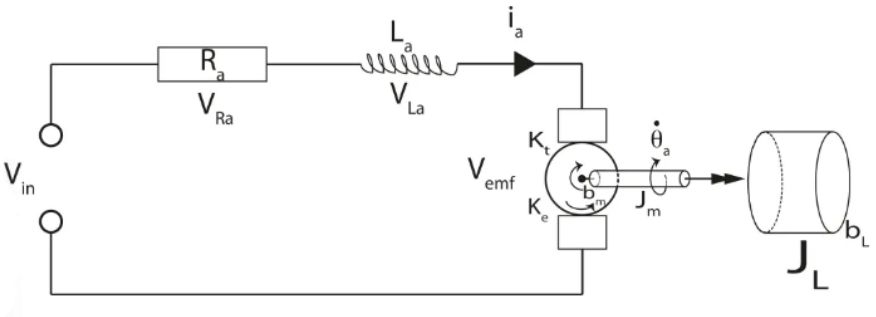


Figure 2.2: Electrical model of a DC motor.

These individual voltage components can be expressed in terms of the motor parameters and state variables:

$$V_{\text{in}} = u \tag{2.9}$$

$$V_{R_a} = R_a i_a \tag{2.10}$$

$$V_{L_a} = L_a \frac{di_a}{dt} \tag{2.11}$$

$$V_{\text{emf}} = K_e \omega_a \tag{2.12}$$

where:

- u is the applied voltage to the armature circuit.
- R_a is the armature resistance of the motor.
- L_a is the armature inductance of the motor.
- i_a is the armature current of the motor.
- K_e is the motor back-EMF constant.
- ω_a is the angular velocity of the motor.

Substituting Equation 2.9 through Equation 2.12 into Equation 2.8 gives the complete differential equation for the electrical dynamics of the motor [12]:

$$-u + R_a i_a + L_a \frac{di_a}{dt} + K_e \omega = 0 \tag{2.13}$$

The mechanical torque generated by the motor, which serves as the stabilizing control torque for the bicycle, is directly proportional to the armature current. This relationship is described as:

$$\tau_{\text{gyro}} = I_f \dot{\omega} = K_m i_a \tag{2.14}$$

where K_m is the motor torque constant, and I_f and $\dot{\omega}$ are the inertia and angular acceleration of the reaction wheel, respectively.

2.1.4 State Space Model

By combining the mechanical and electrical dynamics, the system can be described by the following equations [13]:

$$I\ddot{\theta} \approx mgl\,\theta - I_f \dot{\omega} + b\omega \tag{2.15}$$

$$\tau_{gyro} = I_f\,\dot{\omega} = K_m i_a \tag{2.16}$$

$$-u + R_a i_a + L_a \frac{di_a}{dt} + K_e \omega = 0 \tag{2.17}$$

These equations are used to derive a state-space representation of the system [13].

The input $u(t)$ represents the applied motor voltage, while the output corresponds to the roll angle of the bicycle. The state vector is defined as:

$$\mathbf{x}(t) = \begin{bmatrix} \theta \\ \dot{\theta} \\ \omega \\ i_a \end{bmatrix} \tag{2.18}$$

The system can then be written in standard state-space form [13, 14] as:

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{mgl}{I} & 0 & \frac{b}{I} & -\frac{K_m}{I} \\ 0 & 0 & -\frac{b}{I_f} & \frac{K_m}{I_f} \\ 0 & 0 & -\frac{K_e}{L_a} & -\frac{R_a}{L_a} \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{L_a} \end{bmatrix} u(t) \tag{2.19}$$

$$\mathbf{y} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \mathbf{x}(t) \tag{2.20}$$

2.1.5 Rotational Stabilization Principles

Stabilization using rotating masses is fundamentally based on the conservation of angular momentum [11]. A rotating body possesses angular momentum, and any change in its magnitude or direction requires an external torque. Conversely, actively changing a system's angular momentum generates a reaction torque. This principle can be utilized to counteract the tipping moment of the bicycle.

To practically apply the physics and generate the desired counter-torque, two primary methods are commonly employed: reaction wheel systems and gyroscopic stabilizers. In a reaction wheel system, stabilizing torque is produced by accelerating or decelerating the wheel, which alters the magnitude of its angular momentum [15]. In contrast, gyroscopic stabilization relies on a mass rotating at a continuous speed, where torque is instead generated by tilting the rotation axis to change the direction of the angular momentum vector [16]. Although their mechanical implementations differ, both approaches exploit the exact same underlying physics to actively maintain the system's equilibrium [11]. For the purpose of this project, a reaction wheel system was selected as the method for actively maintaining the bicycle's equilibrium.

2.2 Control Theory

This section introduces the fundamentals of control theory, focusing on the use of feedback to regulate dynamic systems. It describes how control methods are applied to stabilize the bicycle and presents the complementary filter used for state estimation.

2.2.1 PID Controller

A Proportional Integral Derivative (PID) controller is a widely used control strategy that can be used for many different systems. Thus the strategy is based on three

components: proportional, integrated, and derivative action.

$$u(k) = K_p e(k) + K_i \sum_{i=0}^{k} e(i) T_s + K_d \frac{e(k) - e(k-1)}{T_s} \tag{2.21}$$

where:

- $u(k)$ is controller output at sample k.
- $e(k)$ is error at sample k.
- K_p is proportional gain.
- K_i is integral gain.
- K_d is derivative gain.
- T_s is sampling time.

The proportional part reads the current error and reacts proportionally, which increases the system's responsiveness. The integral part accumulates a sum of past errors and makes the system capable of eliminating small steady-state offsets. The derivative part is mostly affected by the rate of change of the error and reacts against it by damping it. This part improves the stability of the system by preventing the oscillation which can occur.

The gains K_p, K_i and K_d are scalar parameters that determine the contribution of the proportional, integral, and derivative parts in the controller. The controller can thus be adapted to achieve the desired performance. Setting individual parameters in this equation 2.21 to zero gives simplified controllers [17, 18].

2.2.2 Feedback Control System

A feedback control system is used to regulate dynamic systems by continuously comparing the measured output with a desired reference value [13]. The difference between these signals defines the control error:

$$e(k) = r(k) - y(k) \tag{2.22}$$

Where:

- $e(k)$ is error at sample k.
- $r(k)$ is the reference value.
- $y(k)$ is the measured output.

The error signal is used by the controller to adjust the system input in order to reduce the deviation over time. In a closed-loop system, see Figure 2.3, the output is continuously fed back to the controller by a sensor, enabling real-time correcting of disturbances and model uncertainties.

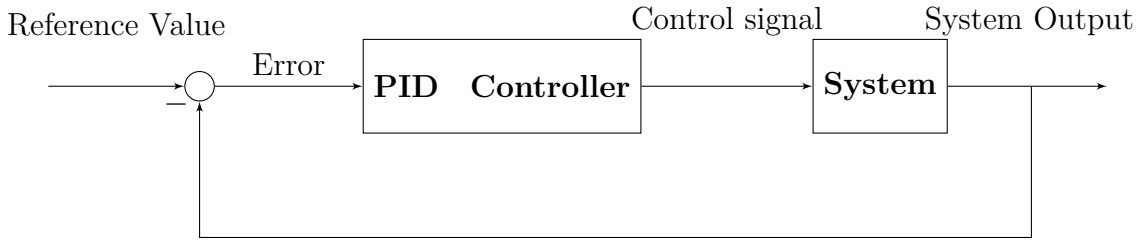


Figure 2.3: Block diagram of a feedback control system.

In this project, the reference value is the upright position of the bicycle, corresponding to a roll angle och zero degrees. The roll angle is measured using the IMU 3.1.4.2 and fed back to the controller. The control error is processed a PID controller to convert an angle error to an increased signal. The resulting signal is sent to the VESC 3.1.5.1, which drives the motor with the calculated power needed to counteract the rolling motion.

2.2.3 Pole Placement Method

The pole placement method is a control design technique used to shape the dynamic behavior of a system by assigning the closed-loop poles to desired locations in the complex plane [13].
The location of the poles determines key system proporties such as responsiveness, stability, and damping. Poles with negative real parts ensure stability, while their distance from the imaginary axis affects how fast the system responds. Complex conjugate poles introduce oscillatory behavior, where the real part determines damping and the imaginary part determines the oscillation frequency.
In general, the desired closed-loop dynamics are defined by a characteristic polynomial:

$$P(s) = (s - p_1)(s - p_2) \cdots (s - p_n) \tag{2.23}$$

By matching this desired polynomial with the characteristic polynomial of the system, controller parameters can be determined such that the system exhibits the specified behavior.

2.2.4 Cascade Control

Cascade control is a control strategy used in systems with multiple dynamic states, particularly when these states operate on different time scales [19]. It improves control performance by separating the regulation of fast and slow dynamics.

A cascade control system consists of two nested feedback loops: an inner loop and an outer loop. The inner loop regulates fast dynamics of the system, while the outer loop controls the main (slower) process [20]. Each loop has its own controller and associated process.

As illustrated in Figure 2.4, the outer controller (Proportional Derivative (PD)) regulates the primary bicycle roll dynamics (G_2), while the inner controller (Proportional Integral (PI)) regulates the faster, electrical dynamic system (G_1). The output of the outer controller serves as the reference input to the inner loop, and the output of the inner loop drives the main process [21].

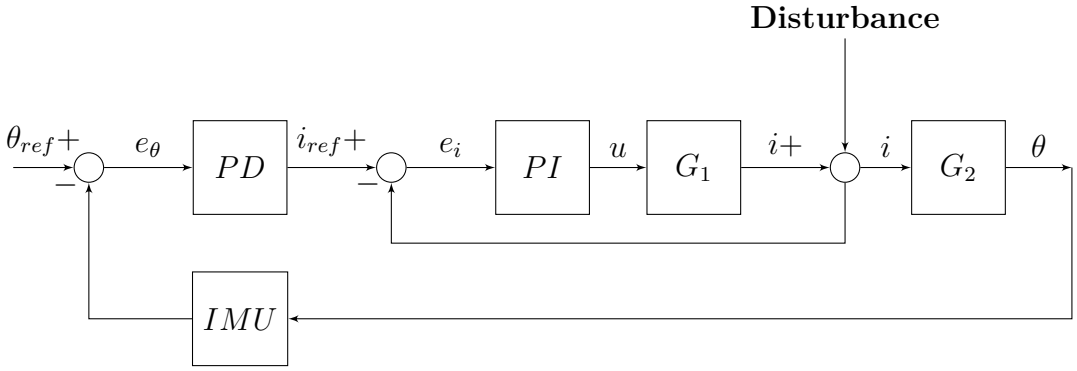


Figure 2.4: Cascade block diagram for the system.

The benefit of using cascade control is the that the disturbances affecting the fast dynamics can be corrected by the inner loop before they propagate to the main system. This improves stability and performance since it reduces the impact of disturbances on the overall system [19].

In this project, cascade control was considered as a potential control strategy. However, alternative methods were ultimately selected, as discussed in later chapters.

2.2.5 Complementary Filter

A complementary filter is used to estimate the roll angle by combining measurements from different sensors in the Inertial Measurement Unit (IMU) with complementary frequency characteristics. It fuses low-frequency information from the accelerometer with high-frequency information from the reaction wheel, resulting in a more robust angle estimate [22].

The accelerometer provides reliable long-term orientation information, but is sensitive to noise and vibrations. In contrast, the reaction wheel provides accurate short-term angular velocity measurements but suffers from drift over time. By combining these signals, the complementary filter compensates for the weakness of each sensor.

The angle estimation is given by:

$$\theta = H \cdot (\theta + \omega \cdot dt) + L \cdot (x_{acc}) \tag{2.24}$$

Where:

- θ is the estimated angle.

- ω is the angular velocity estimated from the reaction wheel.
- dt is the sampling time.
- x_{acc} represents the angle estimated from the accelerometer.

The first term represents the high-pass filtered reaction wheel contribution, while the second term corresponds to the low-pass filtered accelerometer signal. The constraints H and L are weighted factors satisfying with $H + L = 1$. Typical values are $H \approx 0.98$ and $L \approx 0.02$, giving more weight to the reaction wheel for fast dynamics while allowing the accelerometer to correct long-term drift. Adjusting these parameters changes the responsiveness and noise sensitivity of the filter.

3

System Overview

This chapter describes the hardware and software components used in the project. The purpose of each component is presented, and the overal system architecture is introduced. Figure 3.1 illustrates the finished system setup.

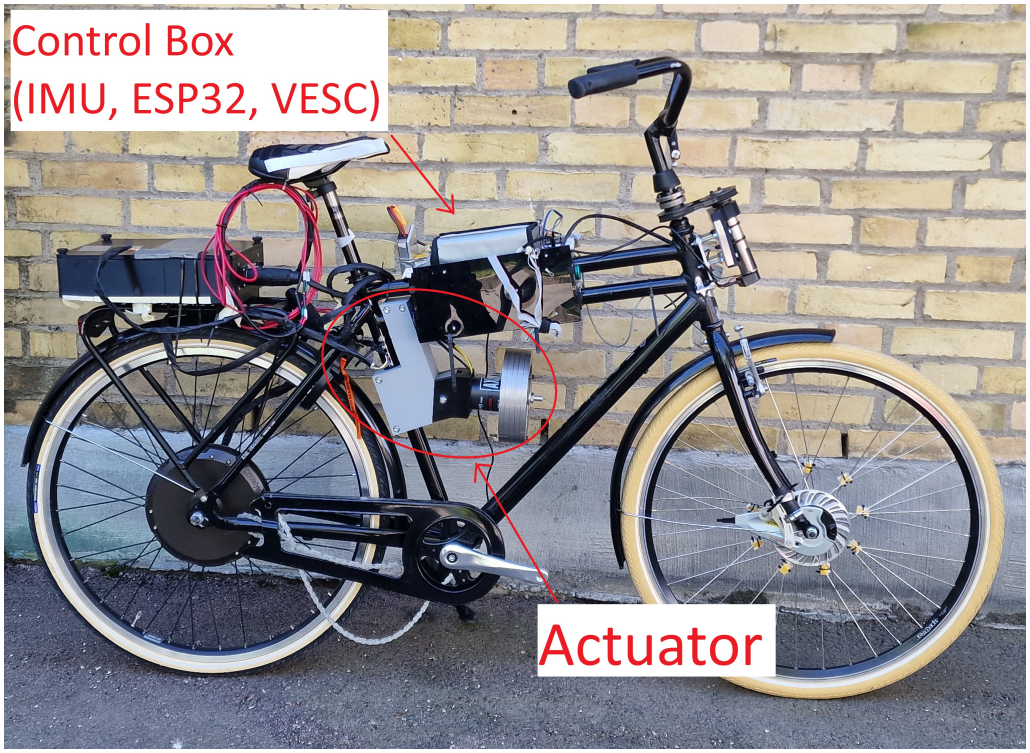


Figure 3.1: Bicycle platform with the control box and reaction-wheel actuator indicated.

3.1 Hardware

The hardware consists of the physical components that enable the detection, control, and actuation of the bicycle and support the implementation of the stabilization system.

3.1.1 Hardware Table

The main hardware components used in the system are summarized in two tables below. Mechanical components make up the system's structure and are listed with mass and dimension information in Table 3.1. The electrical components enable connecting hardware and software and are listed with their capacities in Table 3.2.

Table 3.1: Mechanical components and specifications.

Component	Model	Function	Key Specifications
Bicycle	Monark Karl	Base platform	23 kg (with components except actuator)
Reaction wheel	Custom	Generates angular momentum	6 kg, 18 cm diameter
Actuator mount	3D-printed	Grips the motor holder	–
Motor holder	3D-printed	Holds the motor	–

Table 3.2: Electrical/control components and specifications.

Component	Model	Function	Key Specifications
Motor	AXI 5360/24HD	Drives reaction wheel	3120 W, 65 A, 95 RPM/V, 1.270 kg
Batteries	Turnigy 14.0	Power supply	14000 mAh, 22.2 V, 6 CELL, 2.4 kg
IMU	Pmod NAV	Motion sensing	Angular velocity, acceleration
Microcontroller	ESP32 LOLIN32 Lite	Control unit	ESP32 chip
VESC	VESC 6 MK VI	Motor controller	11–60 V, 80 A continuous

3.1.2 Bicycle

The bicycle used in this project is a seven-geared Monark Karl model, which is shown below 3.2. However, the handle bar and seat have been replaced, and the geras dismantled. The bicycle serves as the base platform onto which the hardware components listed in tables 3.1 and 3.2 are mounted. The total mass of the bicycle, including the mounted hardware but excluding the reaction wheel actuator, is approximately 23 kg. This platform represents an iteration within the AutoBike project among other bicycles and a scooter, building upon previous system designs.

Figure 3.2: Unequipped Bicycle Platform

3.1.3 Stabilization system

The stabilization system uses a motor-driven reaction wheel actuator to generate control torque. The actuator is mounted on the bicycle frame, positioned between the top tube, seat tube, and down tube.

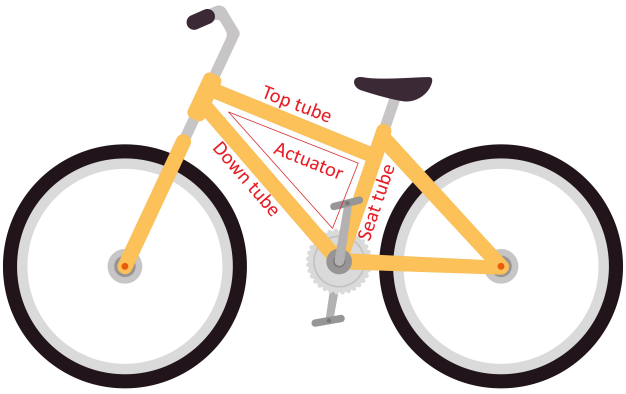


Figure 3.3: Approximate actuator placement on the bicycle frame, showing the intended lokation between the top tube, seat tube, and down tube.

3.1.3.1 Reaction Wheel

The reaction wheel used in this project is made of five discrete masses assembled using bolts and nuts, forming a rotating body. One of the weights has a mass of 1250 g, while the remaining four each have a mass of 950 g (Figure 3.4). All together with bolts and nuts they weigh around 5 kg. The masses have a diameter of 18 cm and the moment of inertia increases with the square of the radius, according to the formula of moment of inertia 3.1.

$$I_f = \sum m_i r_i^2 \tag{3.1}$$

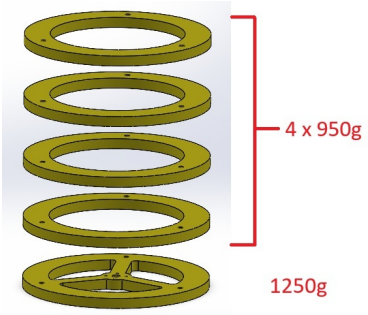


Figure 3.4: Reaction wheel weights.

3.1.3.2 Motor

A motor of the model AXI 5360/24HD V3 GOLD LINE is used in the system to rotate the reaction wheel. The model is a brushless 14 pole out runner DC motor and has an RPM/V of 95 with a maximum power of 3120 W [23]. This DC motor has a maximum current of 65 A and can tolerate this current for up to 20 seconds, which gives us a maximum voltage of 48 V. The maximum efficiency($> 85\%$) achieved for the motor is between 20-59 amperes. The inductance of the motor is assumed to be 1mH as a reference.

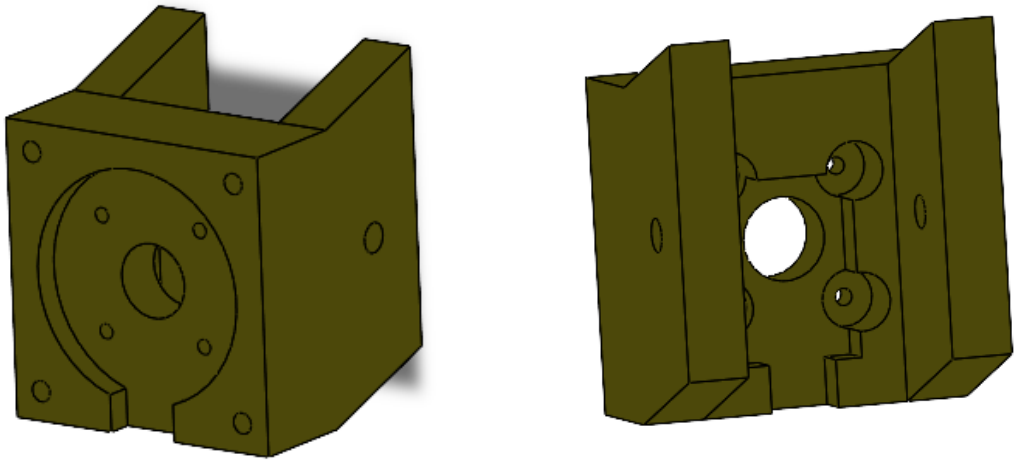
This model of motor was designed for scale planes up to fifteen kilograms with low RPM and a large torque. The motor is equipped with three ball bearings on the shaft and an additional bearing supporting the rotating outer case.

Figure 3.5: Motor

3.1.3.3 Motor holder and actuator mount

The motor is connected to the actuator mount using a pre-existing 3D-printed motor holder (Figure 3.6).

The actuator mount is a custom 3D-printed part developed in this project that connects the motor holder to the bicycle frame. The design ensures proper alignment and structural stability during operation (Figure 3.7).



(a) Motor attachment

(b) Actuator mount attachment

Figure 3.6: Motor holder SolidWorks model

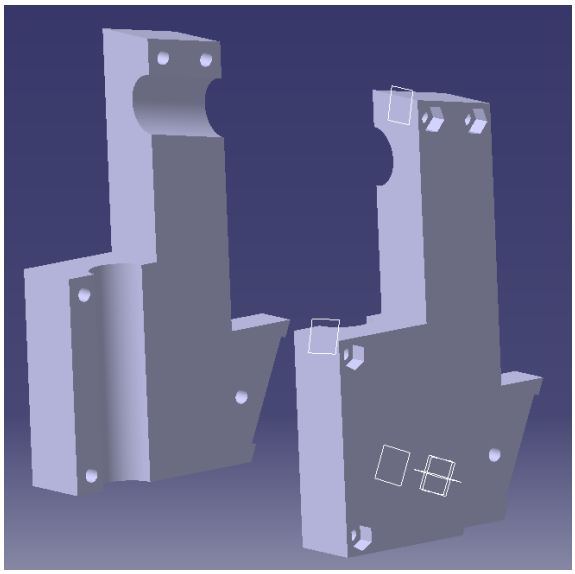


Figure 3.7: Actuator mount Catia model

3.1.3.4 Mounting

Bolts and nuts are used to assemble the reaction wheel and secure the motor mount to the bicycle frame. The mounting ensures that the actuator is rigidly attached to the frame, minimizing unwanted vibrations and ensuring stable transmission of torque to the structure.

3.1.4 Electronics and Control

This section describes the electronic and control components used in the system. These components provide power, sensing, and signal processing, and act as the interface between the software and hardware.

3.1.4.1 Battery

The battery employed is a TURNIGY 14,000 mAh Lithium Polymer (LiPo) pack, see Figure 3.8, serving as a portable, high-capacity power source for the hardware system. It consists of a 6-cell (6S) arrangement with a nominal voltage of 22.2 V and an energy capacity of about 310.8 Wh. With a discharge rating of 12–24C, it is well-suited for use in applications that demand stable, high-current output, including embedded hardware platforms. The pack is equipped with a primary XT-style power connector and a balance connector to enable safe charging and cell monitoring.

Figure 3.8: Turnigy battery 14000 mAh

3.1.4.2 Inertial Measurement Unit

The bicycle is equipped with an IMU of the model Pmod NAV, see Figure 3.9, developed by Digilent [24]. The unit includes an accelerometer, gyroscope, and magnetometer for each Cartesian axis, as well as a barometer of the model LPS25HB [25]. The accelerometer measures linear acceleration in m/s^2, while the gyroscope provides angular velocity in rad/s [26]. The magnetometers and barometer are not of use for this project, as they do not contribute to the stabilization of the bicycle.

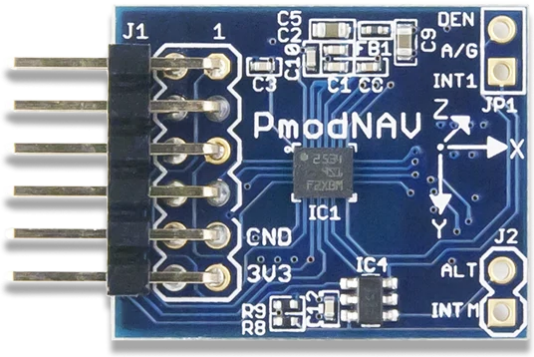


Figure 3.9: IMU Pmod NAV

3.1.5 ESP32 LOLIN32 Lite

Figure 3.10 displays the LOLIN32 Lite, which is a compact development board based on the ESP32 microcontroller, which integrates processing capabilities and wireless communication in a single unit. In this project, it serves as the main control unit responsible for data acquisition, control computation and communication with the motor controller.

Sensor data from IMU is read by the ESP32 and used to estimate the roll angle and angular velocity of the bicycle. Based on this information, a PID controller is implemented on the microcontroller to compute the control signal required to counteract deviations from the upright position. The resulting control signal is then transmitted to the VESC motor controller, which actuates the motor to generate the required stabilizing torque. Compared to a distributed system with separate control and communication units, the ESP32 enables a more integrated approach where sensing, control, and communication are handled within a single embedded platform. This reduces system complexity and allows for efficient real-time operation.

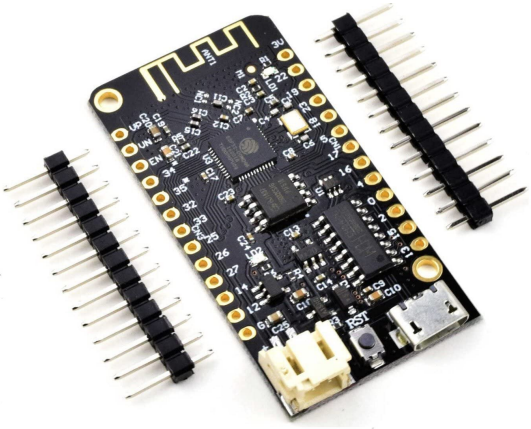


Figure 3.10: Lolin32 lite with esp32

3.1.5.1 VESC

The Vedder Electronic Speed Controller (VESC) 6 MK seen in Figure 3.11, is a motor controller used to regulate the brushless motor driving the reaction wheel. It controls the motor by adjusting current, voltage and rotational speed based on input signals from the control system. The controller supports configurable limits for current, voltage, power, and RPM, allowing safe and flexible operation [27]. It operates within a voltage range of 11.1 V and 60 V and can handle peak currents up to 210 A, with a recommended continuous current of 80 A and a maximum motor current of 160 A.

An internal IMU is integrated in VESC hardware, however, it is not utilized in this project, as external sensors are used for state estimation.

The VESC acts as the interface between the control unit (My Reconfigurable Input Output (MyRIO)) and the motor, executing control commands by converting them into appropriate electrical signals.

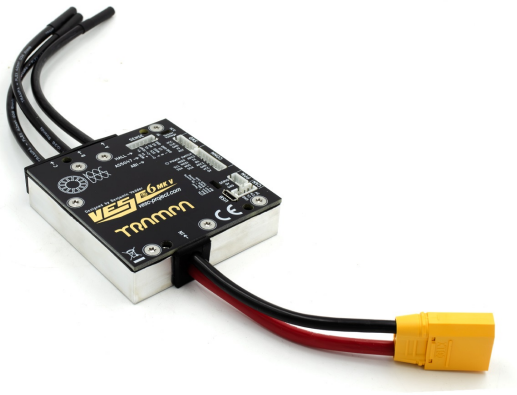


Figure 3.11: VESC 6 MK

3.1.6 Discarded components

Some components have been part of the project but have later been replaced with alternative, more convenient solutions. They are still alternative solutions that might be used and maybe more preferable in other different conditions.

3.1.6.1 MyRIO

The NI MyRIO 1900 (National Instruments) is used as the main control unit in the system, see Figure 3.12 It combines a reconfigurable FPGA (Xilinx Z-7010), enabling fast and deterministic control execution [28]. The processor features two cores with a clock speed of 667 MHz and supports multiple input and output interfaces. The MyRIO is well suited for control applications due to its ability to perform parallel and execute multiple control tasks in real time. The integrated FPGA allows low-level signal processing directly from sensors, enabling fast response times and high control accuracy. The device can communicate with external systems via USB or WiFi [28], which allows real-time monitoring and parameter tuning during operation. However, to improve communication stability and extend the range, an external communication unit like an Industrial Cellular Router (ICR) can be used in the system.

Figure 3.12: myRIO controller

3.2 Software

The software used in this project supports modeling, simulation, and real-time control of the system. It enables the implementation of control algorithms, communication with hardware components, and validation of system behavior prior to physical testing.

3.2.1 MATLAB and Simulink

MATLAB is a numerical cumputing environment developed by MathWorks, widely used for data analysis, signal processing, and control system design [29]. It provides a programming language optimized for numerical computation, visualization, and algorithm development [30].

Simulink, built on MATLAB, is a graphical block-diagram environment used for modeling and simulating dynamic systems [14]. It allows the design and testing of control systems without the need for low-level programming, making it suitable for validating system behavior prior to implementation on hardware.

3.2.2 LabVIEW

Laboratory Virtual Instrument Engineering Workbench (LabVIEW) is a workbench for graphical programming developed by National Instruments, based on data flow programming [31]. It uses virtual instruments (VIs) and graphical connections (wires) to represent program structure and data flow.

In this project, LabVIEW is used to implement control algorithms on the myRIO platform and to handle real-time data acquisition and control. Its integration with hardware enables efficient testing, monitoring, and adjustment of system parameters during operation.

3.2.3 VESC-Tool

The VESC Tool is an open-source software used to configure, control, and monitor the VESC motor controller [32]. It provides an interface for setting parameters such as current limits, voltage limits, and motor characteristics.
In this project, the VESC Tool is primarily used for configuration and diagnostics of the motor controller, ensuring safe and reliable operation of the actuator.

3.2.4 Arduino IDE

The Arduino Integrated Development Environment (IDE) is an open-source development environment used for writing, compiling, and uploading firmware to Arduino-compatible microcontrollers. It provides a simple interface for developing embedded applications and supports programming in C/C++ through a wide range of built-in libraries and tools.
In this project, the Arduino IDE is primarily used for developing and uploading control software to the microcontroller, enabling communication with sensors and actuators, as well as implementation of system logic. The built-in serial monitor of the software is used for real-time debugging and data visualization during the testing phase.

4
Method

This chapter describes the methodology of the project, outlining the development process from initial system design to final implementation and testing. It presents the sequence of steps taken throughout the project and highlights the key decisions and milestones that shaped the system.

The chapter includes the development of the control strategy, simulation and analysis in MATLAB, and procedures used for validation and experimental testing. Together, these parts provide a structured overview of how the system was designed, evaluated and refined.

4.1 Simulating in MATLAB

To determine the maximum recoverable roll angle, initial simulations of the state-space model (derived in 2.1.4) were conducted in MATLAB. A single PID controller was first implemented to investigate the maximum roll angle, θ, across various simulated scenarios. This helped identify the primary variables influencing the roll angle and provided deeper insight into the behavior of the state-space system.

The system's states were plotted against time for each scenario. Aligning these time-domain plots allowed for a direct comparison of how different states interacted and evolved simultaneously. These simulations aimed to identify the factors that most significantly limit the maximum angle from which the bicycle can recover. The variables were tested both individually and in combination to determine if coupled effects could allow the bicycle to stabilize from a steeper initial angle.

Key parameters, including current, controller types, disturbances, cascade regulation, zero-angle reference, nonlinear dynamics, time delays and sampling rates, were systematically varied. A 10 second simulation window was chosen as the baseline, if the bicycle remained stable for 10 seconds without unbounded state growth, it was deemed permanently stable, assuming a continuous power supply to the electronics.

4.1.1 Pole Placement Method

The pole placement method was used to determine the controller parameters that achieve the desired closed-loop behavior of the system. The closed-loop poles were

selected as

$$p_{1,2} = -6 \pm j6, \quad p_3 = -15, \quad p_4 = -20, \quad p_5 = -25,$$

These poles correspond to a stable system with a sufficiently fast rsponse and moderate damping. The resulting characteristic polynomial is given by

$$P(s) = (s + 25)(s + 20)(s + 15)(s^2 + 12s + 72). \tag{4.1}$$

From the state-space model in equation (2.20), the transfer function of the system can be expressed as

$$H(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}. \tag{4.2}$$

By matching the transfer function with the desired characteristic polynomial, the gains were determined such that the closed-loop system attains the specified pole locations.

4.1.2 Non-zero Reference Value

A non-zero reference value was introduced to evaluate the motors ability to maintain the bicycle at a specific targeted angle rather than just an upright equilibrium. This test verified whether the model could effectively track and stabilize at a strict angular reference.

4.1.3 Nonlinear State Comparison

A nonlinear model was implemented in the MATLAB code to enable a comparison between the used state space model and the nonlinear model. This allowed for a validation of the linear model. Both models was plotted in the graphs to allow direct visual comparison between the two representations. This was done to ensure that no significant deviations occurred and to verify that the linearization was accurate.

4.1.4 Disturbances on a Stable System

To access robustness, a simulated disturbance to the roll angle was introduced five seconds into the simulation, after the controller had successfully stabilized the bicycle at its calculated maximum roll angle. This test evaluated the system's ability to reject external disturbances and re-establish stability, ultimately defining the maximum disturbance angle the controller could handle.

4.1.5 Time Delay

Artificial time delays were introduced to the control loop to evaluate the system's sensitivity to delay. This determined the maximum acceptable processing or communication delay before the system became unstable while simultaneously improving the resemblance to reality for the simulation.

4.1.6 Regulators and Cascade control

Evaluating different controller architectures and parameters was important for optimizing system performance. By zeroing out the integral or derivative gains, the standard PID controller was simplified into a PI or PD controller. These simple regulators were evaluated to determine if they could stabilize the bicycle more effectively than a full PID setup.

Furthermore, cascade control was implemented to address the significantly different response times of the system's internal dynamics, and to integrate with the preinstalled VESC motor controller. Rather than relying on a single PID controller to estimate the current required to correct the roll angle, the cascade architecture uses an inner PI controller (within the VESC) to regulate current, and an outer PD controller to regulate the angle. This modular approach also allows for future hardware changes without requiring a complete retuning of the control system.

4.2 Validation of Maximum Roll Angle

The theoretical maximum roll angle was physically validated in the laboratory. The bicycle was positioned as an inverted pendulum and manually held at the maximum roll angle calculated by the MATLAB simulations. During this phase, the active electronics were disabled. The physical check confirmed that the simulation maximum roll angle was realistic and provided a safe, practical starting point for subsequent active stabilization tests.

4.3 Hardware Testing

This section describes the practical implementation and testing of the system, where the developed models and control strategies are evaluated under real-world conditions.

4.3.1 Controller Platform Selection

Initially, the system was designed to use the MyRIO to handle control and communication. The MyRIO platform offers high computational capacity and flexibility, making it well suited for complex systems involving multiple sensors or more advanced control strategies.

However, at this stage of the project, a significant issue emerged: the hardware was not responding to the software. This problem could have been caused by several factors, including incorrect wiring connections, improper pin configuration, or compatibility issues between the hardware and software components. Therefore, an investigation was initiated to verify that all essential hardware components were

functioning correctly, namely the NI myRIO, the IMU sensor, the VESC Tool and the motor itself. Each component was tested individually and confirmed to operate properly. Nevertheless, the issue remained unresolved.

Due to time constraints, the project plan was adjusted by replacing the NI myRIO with the LOLIN32 Lite. This change successfully resolved the problem, as the hardware integration became less complex, while the software development environment, using the Arduino IDE, proved to be simpler and more efficient to work with.

The LOLIN32 Lite provides sufficient computational performance for the control task while significantly simplifying the implementation process. It supports development using the Arduino framework, which enables faster prototyping compared to the LABVIEW-based environment required for the MyRIO. In addition, the LOLIN32 Lite integrates wireless communication capabilities such as WiFi and Bluetooth directly on the chip.

The smaller physical size and lower system overhead further contributed to the selection of the LOLIN32 Lite, making it a more efficient solution for the current application while still meeting the system requirements. The choice reflects a trade-off between system flexibility and implementation efficiency.

4.3.2 IMU Test

The functionality of the IMU sensor was tested by connecting the IMU to the LOLIN32 Lite and reading the sensor data through a C++ program uploaded to the board using the Arduino IDE.

The (IMU) provides raw measurements of linear acceleration and angular velocity along the three spatial axes: X, Y, and Z. The accelerometer readings were represented as acceleration components a_x, a_y, and a_z, while the reaction wheel readings were represented as angular velocity components g_x, g_y, and g_z.

The acquired data were processed using a complementary filter, which provided stable measurements at a sampling interval of 10 ms. Subsequently, the tilt angle was calculated using the following equation:

$$\theta = arctan(a_y, a_z)\frac{180}{\pi} \tag{4.3}$$

After calculating the tilt angle, its accuracy was experimentally validated by comparing the obtained angle values with the actual physical orientation of the bicycle.

4.3.3 myRIO Test

The proper operation of the NI myRIO was verified by transmitting data through a program developed in LabVIEW, as shown in Figure 4.1. The device successfully responded by controlling the built-in LEDs and displaying the values measured by the integrated accelerometer.

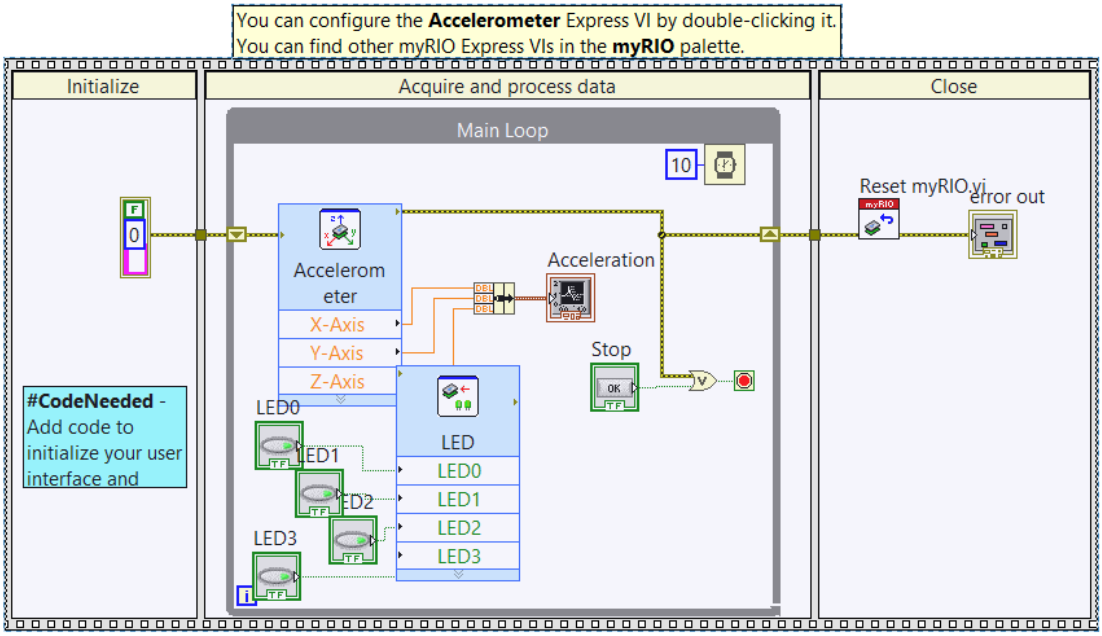


Figure 4.1: Testing myRIO controller throught LabVIEW CODE

4.3.4 VESC With The Motor Test

To verify the operation of both the motor and the VESC, the software VESC Tool was installed and used to control and test the motor. The motor was operated by sending either a specified current command (for example 5 A) or a target rotational speed command (for example 5000 RPM), and it responded successfully in both cases. Furthermore, the minimum current required to rotate the motor under no-load conditions was determined experimentally and found to be 3 A.

4.3.5 Actuator Mount Design and Fabrication

The actuator mount was designed to interface the motor assembly with the bicycle frame. Initial dimensions were obtained through physical measurements of the bicycle combined with reference geometry from CAD models of the frame and the pre-existing motor holder.

The design process was iterative and carried through by prototyping using 3D printing. Several partial prints were produced to evaluate geometric fit, particularly the mounting angle and the interaction with the cylindrical frame tubes. Some of these prototypes can be seen below in Figure 4.2. This allowed verification of alignment and structural compatibility before manufacturing the complete component.

Figure 4.2: Prototype pieces to evaluate fit

To ease assembly, hexagonal cavities were integrated into the design to house nuts, enabling easier tightening of bolts using only one tool. During the first full-scale print, the bolt holes were dimensioned equal to the nominal screw diameter, which resulted in the threads cutting into the plastic during assembly. The design was subsequently updated by increasing the hole clearance, leading to improved assembly and repeatability.

The final design provided a stable mechanical connection between the actuator and the frame, with sufficient rigidity to withstand vibrations induced by the motor and reaction wheel during operation. The completed mount is shown in Figure 4.3.

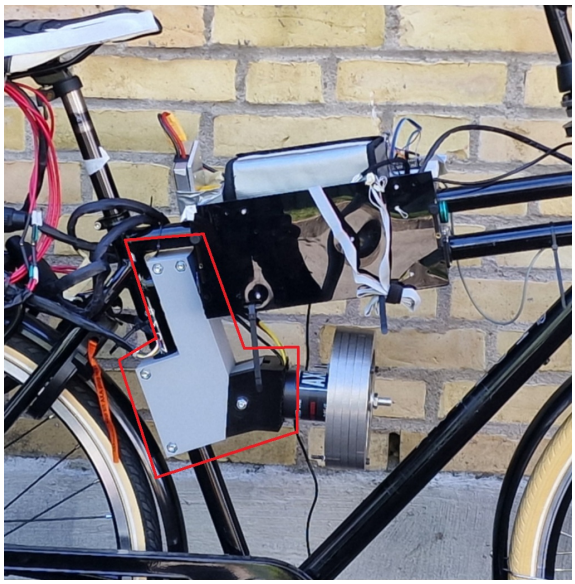


Figure 4.3: Finished Grey Actuator Mount

4.4 Physical Bicycle Test

This section presents the experimental validation of the proposed control system implemented on the physical bicycle platform. It describes the complete integration of the hardware components, the development of the control algorithm, and the real-world testing procedure. Furthermore, it discusses the initial system performance, the challenges encountered during experimentation, the modifications introduced to improve stability and finally, the systematic tuning of the PD controller parameters.

4.4.1 System Hardware Integration

To evaluate the system as a whole, the hardware components were first interconnected by connecting the IMU sensor to the LOLIN32 Lite microcontroller. The microcontroller was then connected to the VESC, which in turn was connected to the motor.The battery was connected to the VESC to supply power to the motor controller and actuator.

4.4.2 Control System Implementation

Using the Arduino IDE, a C++ code was developed to implement the complete control system. The code first defined the pins on the LOLIN32 Lite connected to both the IMU and the VESC.With sampling time 10 ms, the sensor data from IMU was then acquired and processed using a complementary filter to estimate the actual tilt angle of the bicycle. This angle was subsequently sent to a PD controller, which calculated the error between the measured tilt angle and the reference angle. Based on this error, the controller computed the electrical current required to restore balance.

The resulting value represented a reference current command transmitted to the VESC. The motor controller then calculated the difference between the reference current and the actual motor current, generating a corrected output current that drove the motor. Consequently, the motor produced a compensating torque equal in magnitude and opposite in direction to the gravitational torque acting on the bicycle, thereby achieving the balancing effect.

4.4.3 Experimental Testing and Initial System Performance

The integrated hardware and software system was tested experimentally, and it was observed that the reaction wheel responded in the opposite direction of the bicycle tilt, as expected. However, a significant issue was identified: the reaction wheel response lagged behind rapid changes in the bicycle tilt direction, which caused instability and prevented successful balancing. After further analysis, it was concluded that the current supplied to the motor was sometimes smaller than the induced current generated within the motor itself and opposite in direction.

4.4.4 Motor Current Correction Using Back EMF

To address this issue, the motor rotational speed (RPM) was measured using one of the VESC libraries available in the Arduino IDE. The back electromotive force (back EMF) was then calculated, followed by the estimation of the induced current generated by the motor. This induced current was inverted and added to the commanded current, ensuring that only the desired effective current remained. Experimental observations confirmed a clear improvement, as the reaction wheel began responding immediately to variations in bicycle tilt.

4.4.5 PD Controller Gain Tuning

Following this improvement, the next stage involved tuning the PD controller gains, namely K_p and K_d. The proportional gain K_p was responsible for scaling the commanded current and therefore increasing the motor torque, while the derivative gain K_d provided predictive behavior by considering the rate of change of the error signal. Several gain values were tested experimentally by manually adjusting K_p and K_d one at a time. The proportional gain K_p was varied from 4 to 1000, while the derivative gain K_d was varied from 0.5 to 35.

5
Results

This chapter presents the results and findings of the simulated control systems. The results are structured by first confirming the maximum recoverable angle, followed by an evaluation of the differences between various controller systems (PI, PD, PID, Cascade). Thereafter, MATLAB parameter testing simulations are presented to assess the system's robustness under various conditions. Then, a comprehensive evaluation of the best-performing controller is provided to conclude the findings. This section finally presents the results from physical testing and evaluates whether the controller parameters obtained through MATLAB simulations agree with the parameters that produced the best performance in the real system.

5.1 Maximum Recoverable Angle Calculation

Recall the dynamic equation,

$$I\ddot{\theta} = mgl\theta - I_f\dot{\omega} + b\omega,$$

and the relationship,

$$\tau_{gyro} = I_f\,\dot{\omega} = K_m i_a.$$

assuming that the bicycle is in a stationary position where $\ddot{\theta} = 0$, $\dot{\omega} = 0$ and that any friction is negligible, the equation simplifies to

$$0 = mgl\theta - K_m i_a. \tag{5.1}$$

Inserting the limit in current from Table 3.2, the values in Table 5.1 and solve for θ_{max}, one achieves

$$\theta_{max} = \frac{K_m i_a}{mgl} \approx 2.35^\circ, \tag{5.2}$$

which represents the theoretical maximum roll angle that the reaction wheel can counteract under the current limit and simplified torque-balance model.

Table 5.1: Values for parameters that were used in the calculation and simulations.

Variable	Value
m	30 kg
g	9.82 m/s^2
l	0.543 m
I	8.55 kgm^2
I_f	1.99 kgm^2
b	$3.8 \cdot 10^{-4}$
R_a	$63 \cdot 10^{-3}\Omega$
L_a	$1 \cdot 10^{-3}$ H
K_u	0.1005 V/rad/s
K_m	0.1005 Nm/A
i_{max}	65 A
u_{max}	48 V

5.2 Comparative Analysis of Control Systems in MATLAB

To determine the most suitable control strategy for the bicycle, PI, PD, PID, and cascade controllers were evaluated. The performance of each system was compared by analyzing the roll angle, θ, the roll velocity, $\dot{\theta}$, reaction-wheel speed, ω, and motor current, i_a. The proportional, integral and derivative gains (K_p, K_i, and K_d) were tuned to optimize the performance of each respective controller. Simulations were done to identify the maximum initial roll angle from which each controller could successfully stabilize the system.

5.2.1 PI Controller Evaluation

MATLAB simulations were performed to evaluate the feasibility of using a PI controller for this system. A PI controller relies solely on current and past errors, lacking the predictive capability provided by a derivative term. For a highly unstable inverted pendulum system, this lack of damping is critical.

As seen in Figure 5.1 below, the system fails to achieve stability even with a very small initial roll angle of $\theta = 0.00005$ *radians*. The graphs show that the roll angle diverges rapidly towards failure. Because of this, the roll velocity rapidly increases in magnitude, indicating that the bicycle is moving away from the upright equilibrium, and the motor current shows severe oscillations before saturating at its limit. This shows that the controller is overcompensating and is unable to counteract the gravitational forces. Consequently, the standalone PI controller is deemed inadequate for stabilizing the bicycle.

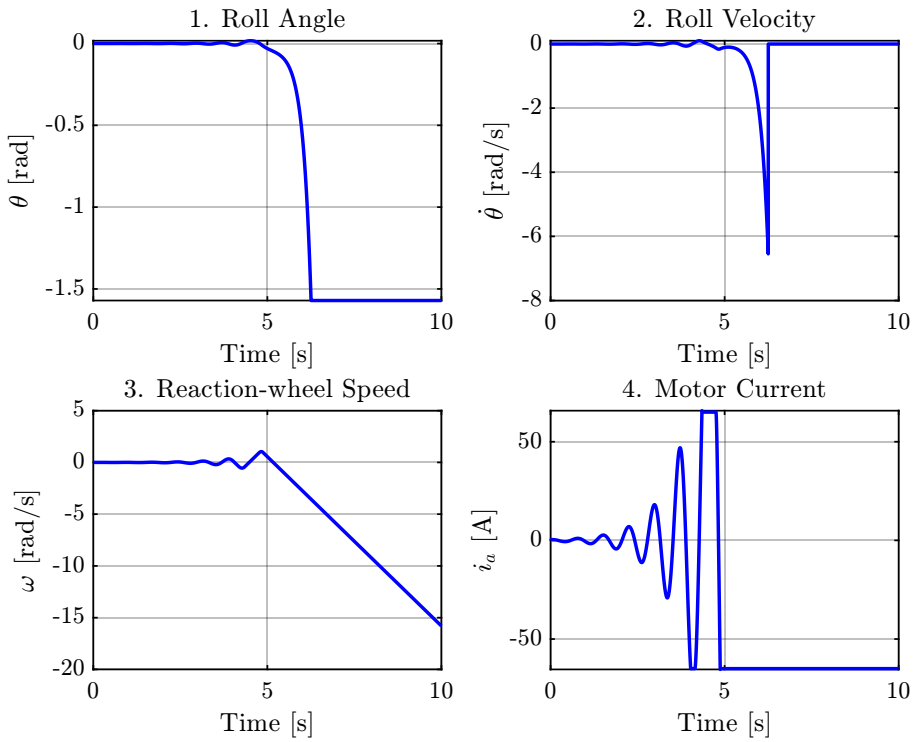


Figure 5.1: PI controller simulation with an initial angle of $\theta = 0.00005$ radians.

5.2.2 PD Controller Evaluation

Because of the derivative action in the PD controller, there is the necessary damping to anticipate and correct future errors, which is vital for stabilizing the bicycle. As seen in Figure 5.2, the PD controller successfully stabilizes the system from the previously calculated maximum recoverable angle of $\theta = 0.041$ radians. However, a closer look at the reaction-wheel speed reveals a limitation: the speed continuously increases over time. This indicates a steady-state error, meaning the actuator has to constantly accelerate the reaction wheel to maintain the bicycle's upright position, which is unsustainable for long-term operation.

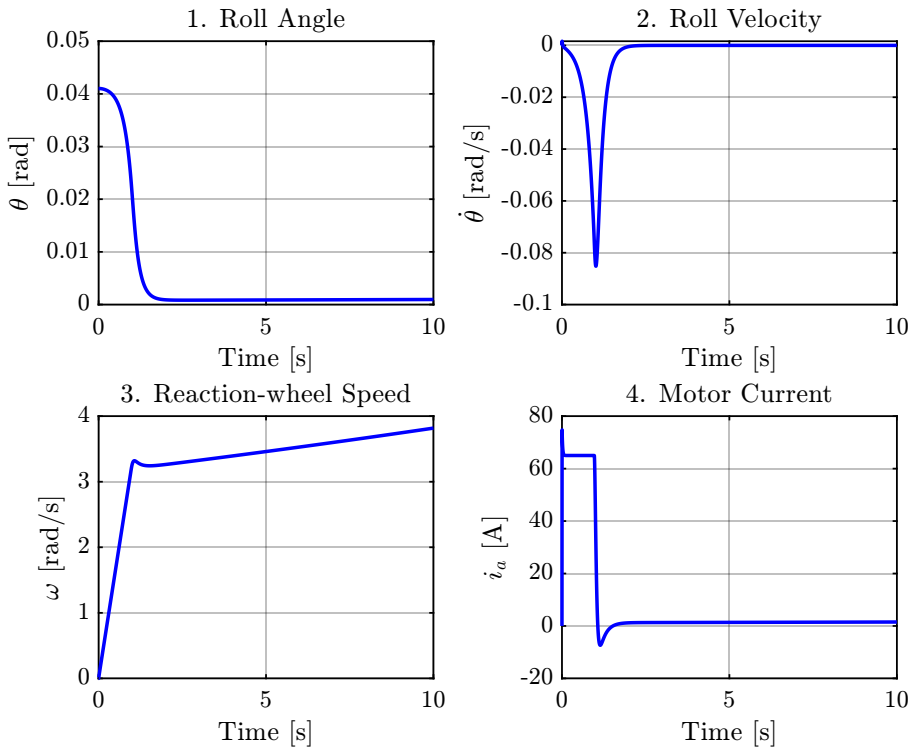


Figure 5.2: PD controller simulation with an initial angle of $\theta = 0.041$ radians.

5.2.3 PID Controller Evaluation

To address the steady-state error observed in the PD controller, an integral term was added to form a complete PID controller. The integral action is designed to eliminate residual errors by accumulating past discrepancies over time. Figure 5.3 demonstrates the PID controller's performance with a maximum initial roll angle of $\theta = 0.039$ radians. After an initial spike to generate the necessary restorative torque, the reaction wheel speed smoothly returns to zero. Similarly, the motor current returns to zero after the bicycle reaches equilibrium. This proves that the PID controller not only stabilizes the roll angle but also successfully manages the reaction-wheel's momentum.

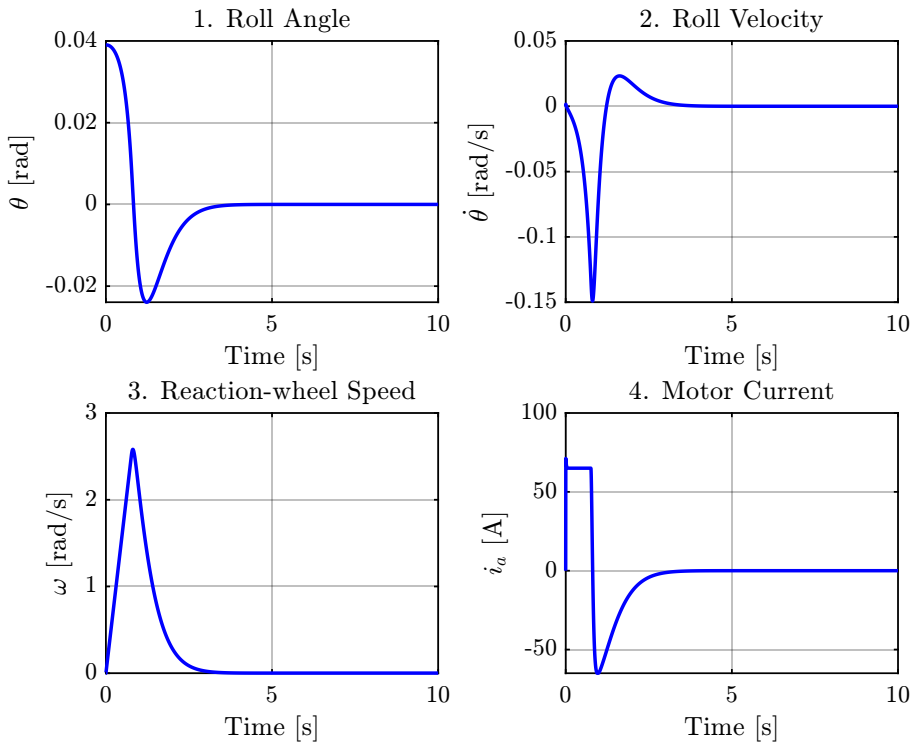


Figure 5.3: PID controller simulation with an initial angle of $\theta = 0.039$ radians.

5.2.4 Cascade Control Evaluation

To further explore stabilization, a cascade control architecture was evaluated. As seen in Figure 5.4, the cascade controller successfully stabilizes the bicycle from a maximum roll angle of $\theta = 0.04$ radians. The system demonstrates a smooth response, with both roll angle and roll velocity returning to equilibrium without noticeable overshoot.

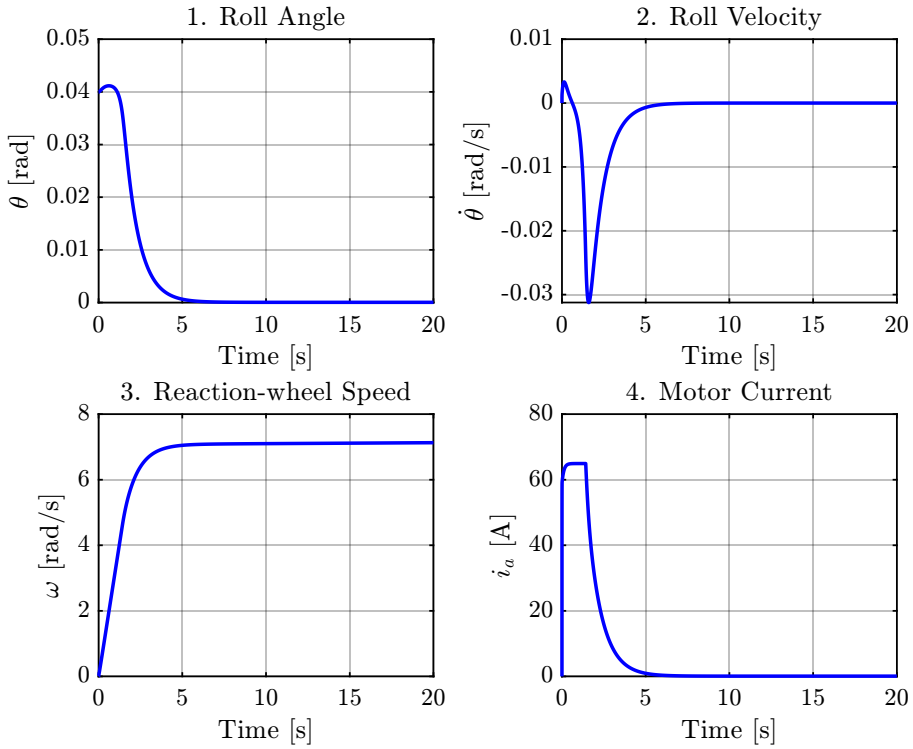


Figure 5.4: Cascade controller simulation with a maximum initial angle of $\theta = 0.04$ radians.

When comparing these results to the standalone PID evaluation, the cascade controller demonstrates several practical advantages that make it superior for physical implementation. While both successfully stabilize the system, the cascade controller recovers from a marginally larger maximum initial angle ($\theta = 0.04$ radians compared to 0.039 radians) and completely eliminates the roll angle overshoot present in the PID response. Furthermore, although the PID controller successfully returns the reaction-wheel speed to zero, it relies on an aggressive control effort characterized by a harsh, instantaneous motor current reversal from postitive to negative saturation. In contrast, the cascade controller achieves stability with a significantly smoother motor current profile that gently decays to zero, reducing mechanical and electrical stress on the system.

Beyond these simulated performance improvements, the cascade control structure is inherently compatible with the physical hardware setup. The multi-loop architecture aligns perfectly with the existing motor drive configuration. Specifically, the outer balancing loop naturally generates a reference command that interface directly with the VESC's integrated inner current loop. This direct compatibility ensures reliable, highly responsive motor control while effortlessly bridging the gap between theoretical simulation and practical application.

5.3 MATLAB Parameter Testing

Having established the baseline performance of the various controllers, this section evaluated the system's robustness under simulated real-world conditions. A series of MATLAB parameter tests is conducted to challenge the control architectures against non-ideal scenarios, including non-zero reference angels, nonlinear system dynamics, external physical disturbances, and sensor time delays. These tests are helpful in identifying the practical operational limits and stability boundaries of the designed control systems before physical implementation.

5.3.1 Non-zero Reference Value

As shown by the subplots in Figure 5.5, the PID controller successfully stabilizes the bicycle at the non-zero reference roll angle, with the roll velocity ($\dot{\theta}$) quickly converging to zero. However, to maintain this constant non-zero roll angle against the continuous gravitational pull simulated by the slope, the system requires a constant restorative torque. This can be seen in the motor current subplot, which settles at a continuous draw of approximately 40 Amps.

In the reaction-wheel speed subplot, the wheel's rotational speed increases linearly. This will be physically unsustainable for long-term operation, as the actuator motor will inevitably reach its saturation point or mechanical speed limit. The maximum angle that the system could handle was $\theta \approx 1.43°$.

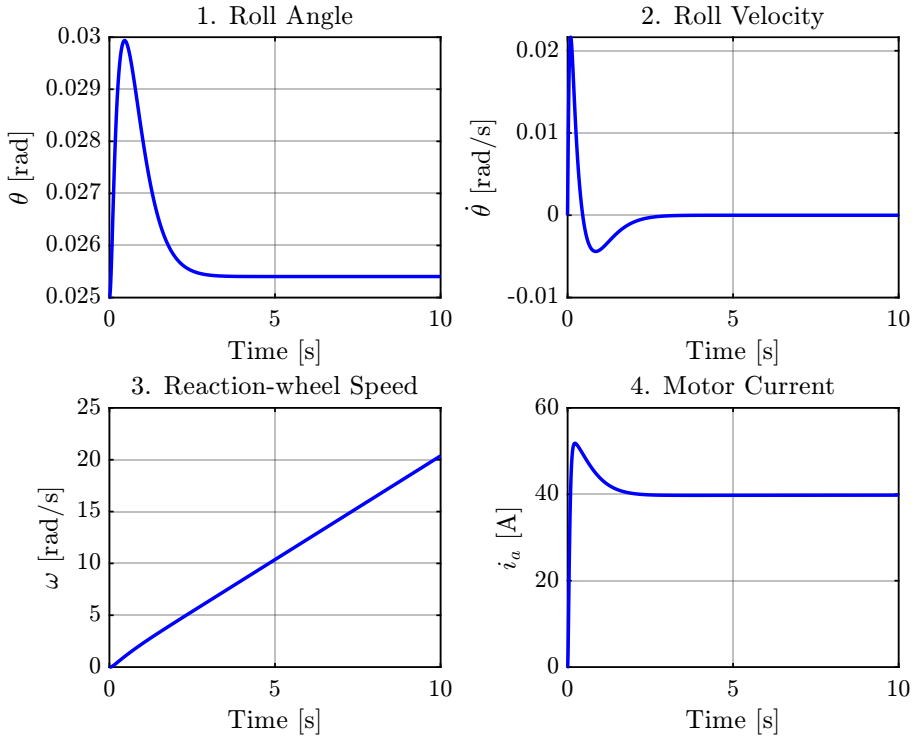


Figure 5.5: PID controller with a non-zero reference value.

5.3.2 Nonlinear State Comparison

Figure 5.6 compares the performance of the PID controller applied to both the linear and nonlinear system models. Across all four subplots, the responses of the two models are visually identical, with the trajectories overlapping almost perfectly. To quantitatively evaluate this alignment, the Root Mean Square Error (RMSE) for the roll angle, θ, was calculated:

$$\mathrm{RMSE}_\theta = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left(\theta_{\mathrm{n},k} - \theta_{\mathrm{l},k}\right)^2} \tag{5.3}$$

where N is the total number of samples, $\theta_{\mathrm{n},k}$ is the nonlinear response, and $\theta_{\mathrm{l},k}$ is the linear response at sample k. The calculation yielded an error of 2.44×10^{-5} rad. This small number confirms that the linear approximation is highly accurate for the operating range tested (small roll angles) and fully validates the use of the linearized model for designing and tuning the bicycle's control system.

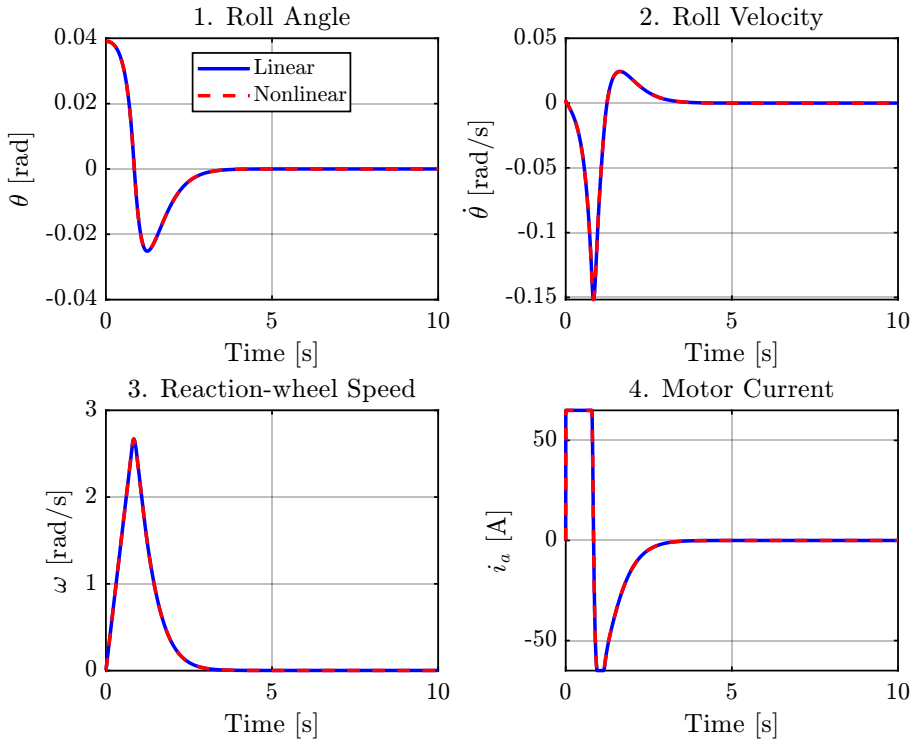


Figure 5.6: PID controller comparison between a nonlinear system and a linear system.

5.3.3 Disturbance on a Stable System

Figure 5.7 illustrates the cascade controller's robustness when subjected to a sudden external force. During the first five seconds, the system demonstrates standard recovery from an initial roll angle of 0.04 radians. As expected, the controller smoothly drives the roll angle and roll velocity to zero, while the motor current return to a

resting state. By $t = 5$ seconds, the system has reached a stable equilibrium, with the reaction-wheel maintaining a constant rotational speed of about 6.5 rad/s.

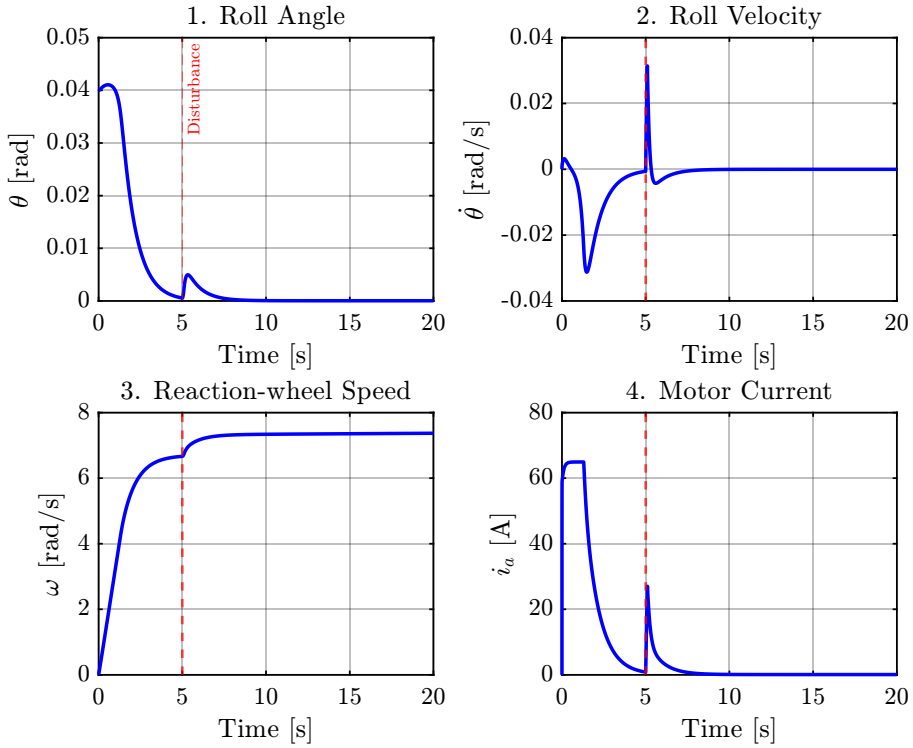


Figure 5.7: Cascade controller response to a 4 Nm external torque disturbance applied to the roll angle for 0.1 seconds.

As indicated by the red dotted vertical lines across all subplots, a simulated physical disturbance is introduced at exactly $t = 5$ seconds. This disturbance is modeled as a 4 Nm external torque impulse applied to the roll angle for a duration of 0.1 seconds. The impulse causes an abrupt displacement in the roll angle and an immediate spike in the roll velocity. In response, the cascade controller sharply increases the motor current to generate the necessary corrective torque. The system is successful in damping the disturbance and returning the bicycle to upright position ($\theta = 0$) in roughly two seconds. To counteract the energy introduced by the disturbance and maintain the restored equilibrium, the reaction-wheel settles at a new, higher steady-state speed of about 7.5 rad/s.

5.3.4 Time Delay

Figure 5.8 illustrates the effects of introducing a 50-millisecond time delay into the PD control loop. This sensor or processing delay significantly degrades the system's transient response, causing pronounced oscillations across all variables as the controller repeatedly overcompensated, based on the outdated feedback. While the system ultimately manages to stabilize the roll angle, the delay forces the motor current (i_a) to oscillate violently, indicating severe and potentially damaging strain

on the actuator. Furthermore, once the initial oscillations subsides, the underlying limitation of the standalone PD controller remains apparent, as the reaction-wheel speed (ω) continues its unsustainable steady-state climb.

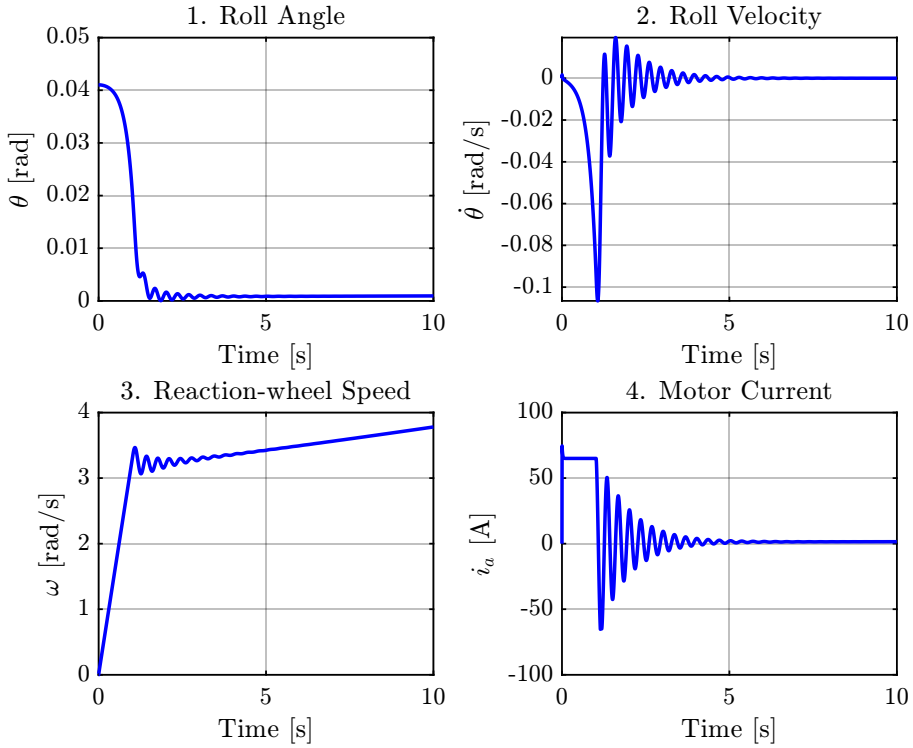


Figure 5.8: PD controller with a time delay of 50 milliseconds.

To further show the boundaries of the system's tolerance to latency, a sweep of delay values was evaluated, as illustrated in Figure 5.9. Table 5.2 summarizes the stability and observed behavior across these intervals.

Table 5.2: Effects of varying time delays on PD controller stability.

Delay	**Stable?**	**Observation**
10 ms	Yes	Rapid stabilization with minimal visible oscillation.
25 ms	Yes	Minor initial oscillations, settling within 2 seconds.
50 ms	Yes	Pronounced initial oscillations, requires more time to stabilize.
75 ms	No	System fails to settle, exhibiting a sustained, bounded oscillation.
100 ms	No	Severe limit cycle with a visibly larger amplitude.

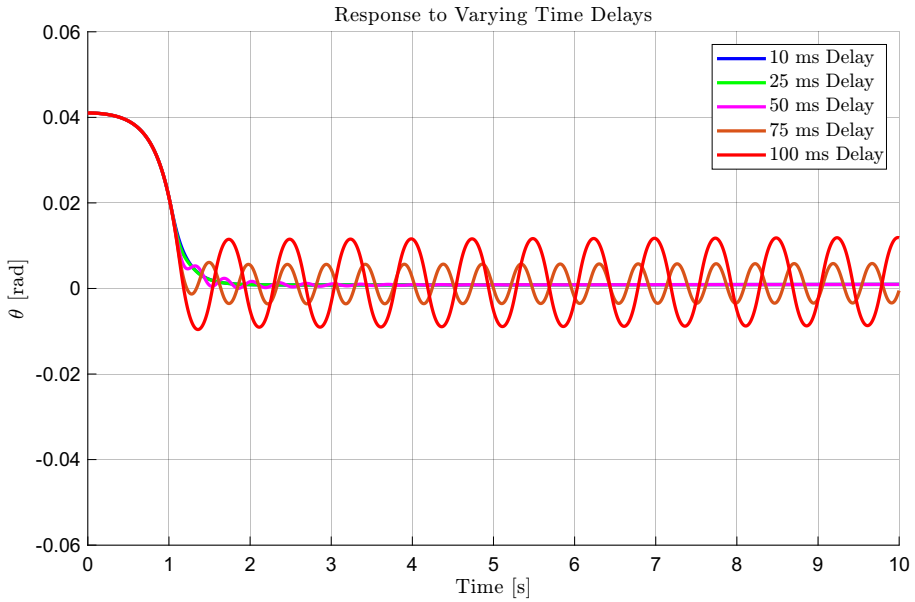


Figure 5.9: Response of the roll angle evaluated across varying sensor time delays.

5.4 Evaluation of Best Controller

To systematically determine the most suitable control strategy, the performance of the PI, PD, PID and cascade controllers were evaluated against objective performance and hardware criteria. The results, derived from the simulated responses of the maximum recoverable initial roll angles, are summarized in Table 5.3.

Table 5.3: Objective performance comparison of evaluated control systems.

Metric	PI	PD	PID	Cascade
Max Angle (rad)	Fails	0.041	0.039	0.040
Settling Time (s)	Fails	∼2.0	∼4.0	∼5.0
Overshoot	N/A	Minimal	High	None
Peak Current (A)	Oscillates	65	±65	65
Final Wheel Speed (rad/s)	Diverges	Unbounded	0	∼7.1
Robustness	Low	Moderate	Good	Excellent
Delay Sensitivity	High	Moderate	High	Low
Hardware Implementability	Inadequate	Unsustainable	High Stress	VESC Compatible

As demonstrated in the comparison, the PI and PD controllers are inadequate for physical implementation due to instability and unbounded steady-state errors in the reaction wheel's velocity, respectively. The decision ultimately narrows down to the PID and cascade controllers. While the PID controller successfully manages the reaction wheel's momentum by returning its final speed to zero, it suffers from significant roll angle overshoot and requires an aggressive current reversal from positive to negative saturation. This harsh electrical demand makes the PID highly sensitive

to system delays and introduces severe mechanical stress.

Due to what is mentioned above, the cascade control system is selected as the optimal choice. It recovers from a highly competitive maximum angle of 0.04 radians with no overshoot and features a smooth current decay curve that protects the motor hardware. Also, it provides robustness to disturbances and seamless implementability, as the outer balancing loop perfectly interfaces with the VESC's existing inner PI current controller.

5.5 Physical Test

The physical testing demonstrated that the reaction-wheel controller was capable of counteracting the bicycle roll motion and improving the balancing behavior of the system. The actuator responded consistently to tilt variations, and the implemented control strategy showed clear agreement with the simulated behavior for small disturbances. However, fully autonomous equilibrium could not be maintained consistently under the current hardware configuration. The performance was limited by factors such as insufficient motor torque, increased total system mass, structural vibrations, and thermal limitations in the actuator assembly. Despite these limitations, the experiments confirmed that the stabilization principle is viable and provided valuable insight into the practical challenges associated with stationary reaction-wheel balancing.

5.5.1 IMU

Figure 5.10 illustrates the acceleration and angular velocity components acquired from the IMU sensor mounted on the bicycle while the bicycle was subjected to slight movements along the Y-axis Figure 5.11 . This procedure was performed to verify the sensor response speed and to ensure that the measured roll angle corresponded accurately to the actual bicycle inclination.

The roll angle θ in Figure 5.10 is presented in the red Box 4 and was calculated by substituting the values of a_y shown in the red Box 1, together with the values of a_z shown in the red Box 2 and g_x shown in the red Box 3, into the complementary filter Equation 2.24.

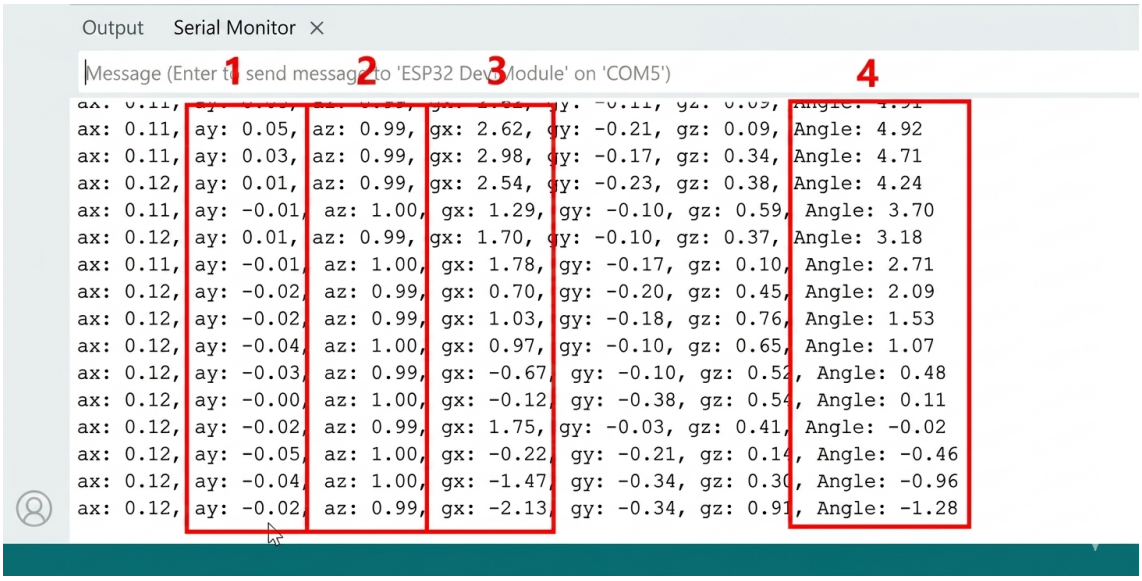


Figure 5.10: IMU Readings

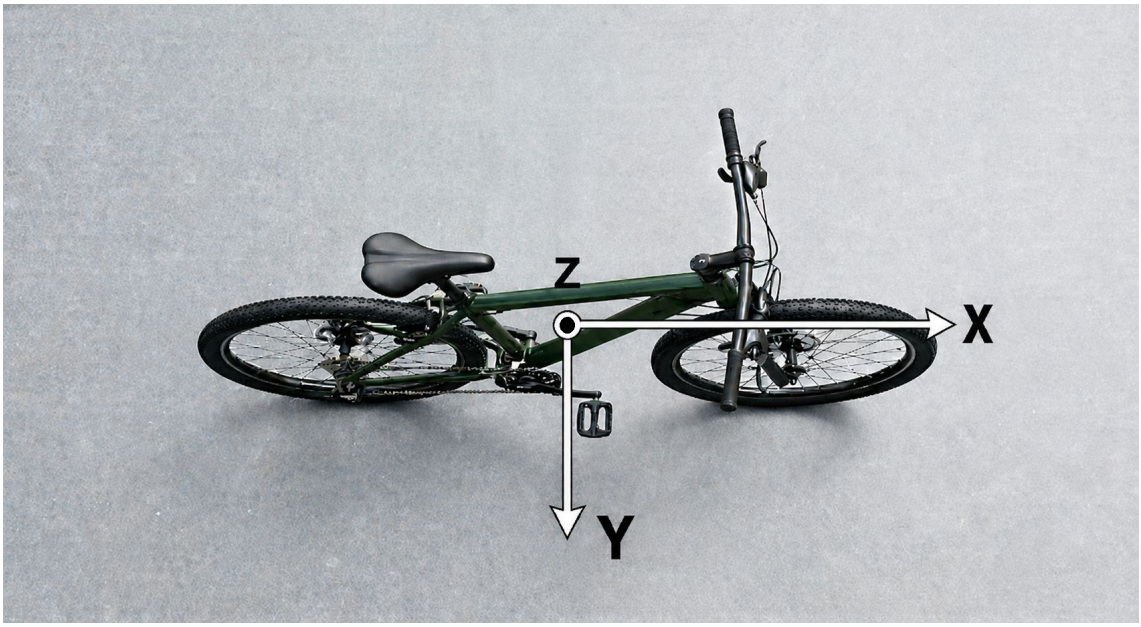


Figure 5.11: Coordinate Axes

5.5.2 myRIO

The functionality of the NI myRIO was verified using the LabVIEW program in Figure 4.1. The developed code was used to acquire acceleration measurements from the integrated accelerometer and visualize the values along the X, Y, and Z-axes. Furthermore, the built-in physical LEDs on the myRIO were controlled to validate the output response of the device.

The results confirm that the myRIO was operating properly, as it was able to acquire

input data from the accelerometer and produce output signals by activating the built-in LEDs.

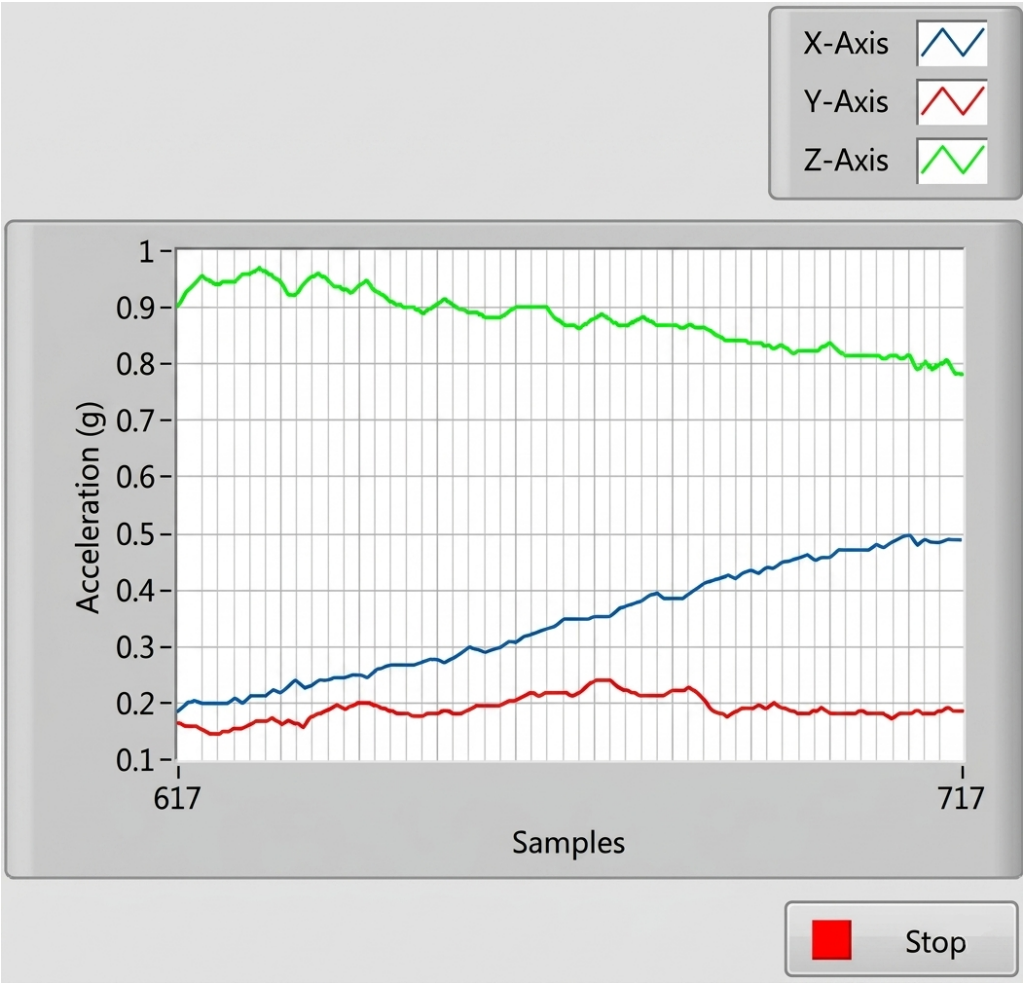


Figure 5.12: myRIO accelerometer measurements along the three axes

5.5.3 Effect of Back-EMF Compensation on Reaction-Wheel Response

The experimental results show the system response before and after applying the back-EMF current correction. The measured variables are the motor rotational speed, the bicycle roll angle, and the total compensated motor current, as shown in Figures 5.13 and 5.14. The total compensated motor current represents the final current command sent to the motor controller, obtained by combining the desired control current with the back-EMF compensation current.

Before applying back-EMF compensation, the reaction wheel responded to the bicycle tilt, but its response was delayed. As shown in Figure 5.13, the motor speed remained relatively high, while the total current changed repeatedly between positive and negative values. This shows that the controller was trying to correct the bicycle tilt by changing the direction of the motor current. However, because the

motor was rotating, it generated an induced current due to back EMF. In some cases, this induced current was opposite to the supplied current and reduced the effective current applied to the motor. As a result, the motor could not generate the required braking or reversing torque at the correct time, so the reaction wheel continued spinning due to its inertia and maintained a high RPM. This caused a delay in the system response and reduced the stability of the bicycle. After applying the back-EMF current correction, the system response improved clearly. As shown in Figure 5.14, the motor RPM followed the commanded current more efficiently, indicating that the motor response became more consistent with the PD control output. The total current continued to change direction according to the required correction, and the added compensation helped cancel the unwanted induced current. Therefore, the effective current applied to the motor became closer to the desired control current. This allowed the motor to accelerate, decelerate, and reverse more effectively, preventing unnecessary wheel rotation and allowing the reaction wheel to respond more directly to changes in the bicycle tilt.

Overall, the comparison between Figures 5.13 and 5.14 confirms that back-EMF compensation improved the dynamic response of the reaction-wheel system. By calculating the induced current and adding its inverse to the commanded current, the controller was able to reduce the negative effect of back EMF. This allowed the reaction wheel to respond more quickly to bicycle tilt variations and generate corrective torque with less delay. Although complete self-balancing was not achieved, the compensated system showed better responsiveness and improved stability, demonstrating the importance of including back-EMF correction in the motor control strategy.

In Figure 5.13 and Figure 5.14, the horizontal axis represents time progression during the experimental test where the sampling index is 10 ms.The vertical axis represents the measured values of the three plotted signals: motor rotational speed in RPM, bicycle roll angle, and total commanded motor current (without the back-EMF compensation current in Figure 5.13).

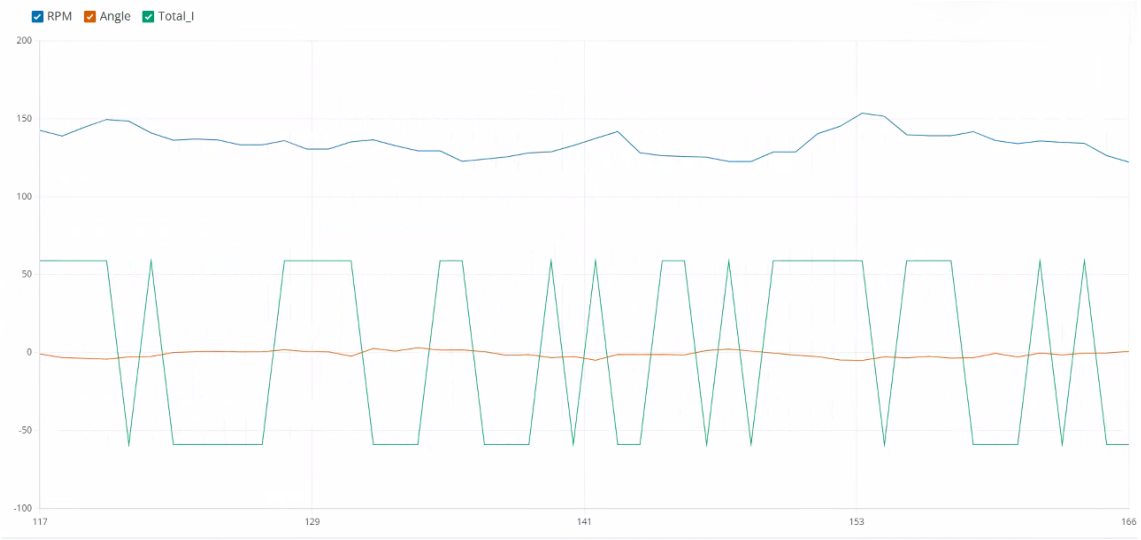


Figure 5.13: Experimental response of the reaction-wheel system without back-EMF current compensation over time.

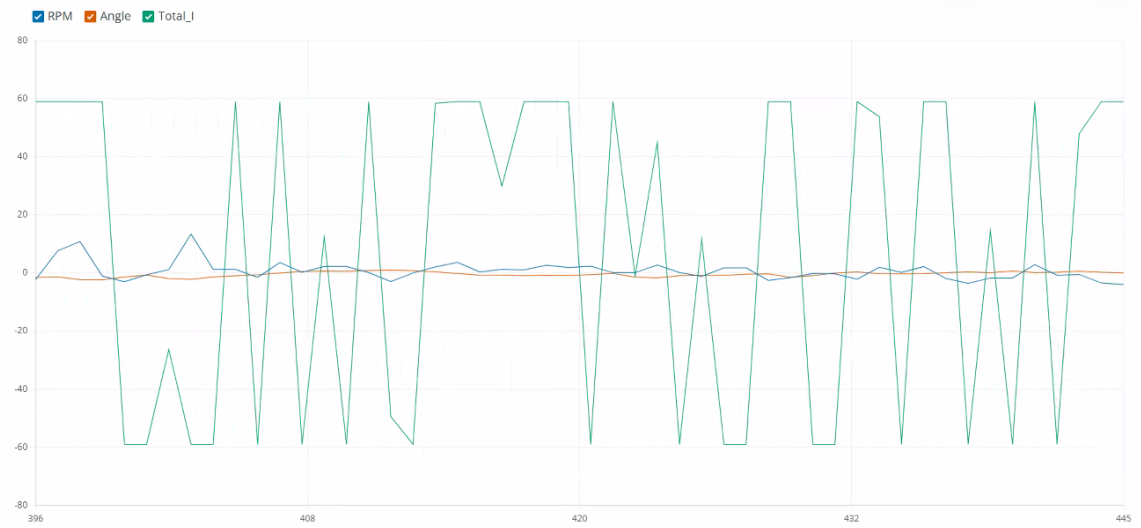


Figure 5.14: Experimental response of the reaction-wheel system with back-EMF current compensation over time.

5.5.4 Experimental Evaluation of PD Controller Tuning

From the experimental results, it was observed that changing the PD gains had a clear effect on the reaction-wheel response. Increasing the proportional gain K_p increased the magnitude of the commanded motor current, which allowed the reaction wheel to generate a larger corrective torque against the bicycle tilt. Therefore, higher K_p values generally improved the strength of the response. However, when K_p became too high, the current command changed more aggressively, which could lead to oscillations and less smooth behavior.

The derivative gain K_d affected how quickly the controller responded to changes in the tilt direction. Increasing K_d improved the reaction of the system to rapid changes by considering the rate of change of the roll error. This helped the reaction wheel respond earlier and reduced the delay between the roll motion and the corrective motor action. However, when K_d is too large, the system can become overly sensitive to rapid signal changes, leading to an unstable or noisy response.

Although the tested K_p values improved the corrective response of the reaction wheel, the generated torque remained insufficient to achieve bicycle balancing. Nevertheless, the experimental results showed that the fastest and most responsive reaction-wheel behavior Figure 5.15 during changes in roll direction was obtained with $K_p = 200$ and $K_d = 25$.

In Figure 5.15, the horizontal axis represents time progression during the experimental test where the sampling index is 10 ms. The vertical axis represents the measured values of the three plotted signals: bicycle roll angle, and total commanded motor current.

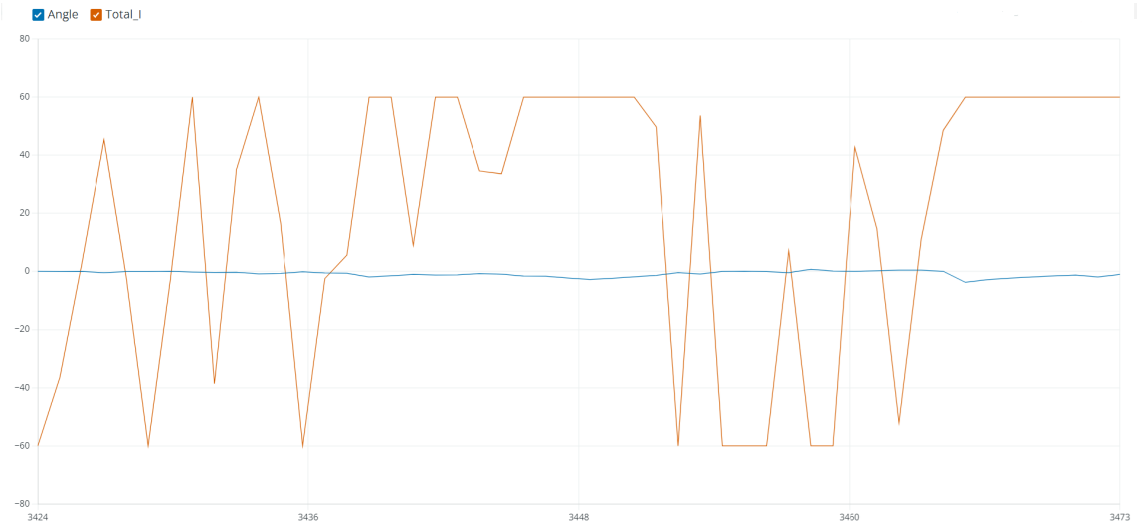


Figure 5.15: Experimental response of the PD controller to the bicycle roll angle, with $K_p = 200$ and $K_d = 25$.

5.5.4.1 System Limitations and Proposed Improvements

Limitation / State: Bicycle

Observed Symptom: Overweighted

Likely Cause:
The bicycle itself was relatively heavy, in addition to the added project components and battery system, resulting in a total mass of approximately 41 kg.

Effect on Balancing:

The increased total mass increased the moment of inertia of the system, requiring a larger corrective torque from the reaction wheel. This reduced the effectiveness of the actuator and increased the response requirements of the control system.

Proposed Improvement:
Reduce the total system weight by using lighter structural materials, a lighter bicycle frame, and more compact electronic and battery components.

Limitation / State: Motor

Observed Symptom:
The motor power was insufficient for the requirements of this project, as even very small roll angles, such as 1°, as shown in Figure 5.16, required the motor to draw the maximum allowable current of 59 A in order to generate the necessary corrective torque. This resulted in a significant increase in motor temperature, which could potentially lead to motor damage or burnout if maintained for extended periods.

Likely Cause:
The selected motor was unable to provide sufficient continuous torque for the balancing requirements of the system.

Effect on Balancing:
The motor was unable to generate sufficient corrective torque to stabilize the bicycle, especially during rapid tilt changes or continuous balancing operation.

Proposed Improvement:
Use a motor with higher torque and power capability, or redesign the reaction wheel system to increase the generated balancing torque.

Limitation / State: Actuator Mount

Observed Symptom:
Relatively strong oscillations

Likely Cause:
Insufficient structural rigidity and mechanical vibrations in the actuator mounting system.

Effect on Balancing:
The oscillations introduced disturbances into the control loop and affected measurement stability, reducing the overall balancing performance.

Proposed Improvement:
Increase the stiffness of the actuator mount and improve the mechanical fixation to

reduce structural vibrations and oscillations.

Limitation / State: Motor Attachment

Observed Symptom:
Poor airflow around the motor resulted in insufficient cooling, which caused the motor attachment to overheat and melt Figure 5.17.

Likely Cause:
The motor mounting configuration restricted airflow around the motor surface, limiting heat dissipation.

Effect on Balancing:
Insufficient cooling increased the motor temperature during operation, reducing efficiency and increasing the risk of overheating during extended balancing attempts.
Proposed Improvement:
Redesign the motor attachment to improve airflow around the motor or add an active cooling system such as cooling fans or heat sinks.

```
Angle:-2.23     Total_I:59.00
Angle:-1.78     Total_I:-59.00
Angle:-1.54     Total_I:59.00
Angle:-1.09     Total_I:-59.00
Angle:-1.04     Total_I:59.00
Angle:-1.33     Total_I:59.00
Angle:-1.83     Total_I:59.00
Angle:-1.80     Total_I:59.00
Angle:-1.74     Total_I:59.00
Angle:-1.89     Total_I:59.00
Angle:-1.30     Total_I:-59.00
```

Figure 5.16: PD controller current response.

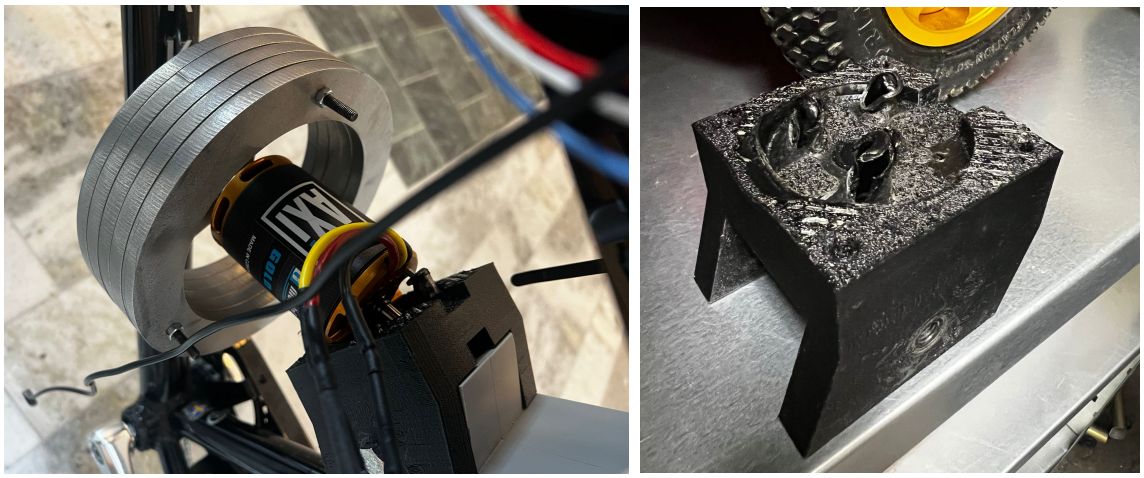


Figure 5.17: Melted motor attachment.

5.5.4.2 Comparison Between Simulation and Hardware

The simulation results indicated that the bicycle could theoretically be stabilized using the designed controller. In MATLAB, the controller was able to reduce the roll angle toward the upright equilibrium. This shows that the control strategy was suitable for the simplified mathematical model.

However, the physical prototype did not fully match the simulated behavior. During hardware testing, the reaction wheel was able to counteract the roll motion, but stable autonomous equilibrium was not consistently achieved. This difference is mainly due to the idealized assumptions used in the simulation compared with the practical limitations present in the physical system.

A major difference was the total system mass. The simulation was based on the original bicycle configuration, with a mass of approximately 23 kg, while the physical tests were performed using a heavier bicycle of approximately 34 kg. This increased the gravitational torque and reduced the maximum recoverable roll angle. Under the same conditions, the simulated system could theoretically recover from approximately 2.35°, while the heavier physical setup reduced this value to approximately 1.8°. In addition, the simulation used a maximum current of 65 A, whereas the physical tests were limited to 59 A for safety and motor protection.

Mechanical effects also contributed to the difference between simulation and hardware. The simulation assumed a rigid actuator connection, while the physical prototype experienced vibrations from the motor, reaction wheel, and plastic mounting structure. These vibrations affected the sensor measurements and the control response, making stabilization more difficult in practice.

Furthermore, the real actuator was limited by heating, back-EMF, current saturation, and response delays. As a result, the motor could not always generate the required stabilizing torque as quickly as assumed in the simulation.

6
Conclusion

This project investigated the feasibility of stabilizing a stationary bicycle using a reaction wheel actuator. The bicycle was modeled as an inverted pendulum, and several control strategies were evaluated through MATLAB and physical testing.

The simulations showed that stabilization is theoretically achievable for the modeled system. PD, PID, and cascade controllers were all capable of stabilizing the bicycle, while the PI controller lacked sufficient damping for stable operation. The developed state-space model also showed good agreement with he nonlinear model for small roll angles, making it suitable for controller design and analysis.

During physical testing, the reaction wheel system was able to counteract roll motion and noticeably improve the balance behavior of the bicycle. However, fully autonomous equilibrium could not consistently be maintained using current hardware configuration. The physical system proved to be highly sensitive to disturbances, delays, vibrations, and actuator limitations, resulting in deviations between simulation and real-world performance. Differences between assumed model parameters and the physical system, such as the total bicycle mass, also contributed to discrepancies between the simulated and experimental behavior.

Despite these limitations, the project achieved significant progress toward a stationary self-balancing bicycle. The actuator system, controller implementation, sensor integration, and embedded control platform were successfully developed and validated. The transition from a myRIO-based platform to an ESP32 also simplified implementation while still providing sufficient real-time performance.

Future work should focus on improving the mechanical rigidity of the system, reducing vibrations, optimizing controller tuning, and replacing the current motor with a more suitable one capable of delivering higher torque and a faster dynamic response. More advanced estimations and control methods could also further improve system stability.

In conclusion, the project demonstrates that reaction wheel stabilization of a stationary bicycle is a promising approach. Although complete autonomous balancing was not fully achieved, the system was capable of approaching stable equilibrium and provides a foundation for continued development within the Autobike project.

References

[1] European Commission, "Its & vulnerable road users," 2024. Accessed: 2026-02-13.

[2] BBC News, "Volvo unveils cyclist alert-and-brake car system," 2013. Accessed: 2026-02-13.

[3] 4activeSystems, "4activebs - euro ncap bicyclist targets," 2026. Accessed: 2026-05-13.

[4] L. Raffaëlli *et al.*, "Facing adas validation complexity with usage oriented testing," *arXiv*, 2016.

[5] U. Erdinc, "Autonomous bicycle control and trajectory tracking," master's thesis, Chalmers University of Technology, Gothenburg, Sweden, 2018.

[6] A. P. Group, "Ssy226 project report," tech. rep., Chalmers University of Technology, Gothenburg, Sweden, 2018.

[7] Unknown, "State estimation for the autobike project," master's thesis, Chalmers University of Technology, Gothenburg, Sweden, 2020.

[8] Eric and Jared, "Reaction wheel stabilization of the autobike," master's thesis, Chalmers University of Technology, Gothenburg, Sweden, 2024.

[9] D. J. N. Limebeer and R. S. Sharp, "Bicycles, motorcycles, and models," *IEEE Control Systems Magazine*, vol. 26, no. 5, pp. 34–61, 2006.

[10] COMSOL, "Inverted pendulum," 2006. Accessed: 2026-03-12.

[11] C. Nordling and J. Österman, *Physics Handbook for Science and Engineering*. 9 ed., 2006.

[12] Johan S, "Modelling of a dc motor," 2017. Accessed: 2026-03-12.

[13] K. Ogata, *Modern Control Engineering*. Pearson, 5 ed., 2010.

[14] MathWorks, "Simulink," 2026. Accessed: 2026-03-11.

[15] NASA, "Reaction wheels," 2026. Accessed: 2026-05-05.

[16] Encyclopaedia Britannica, "Gyroscope," 2026. Accessed: 2026-05-05.

[17] K. J. Åström and T. Hägglund, *PID Controllers: Theory, Design, and Tuning*. Instrument Society of America, 1995.

[18] B. Lennartson, *Reglerteknikens grunder*. Lund, Sweden: Studentlitteratur, 4 ed., 2000.

[19] ScienceDirect, "Cascade control - an overview," 2026. Accessed: 2026-05-05.

[20] Control.com, "Cascade control," 2026. Accessed: 2026-05-05.

[21] MathWorks, "Designing cascade control system with pi controllers," 2026. Accessed: 2026-05-05.

[22] AHRS Documentation, "Complementary filter," 2026. Accessed: 2026-05-05.

[23] Model Motors, "Axi 5360/24hd v3 gold line motor," 2026. Accessed: 2026-03-11.

[24] Digilent, “Pmod nav imu documentation,” 2025. Accessed: 2026-03-12.
[25] Digilent, “Imu reference manual,” 2025. Accessed: 2026-03-12.
[26] Barak Or, “Inertial measurement unit explained,” 2025. Accessed: 2026-03-12.
[27] Trampa Boards, “Vesc 6 hp - high power pcb,” 2026. Accessed: 2026-03-12.
[28] National Instruments, “myrio-1900 user guide and specifications,” 2025. Accessed: 2026-03-12.
[29] MathWorks, “What is matlab?,” 2026. Accessed: 2026-03-11.
[30] B. Hunt, R. Lipsman, and J. Rosenberg, “A guide to matlab,” 2026. Accessed: 2026-03-11.
[31] National Instruments, “Benefits of graphically programming in labview,” 2025. Accessed: 2026-03-11.
[32] VESC Project, “Vesc tool,” 2026. Accessed: 2026-03-12.
[33] R. Acharya, “Special topics,” in *Understanding Satellite Navigation*, pp. 313–350, Elsevier, 2014. Accessed: 2026-03-11.
[34] R. Marchthaler and S. Dingler, *Kalman Filter*. Wiesbaden, Germany: Springer, 2026.
[35] RAD, “Industrial cellular routers,” 2026. Accessed: 2026-03-12.
[36] Induo, “Teltonika rut955,” 2026. Accessed: 2026-03-12.
[37] Elfa Distrelec, “Yw1b-v4e02r emergency stop switch,” 2026. Accessed: 2026-03-12.
[38] Elfa Distrelec, “471001264143 rocker switch,” 2026. Accessed: 2026-03-12.
[39] Gens Ace, “12s 5000mah 60c lipo battery,” 2026. Accessed: 2026-03-12.
[40] Gens Ace, “3s 1800mah lipo battery,” 2026. Accessed: 2026-03-12.
[41] RCMakerLab, “Calculate the maximum current of a lipo battery,” 2026. Accessed: 2026-03-12.
[42] Siemens, “Configuring of cascade control,” 2026. Accessed: 2026-05-05.

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY