A Linear Parameter-Varying Control Method for Inline Wheel Systems

by

Ronald Grant Smith

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Engineering
(Electrical Engineering)
in the University of Michigan-Dearborn
2021

Master's Thesis Committee:

Associate Professor Sridhar Lakshmanan, Committee Chair
Lecturer II Randy Boone
Lecturer I Paul Muench

Ronald Grant Smith

rongrant@umich.edu

ORCID iD: 0000-0002-7711-778X

Acknowledgements

I would like to express my deepest thanks and gratitude to my faculty advisors Paul Muench and Sridhar Lakshmanan. Both have served as great scholarly examples to me and have encouraged, inspired, and led me throughout my time as a graduate student at the University of Michigan-Dearborn. To Sridhar, for his clear direction and oversight over the entire team of faculty and students who have worked on the bicycle stability and control over the past few years, as well as for always challenging me to be my best self and to do my best work. To Paul, for always believing in me. It was his faith that convinced me in the first place that I could balance a bicycle. They have both taught me so much more about life than just bicycle control.

I would also like to acknowledge Alireza Mohammadi, for his hard work and dedication in working with Ziad Fawaz and myself. His training and willingness to work side-by-side with us greatly improved my background and foundation in control theory. His ideas and recommendations are found throughout this work. Especially in the area of linear parameter-varying control.

Lastly, I would like to acknowledge Ziad Fawaz for his dedication and for the true friendship that developed over the course of our time working together on this project. Although we worked individually on our own theses and control analysis, almost all of our experimental testing was accomplished together. Ziad always showed up as a team member. He was willing and eager to help me through each milestone of our research. I will never forget the enthusiasm, laughter, joy, frustration, grit, perseverance, and ultimate satisfaction that we enjoyed while working together.

Table of Contents

List of Tables

List of Figures

List of Appendices

Abstract

The design of the bicycle and other single-track systems are continually evolving and have become a key tool for people and goods transportation worldwide [1],[2]. The form factor, carrying capacity, maneuverability, and cost of single-track vehicles makes them advantageous in a variety of circumstances and justifies their use case in the 21st Century [2] [3],[4]. As autonomous double track vehicles arrive on public roads, it is natural that single-track autonomous systems will be developed as well; however, the unstable and non-minimum phase dynamics of single-track vehicles make their control have an additional layer of complexity compared to double track vehicles. Although many researchers have provided commentary on the stability and tracking of a riderless bicycle, relatively few bodies of work have validated their analysis through experimental testing. This work successfully demonstrates that, through gain scheduling, a PID-type controller can balance a riderless single-track vehicle by using a linear actuator to implement front-fork steering control. This control system is novel in the way in which the front fork is actuated. The manual PID tuning process outlined in this body of work is also unique, as well as the specifics of the control law (although PID controllers have been used by other authors). The works of other authors on this topic is briefly summarized and a second-order dynamics system model is derived. Then controller analysis is simulated and then validated experimentally. Suggestions are also made on next steps that can be taken to build upon the work outlined in this thesis.

Chapter 1 Background and Motivation

1.1 Bicycle History

Bicycles have been a part of society and daily transportation since the 1800's [5]. First proposed by Baron Karl von Drais in 1817, the basic concept of the machine (a rider sitting on top of two wheels, with a steering mechanism) has remained consistent for over two hundred years. Throughout the early and mid-1800's, improvements were made to the bicycle design [6]. These improvements include pedals, gearing, brakes, diamond shaped frames, spring suspension [7], and rubber tires. Since the late 1800's, the core bicycle design has remained surprisingly consistent. In fact, the "safety bicycles" of the late 1800's, would not seem out of place if seen on roads and sidewalks today. Figure 1 shows a bicycle from 1885, Figure 2 shows a modern women's bicycle. The general shape and architecture of the bicycle shows little difference.

Figure 1: JK Starley safety bicycle (1885) [8].

Figure 2: A modern bicycle [9].

While the basic design of the bicycle has not changed over the last one hundred years, it is not due to a lack of imagination. Many design alternatives have been developed including tricycles, quadracycles, and recumbent bicycles. The bicycle has also been specialized for different applications (on-road touring, off-road trails, short distance sprints, long distance travel, and even in-home fitness) [10]. Also, introductions of new mechanical propulsion technologies have been applied to the bicycle design. In modern day, it is possible to purchase bicycles that are powered by electric motors, gas motors, or that are exclusively human powered. Additionally, powered propulsion has led to design derivatives of the bicycle that include the motorcycle, dirt bike, and scooter (or "moped").

1.2 The Bicycle Architecture

What has allowed the bicycle to maintain its popularity for so long? Bicycles are used throughout the world for daily transport as well as for recreation. Comparing it to other forms of ground transportation (carts, automobiles, trains, buses, etc.), there are a few attributes that stand out:

- Form factor
- Specific power and energy efficiency
- Terrain capability
- Maneuverability
- Cost

Form Factor. In general terms, a bicycle occupies most of its space in 2-dimensions. It is a long and slender object with its height and length being much greater than its width (unlike carts, buses, trains, and automobiles which span in all 3-dimensions). The form factor of the bicycle makes it easy to store, takes up less road space, and enables it to enter tighter pathways such as sidewalks, narrow corridors, mountain trails, etc. At the same time, the tall and slender

design of the bicycle makes it inherently less stable. In fact, most bicycles are unstable when under motion and rely on the skill of the rider to maintain themselves upright and balanced. Despite these limitations, the form factor of a bicycle makes it an attractive solution when a smaller vehicle footprint is desired. In general, the smaller form factor of a 2-wheeled system also makes it a lightweight travel solution.

Specific power and energy efficiency. Most bicycle frames are made of metal or composite tubing arranged in a triangular truss structure. The shape makes the structure particularly strong and capable of supporting many times its own weight in payload. Additionally, since a bicycle has less wheels and moving parts than a 4-wheeled vehicle, the internal friction in the system is far lower. The mechanical gearing enables an efficient method of applying power to the wheels. Additionally, the slender cross section leads minimizes air drag when travelling at speed. The combination of these features makes the bicycle a much lower effort machine than 4-wheeled vehicles.

Terrain capability. In-line wheel systems offer good terrain versatility (when properly equipped) but are not considered as robust as 4-wheeled vehicles. Bicycles and motorcycles are regularly operated off-road on dirt, mud, snow, and ice surfaces for recreational purposes. Their carrying capacity and stability are limited on low-μ (coefficient of friction) surfaces. However, their smaller cross section can be an advantage when travelling through particularly difficult terrains such as mountainsides and forests [11].

Maneuverability. Bicycles and motorcycles are well known for their ability to handle tight turning tracks. As a tricycle or 4-wheeled vehicle negotiates a turn, the inside and outside wheels must travel at different speeds (to avoid skidding) since they are operating on different turn radii, but at the same angular velocity. 4-wheeled vehicle systems often require complex

steering mechanisms and axle differentials to negotiate tight turns, adding weight and costs to the vehicle. In-line wheel systems have simple (and lighter) steering systems. Combining the lean of the rider as well as changing the steering angle can allow for tight turning [12]. As previously mentioned, the smaller frontal cross section also allows in-line vehicles to maneuver through narrow corridors.

Cost. Because of its size, bicycle components are small, and relatively cheap compared to other forms of transportation, making it an efficient transport option from a budget standpoint.

1.3 Future Bicycle Applications: Autonomous Bicycles

The bicycle has been an integral transportation vehicle for the last 100 years, but will that change in the future? The transportation industry is expected to go through a major disruption cycle over the next 30 years, now referred to as Auto 2.0. Improvements in battery technology, the Internet of things, modern control theory, and autonomous systems are the driving factors behind Auto 2.0. While much research and development are focused on automobiles, trucks, and buses, little research has been focused on in-line wheel systems (bicycles and motorcycles) at this time. However, just as 4-wheeled vehicles have shared the road with 2-wheeled systems for the last 100 years, it is fair to assume the bicycle will continue to exist alongside its larger, and more costly, 4-wheeled counterparts in the future as well. In fact, an autonomous in-line wheel system is arguably just as feasible as a 4-wheeled system.

The smaller size and weight of a 2-wheeled system intrinsically offers more safety than a larger 4-wheeled system. When operating in proximity to humans (at low speeds), the worst-case scenario is probably severe-injury, not death.

1.4 Research Emphasis – Single Track Stability Control

The purpose of this thesis is not to present an exhaustive list of applications, or a business justification, of an autonomous bicycle system. Rather, it is to focus on how an autonomous bicycle system can operate independent of a human, by taking on the tasks generally performed by the rider. Specifically, how a bicycle may balance and maneuver itself.

A lot of research and development is underway for vision, obstacle detection, and path planning for autonomous vehicles. While there are several different approaches, it is assumed that the vision, sensing system, and path planning algorithms developed for 4-wheeled vehicles could be adapted to an in-line wheel system as well [13], [14].

Unique to an in-line wheel system is the way in which the vehicle will balance, lean, and steer. Later sections of this work will deal with how an in-line system may be designed for autonomous operation. A balance and steering controls system will be presented along with the required sensors and actuators required for its operation Chapter 2. A controller will be developed using linear parameter varying (LPV) control Chapter 3. Experimental results with the controller will be reported and analyzed to demonstrate the proof-of-concept Chapter 4. The thesis will conclude with closing remarks and recommendations for future work on the subject.

Chapter 2 The Autonomous Bicycle System

2.1 Autonomous System Architecture

As mentioned in Chapter 1, a fully autonomous bicycle must be able to perform the actions generally performed by the human rider. This includes the following:

- Path planning for most optimal route of travel,
- Identify, track, and avoid obstacles and other vehicles,
- Communicate the vehicle's intended path to other drivers (example: signal a turn),
- Maintain balance (or equilibrium state) of the bicycle, and
- Propel the bicycle forward.

In addition to the operations performed by a rider, AV's must be able to regularly communicate with external communication systems including GPS, global path planning guidance systems, and vehicle OEM (Original Equipment Manufacturer) data clouds. These additional communications are necessary to support successful AV operation [14]. As Pendleton states [13], most aspects of the AV system architecture (planning, perception, and external communication) are not vehicle specific and can be applied across vehicle types. Behere recommends a functional architecture for AV's that separates the manipulation of the vehicle platform from a "Cognitive Driving Intelligence" which handles the tasks of planning and perception [14]. A similar architecture is outlined in this thesis which separates the functional control of the autonomous inline wheel system into two subsystems: The Bicycle Control Platform, and the Intelligent Driver (ID) system. Figure 3 outlines the proposed system architecture for an autonomous vehicle bicycle system and outlines the tasks that would be performed by the Bicycle Control Platform and the Intelligent Driver System.

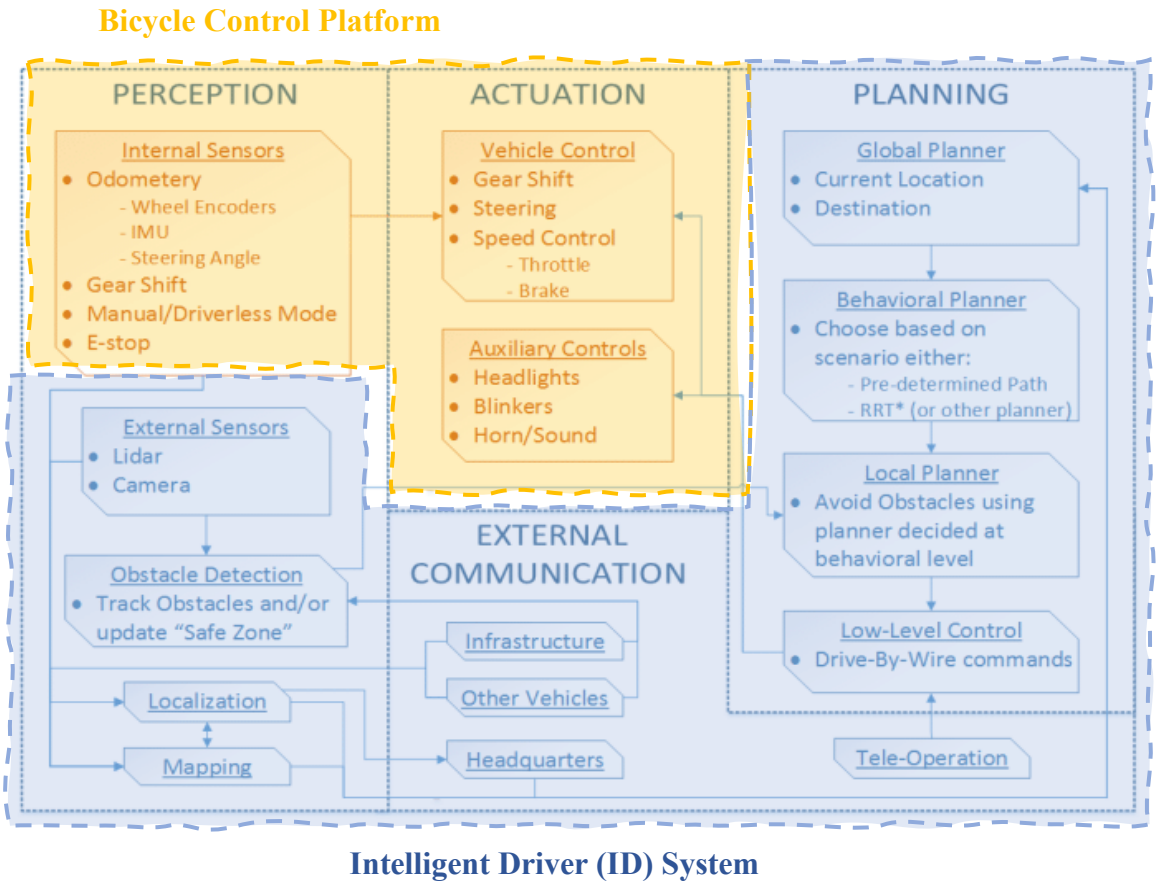

Figure 3: System architecture for autonomous vehicles [7].

When considering an autonomous bicycle system, a vehicle platform must be developed that can interact with the ID system and external communication sources. For the purposes of this study, it is assumed that all external communications will be controlled by the Intelligent Driver System. The interaction between the Intelligent Driver System and Bicycle Control Platform is defined in Figure 4.

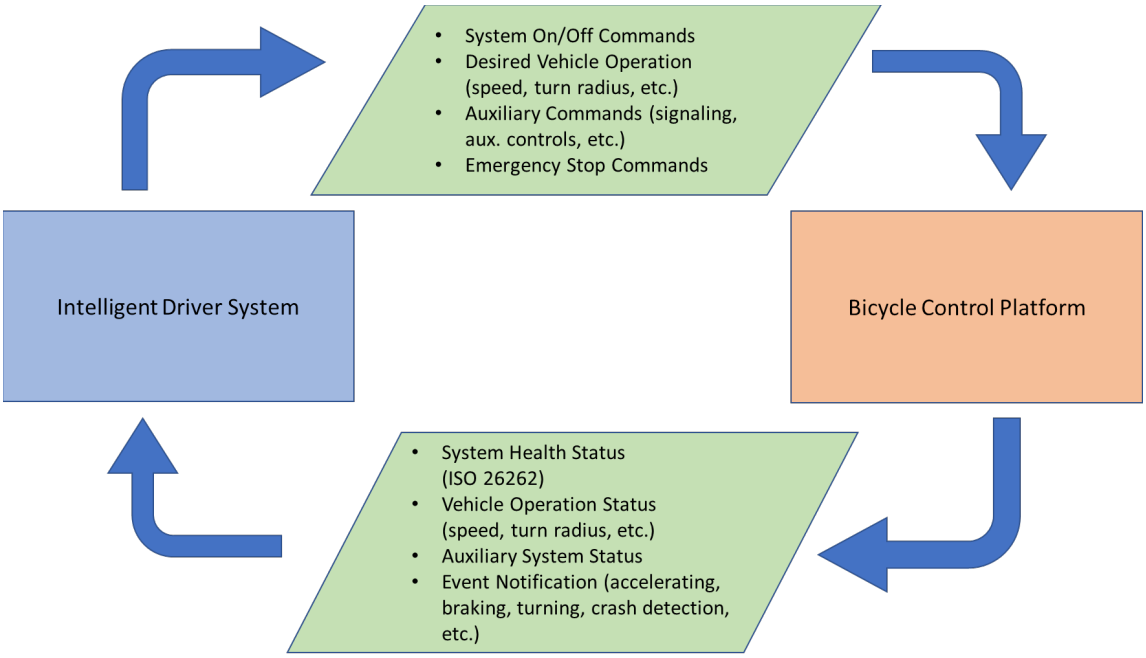

Figure 4: Driver intelligence and bicycle control platform systems interactions.

The ID system will send information to the bicycle control platform informing how the bicycle is to maneuver. This information would include a desired speed of travel, turn radius, rate of acceleration/deceleration, etc. Additional commands (such as signaling, horn requests, and powering auxiliary systems) may also be sent by the ID system. The intelligent driver system would also be capable of sending power on/off commands to the bicycle control platform, as well as emergency stop commands.

The bicycle control platform will interpret the commands from the intelligent driver system and act accordingly. The bicycle control platform will report to the ID system the vehicle operation status and auxiliary systems status. The bicycle control platform may also report system faults and event notifications to the ID system, thereby improving functional safety [15], [16].

2.2 The Bicycle Control Platform

The bicycle control platform's primary objectives are to balance, steer, and propel the vehicle. As shown in Figure 4 and in [13], the vehicle control platform receives high level commands from the ID system and must determine for itself how to implement the commands. It is therefore necessary that the bicycle platform has its own controller. Figure 5 outlines the subsystems of the bicycle control platform.

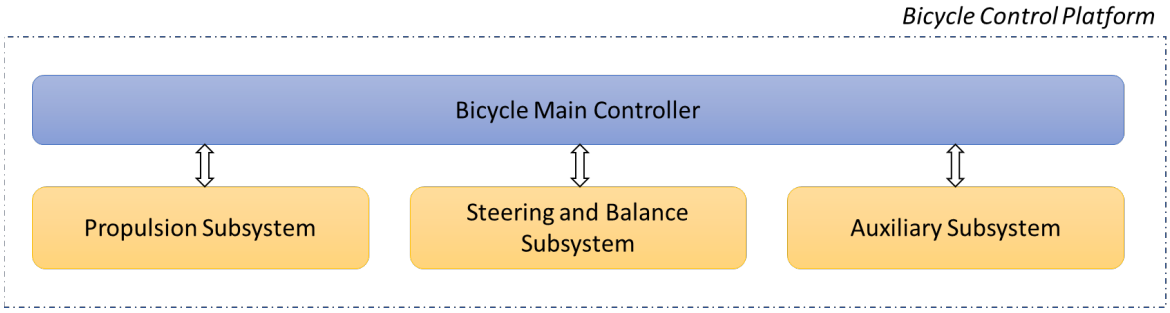

Figure 5: Bicycle control platform system diagram.

The subsystems of the bicycle control platform the following functions:

- Bicycle Controller – receives commands from the ID and sends commands to the subsystem actuators.
- Propulsion Subsystem – propels (or stops) the vehicle's forward motion.
- Steering and Balance Subsystem – balances and steers the bicycle.
- Auxiliary Subsystem – supports auxiliary actions (signaling, for example).

2.2.1 Steering and Balance Subsystem

It is important to note prior work on the development of autonomous bicycle systems. There are several works reported in literature of riderless, or robotically balanced, bicycles. Much of the work is focused on stability analysis, though some works have included the task of path tracking [17]. Figure 6, provided by Arend Schwab [18], gives a general overview of works performed on the subject:

Control outputs	Author	Stabilising task	Tracking task	Control inputs	Successful
Steer torque	Ruijs and Pacejka [71]*	ϕ		$\dot{\delta}, \dot{\phi}, \phi$	Yes
	Saguchi et al. [118,119]	ϕ		$\dot{\delta}, \delta, \dot{\phi}, \phi, \alpha, \gamma$	Yes
	Michini and Torrez [120]	ϕ		$\dot{\phi}, \phi$	Yes
	Andreo et al. [121,122]	ϕ		$\dot{\phi}, \delta$	Yes
Steer torque and moving mass torque	Iuchi et al. [123]	ϕ	ϕ	$\delta, \phi, \dot{\phi}, \beta$	Yes
Steer torque and gyroscope	Levandowski et al. [75,76]*	ϕ	y	Unclear	Yes
Steer angle	Miyagishi et al. [124]*	ϕ	y	$\phi, \dot{\phi}, \delta, \gamma$	Yes
	Tanaka and Murakami [77]	ϕ	y, γ, k	$\phi, y, \gamma, k, \delta$	Yes
	Yamaguchi [125]	ϕ		$\phi, \dot{\phi}, \ddot{\phi}$	Yes
Steer torque and angle	Lenkeit [126]*	ϕ		δ or τ_δ	Unclear
Steer angle and moving mass position	Nagai [74]	ϕ	y	y, δ, ϕ, β	Yes
Moving mass torque	van Zytveld [70]	ϕ		$\phi, \dot{\phi}, \beta, \dot{\beta}$	No
Moving mass position	Yamakita et al. [127–130]	ϕ, β	y	δ, ϕ, β, y	Yes
Gyroscope orientation	Thanh and Parnichkun [86]	ϕ		ϕ, φ	Yes

Notes: A star (*) indicates the control is for a motorcycle. ϕ is roll angle, δ is steer angle, α is slip angle, γ is yaw angle, y is lateral position, k is the curvature of the path, τ_δ is steer toque. β is the lean angle of a moving mass and φ is the orientation of the gyroscope.

Figure 6: Summary of research on bicycle stability and path tracking [18].

There are three recognized methods for balancing and steering the bicycle. All have been demonstrated experimentally (at least in a limited fashion):

- Actuating a front fork,
- Applying a leaning mass, and
- Using a gyroscopic force (balance only).

Front wheel steering: Front wheel steering is the most validated method for balance and stability. Miyagishi explains that the steer torque provides a larger input when compared to posture (lean) control [19]. Using the steering angle of the bicycle as a control input for balance is also beneficial for path tracking. It will be shown in Chapter 3 that a direct relationship can be made between the steering angle and the path of the bicycle during a steady turn. All the authors identified by Schwab were able to use steering actuation to balance the bicycle (or motorcycle), though their approaches often differed [20].

Many authors have used steering torque as their control input to achieve balance control. Pacejka balanced a motorcycle by using steer torque control to set the steering and roll angles.

By setting the roll angle, they also demonstrated that motorcycle was able to follow a path. Saguchi employed a robotic machine as a rider on a bicycle. The rider robot utilized a steering torque control to stabilize the bicycle along a straight line as well as a constant curve. Michini utilized a DC drive motor connected to the front steering fork. Others have also noted success utilizing DC and servo motors as control actuators. Roll rates were measured with gyroscopes and accelerometers, and the steering angle was measured in many cases using a radial encoder attached to the front fork. [18], [21], [22].

Using the steering angle as the stability command input has also been trialed successfully. However, Michini notes steer torque as a preferred approach by citing other authors [22]. Other authors who used the steering angle as the command input developed a hybrid approach. Lenkeit even went as far as applying both methodologies, using the steering angle as a command input for lower speeds, and steering torque as the command input for higher speeds [23].

Leaning mass: First attempted by Van Zytveld [18],[24], using a leaning mass has been successful at keeping a bicycle upright. Yamakita has been able to achieve stability (using numerical simulations) by treating the bicycle as an inverted pendulum [25]. The leaning mass is actuated on a lever arm to counteract gravity pulling the bicycle to the ground. Leaning mass is considered higher effort than using the front fork steering to stabilize the bicycle. It appears primarily useful when the bicycle is stationary, or at very low speeds. However, once the bicycle begins to move with some speed, treating the bicycle as an inverted pendulum become less accurate because of the bicycle's ability to steer [18].

Gyroscopic force: Some projects have used gyroscopic effects to stabilize a bicycle. By accelerated/decelerating a gyroscopic mass, a counter torque is applied to the bicycle to maintain its upright stability. Like the leaning mass method, studies have been limited to the low-speed

ranges [18]. It will not be considered further for this project, as it is thought that a high amount of power would be required to maintain a gyroscopic mass at speed. From an application perspective, a gyroscope presents a significant packaging challenge, as any spinning mass would need to be shielded or positioned in a way such that is protected from damage or pose a safety risk to its environment (in the case of a crash). Table 1 provides a comparison of each bicycle stability control method.

Table 1: A comparison of each bicycle stability control method.

Balance Method	Advantages	Disadvantages	Path Tracking Capability
Front fork actuation	Lower effort Direct linkage to path tracking	No stationary control	Demonstrated
Leaning mass	Simple for stability control Stationary control	Higher effort Difficult to perform tight turns	Demonstrated
Gyroscopic force	Stationary control	Highest effort Packaging challenges	Not demonstrated

Of the methods listed above, using a form of front fork steering control shows the most promise for an autonomous bicycle system. The steering fork can provide a large enough input to stabilize the bicycle across most speed ranges. It also provides more options for successful path tracking since the steering angle of the bicycle can be measured. However, since steering is ineffective when the bicycle is stationary, some other feature or control methodology would be required to stabilize the bicycle as it comes to a stop. Figure 7 outlines the steering and balance subsystem diagram.

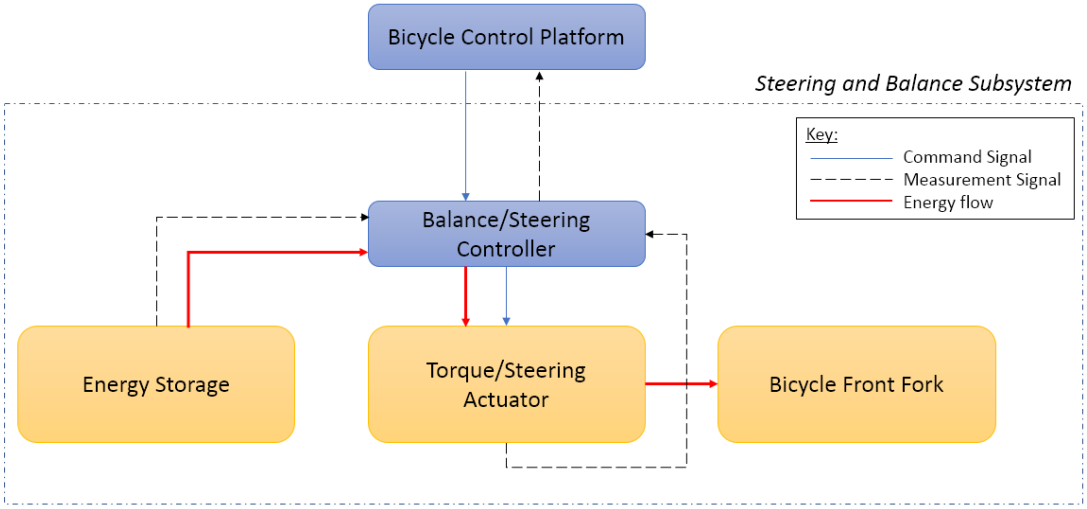

Figure 7: Steering and balance subsystem diagram.

Hybrid Control Methodologies: Nagai was an early author to propose a combined control system for both stabilizing and path tracking simultaneously. Nagai compared a control model using exclusively front fork actuation to a model using a leaning mass/front fork actuating combination. Nagai observed that time required to execute a lane change was improved using the two-input approach [17]. In a more recent work, Yamakita also concludes that a two-input approach (using both the leaning mass and steering handlebar in a closed-loop control) proves more capable of balancing a bicycle system than using a leaning mass by itself [26].

2.2.1.1 Steering Actuation – Linear Actuation

As noted previously, many others have studied the application of balancing a bicycle by using a front wheel steering system (refer to Section 2.2.1). A preferred approach has been to use a steering actuation of the front handlebar, with many opting for a torque-controlled steering approach [20]–[22], [27], [28]. In this body of work, a similar approach is proposed; however, a novelty of the work is the way in which the torque is applied.

Many authors have used servo motors to apply a steering torque [27], [28]. Others have installed a DC motor into the front steering fork to actuate the system [22]. Both methods have their own inherent strengths and weaknesses. Using a servo motor provides precise steering fork control and a reasonable actuating speed. A servo motor contains a closed-loop feedback controller which allows it to react to commands precisely and quickly. The internal gearing of a servo motor allows for some torque multiplication. An added benefit of using a servo motor is that the position (angle) of the motor is always known, which is also necessary when trying to steer a vehicle. Servo motors can be expensive, especially in applications where high torque or dynamic loading is required. It is important that the gearing of the servo motor can withstand any shock or torque feedback in the system. A robust servo motor may require expensive and high-quality components to meet durability requirements. A simple DC motor mounted axially with the front fork is simple and directly accomplishes the task of providing a steering torque to the bicycle. One tradeoff with the simplicity of an using a DC directly integrated into the front for is that there is no torque multiplication; therefore, a larger motor is required. Additionally, a DC motor does not provide any feedback to the steering angle of the bicycle. It will be shown in Chapter 4 that knowledge of the steering angle is not required to balance an in-line wheel system; however, the steering angle is directly related to the path of the vehicle. Therefore, some authors in other works that have used a DC motor have also included a separate encoder to know the steering angle of the bicycle [22].

In this thesis, and its predecessor from the same research group, a new approach for applying torque actuation is proposed [24]. Instead of using a coaxial motor to apply a steering torque to the front fork of the bicycle, the team proposed a more human-centric approach with the use of a linear force applied to the handlebar, the same way a human would rotate the front

fork on a bicycle or motorcycle. Although a fully autonomous system need not take its design queues from a human-operated system (in many ways, efficiencies are gained by excluding the human operator from the design process), thinking about the way a human operates a bicycle led us to identify some distinct characteristics about the linear application of force to a handlebar that are desirable:

1) Significant torque multiplication,
2) Good back-drivability characteristics, and
3) Form factor.

Significant torque multiplication: A linear force applied to the handlebar, off the center axis of the steering fork rotation, provides a torque multiplication factor proportional to the distance from the center axis to point of application. Given that linear actuators are already capable of a high linear force output, it is possible to utilize a smaller actuator (or one that is tuned for high speed/low torque applications) and benefit from using the handlebar as a lever arm.

Good back-drivability characteristics. Various linear actuation methods exist and while some linear electric motors use direct linear motion generators [29], most industrial applications use rotary DC motors coupled with a gearing system. The mechanical gearing mechanisms produce high friction and inertia that must be overcome to move the actuators, limiting the back-drivability of the actuating system. Most motor specifications list a stall torque that must be overcome to move the motor.

The desired level of back-drivability in an actuator depends greatly on the application. In the field of robotics, actuators with high back-drivability characteristics are being developed [30]–[35]. For robotic systems that interact with or near humans, having a low force back-drivable system minimizes the risk of injury. In other applications, the need for higher

controllability, and better dynamics necessitates a more back-drivable system. In Na's design proposal for a cheetazoid leg actuator [33], he laments the use of gearing used in other robotic systems stating that "the nonback-drivable actuation system limits the maximum speed of the robotic joint, which results in a slow response time and a poor controllability". In contrast, linear actuators for machining, load lifting, and load articulation, are required by their very nature to be nonback-drivable for safety reasons.

For the purpose of actuating the front fork of a single-track vehicle, the actuator must be able to absorb any shock forces from the road surface, it should also be able to withstand any impact forces should the vehicle be knocked over; therefore, a back-drivable actuator is highly desirable as it will improve the durability of the system [36], [37]. This is especially true if the single-track vehicle is designed for operating on off-road surfaces where large objects and uneven surfaces will be encountered. For semi-autonomous or "optionally-manned" platforms, a highly back-drivable motor could be selected to allow for a low effort handlebar actuation when operated by a human. A good example already existing in a commercial application is trunk doors lid actuators, which are both manually and electrically actuated.

In summary, both highly back drivable and non-back drivable linear actuators are available. The amount of back drivability required in the linear actuator can be tuned for each application but may have tradeoffs with other design parameters [38], [39].

Form Factor. Compared to a box-shaped or wide-cylindrical rotary motor, a linear actuator has a long slender tube design. As noted in [40], the long tubular design of the linear actuator may have a packaging advantage over a rotary motor. In the case of a bicycle or motorcycle, the linear actuator is packaged in the same area as the top tube (or cross bar) of a traditional male diamond shaped bicycle frame; it could therefore be packaged with minimal

changes to the current architecture. Whether or not the form factor of a linear actuator is advantageous over a rotary motor will depend on the specific application and design of the autonomous single-track vehicle. Table 2 provides a qualitative comparison of the front fork steering actuation methods discussed.

Table 2: Front fork steering actuation methods.

	Torque Multiplication	Back-drivability	Form Factor
DC Motor	-	+	-
Servo with Gear Reduction	+	-	-
Linear Actuator	++	+	-

2.2.2 Propulsion Subsystem

As referenced in Chapter 1, powered propulsion systems for single track systems are not a new development and a variety of applications are available for consumer and commercial use. Motorcycles and mopeds have propulsion systems integrated into the vehicle architecture. Some bicycles are also sold with propulsion (or propulsion assistance) motors; there are also aftermarket kits available [41]–[43]. For example, recent improvements to lithium-ion power density and smaller packaged motors have led to the innovation of electric mountain bicycles known as e-bikes [42].

Single track propulsion is available in both gas and electric powertrains. While they both may differ in terms of the components used, they are architecturally similar in that they both include an energy storage device (a battery or fuel tank) and a motor that converts electric or chemical energy into mechanical power.

The propulsion subsystem is outlined in Figure 8.

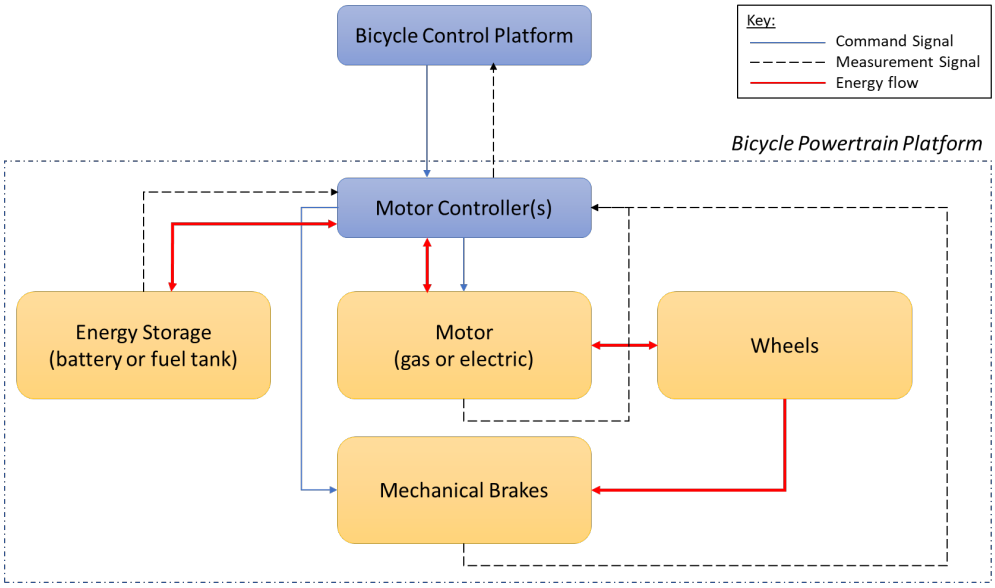

Figure 8: Bicycle propulsion subsystem block diagram.

The main bicycle controller takes the speed and acceleration commands from the ID system and sends the appropriate control commands to the motor controller to actuate the motor and drive the wheels. Motor controllers contain the electronic components that would actuate the motors and mechanical brakes to provide torque and braking power to the wheels. If the vehicle utilized an electromechanical powertrain, it would also be possible to capture energy from the wheels through regenerative braking [44].

2.2.3 Auxiliary Subsystem

An auxiliary subsystem would also be required for interacting with any sensors or actuators associated with the bicycle that are not specific to the IDS, propulsion, or steering and balance systems. This would include any system set up for external communication (turn signals, brake lights, etc.), as well as any systems that may be application specific. It could also include systems for diagnostics and data storage.

2.3 Prototype Bicycle System

A prototype bicycle was constructed for this thesis with the purpose of proving out the conceptual design on an autonomous bicycle system, and for development of a steering and balance control system [40]. The initial iteration of the self-balancing bicycle system was built as part of a class (ECE 560) project. After initial trials and testing, further modifications were made as part of this thesis to power the bicycle. A brief system overview is provided below.

2.3.1 System Overview

Based on the encouragement from Paul Muench, the instructor for ECE 560, to try something new, it was determined early on to use a linear actuator for steering and balance control. A 26" Mongoose Estate mountain bike was chosen as the chassis for the bicycle system. The absence of a top tube was favorable for packaging the linear actuator. The bicycle was also cheap and locally available.

An MPC LAD-HS10 high speed linear actuator was attached from the top of the seat tube to the handlebar for steering actuation of the bicycle. The linear actuator was linked up to an Arduino microcontroller that monitors inputs from an inertial measurement unit (IMU). The IMU reports the roll angle and roll rate of the bicycle to the Arduino microcontroller. The Arduino then actuates the linear actuator to rotate the handlebar to counter steer against any rolling action to maintain the bicycle in an upright position. Figure 9 shows the layout of the prototype bicycle system.

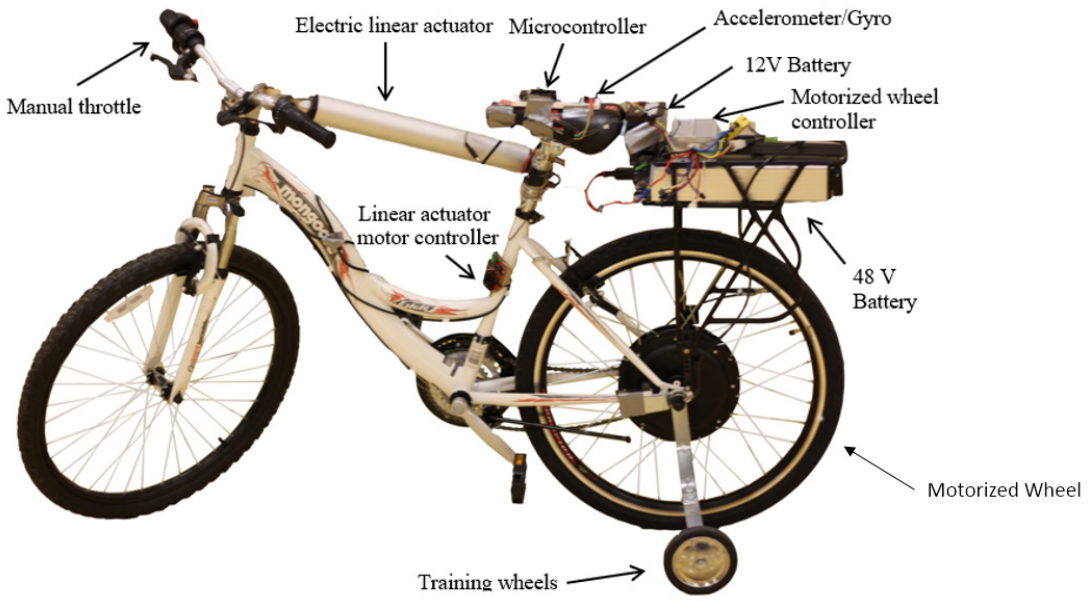

Figure 9: Prototype bicycle system.

A bicycle electric motor conversion kit was purchased with an in-hub electric motor to add propulsion to the bicycle. A rack was mounted above the rear wheel which housed the required battery packs and many of the electronics components. The conversion kit came with a twist throttle for control. Modifications were made to the wiring to control the electric motor from a microcontroller. Further details about how bicycle was actuated and controlled are described in the sections below. Table 3 describes some of the components of the bicycle system and their specifications. Table 4 summarizes the attributes of the entire bicycle system.

Table 3: Bicycle system component summary.

Component	Specifications
Expert Power EXP1245 12-Volt Battery	Capacity: 4.5 Ah
48-Volt Battery	Capacity: 15 Ah Peak Discharge Rate: 2.5 A
Linear Actuator	Stroke: 10 inches Linear Force: 225 lbf. Actuating Speed: 65 mm/sec
Jaxpety Electric Motor	3-phase A/C synchronous motor Power Limit: 1 kW
MPU 6050 3-axis Gyroscope/Accelerometer	I2C Protocol
Arduino Microcontroller	Pinout Voltage: 5 Volts
DRI004 Motor Driver	Input Voltage Range: 7-24 Volts Peak Current Rating: 50 Amps
Men's Bicycle	Size: Medium Wheelbase: 42.25 inches

Table 4: Bicycle system attribute summary.

Attribute	Specification	Value
Mass	Total System Weight	75 lbs.
	Instrumentation Weight	44 lbs.
Center of Gravity	Front/Rear Weight Distribution	33/67
	CG Height	25.5" (approx.)
Propulsion Performance	Maximum Speed	25mph
	Electric Range	30+ miles
Steering System	Maximum Steering Rate	30 degrees/second

2.3.2 Prototype Propulsion System Design

A 48 Volt Jaxpety Rear Wheel Conversion Kit was purchased for propulsion, and it features an in-line hub motor designed for installation on the rear wheel. It also included a motor

controller box, brake handles, and an electric throttle. Specifically, the Jaxpety kit was selected because it was size compatible with the selected bicycle (it utilized a 26” diameter wheel), and because of its affordable price. A 48-volt bicycle battery was also purchased, with a bicycle rack that mounts vertically above the rear wheel axle.

The Jaxpety Kit included a 3-phase 1000-Watt brushless electric motor, capable of reaching speeds of 28 miles per hour [45]. The wires from the motor controller to the throttle were routed through a wiring breadboard for signal identification. The voltage signal from each wire was monitored during operation of the motor and manual throttle so that the power and signal wires could be separated.

After each wire was identified, the wires from the motor controller were routed to an Arduino microcontroller. The throttle control signal wire was connected to an Arduino pulse width modulation (PWM) port. By modulating the PWM duty cycle, the motor speed could be controlled. A bench test with the powered wheel raised off the ground demonstrated the ability to control the motor speed. Figure 10 shows an image of the test setup.

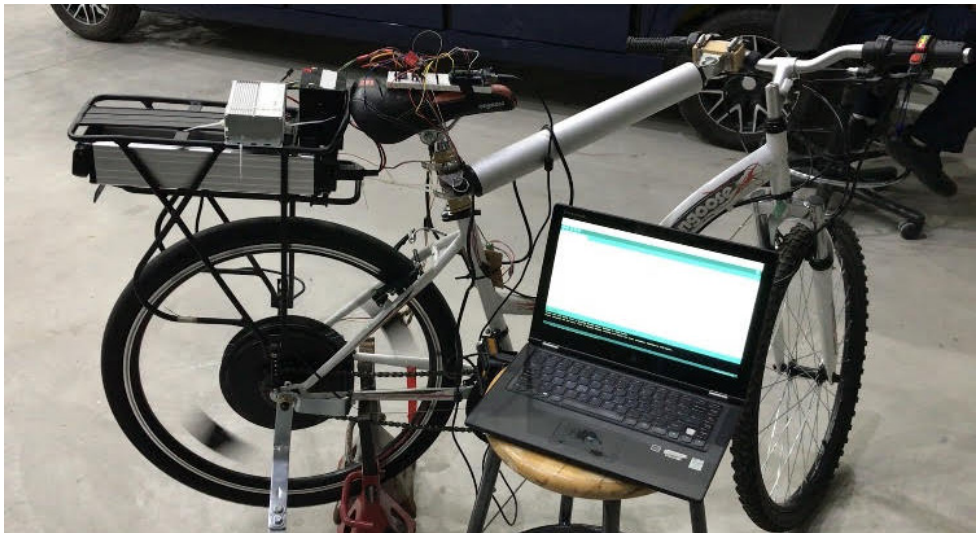

Figure 10: Propulsion system test with the rear wheel raised off the ground.

A hall effect sensor was mounted on the rear of the frame of the bicycle (on one of the chain stays), and a magnet was attached to the spokes. As the magnet passes by the hall effect sensor, a signal was sent to the microcontroller to count each rotation. The microcontroller counted the elapsed time between each wheel rotation to calculate the speed at which the wheel was spinning. Figure 11 graphs the vehicle speed achieved during the benchtop testing compared to the PWM signal input into the motor controller.

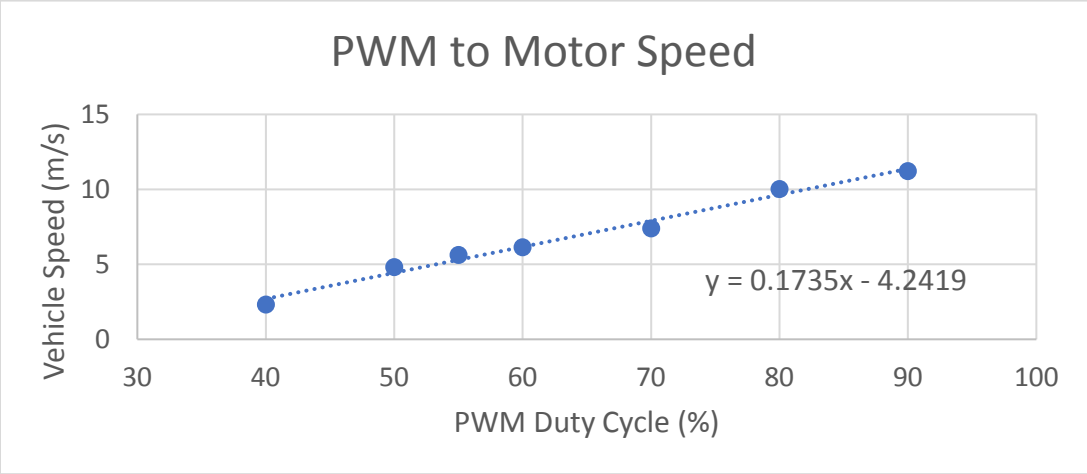

Figure 11: Motor PWM signal mapping to wheel speed.

Once the mapping of the wheel speed to PWM duty cycle was complete, it was possible to use an open-loop controller to control the speed of bicycle. Prior to connecting the throttle signal wire to the Arduino PWM port, some manual testing was performed with constant voltage signal in series with a potentiometer. It was noted that the motor controller had a non-linear response to quick adjustments of the potentiometer. If the throttle command were increased, the motor controller would quickly spool up the motor to the higher speed, but sometimes it would overshoot, and then settle back to the commanded speed. We therefore understand that the throttle signal from the Arduino to the motor controller is commanding a set angular speed for the motor, rather than simply controlling the current to the AC motor.

2.3.3 Bicycle Balance Subsystem Design

An MPC LAD-HS10 high speed linear actuator was attached from the top of the seat tube to the handlebar for steering actuation of the bicycle. On the seat tube, the linear actuator was mounted slightly offset to the right-hand side of the bicycle. On the handlebar, the actuator was attached on the left-hand side, close to the handle grip. This served two purposes: 1) it helped center the weight of the linear actuator over the center of the bicycle, which improves the balance of the bicycle and 2) with the length of the actuator selected, crossing the linear actuator over the center axis of the bicycle helped to balance the range of travel of the actuator such that it was capable of turning the handlebar the same distance in either the clockwise or counter-clockwise directions.

The MPC LAD series linear actuator came in two variants. Like most linear actuators, the standard model featured a higher torque capability [46]; however, MPC also offers a high-speed actuating version of the linear actuator [47]. The high-speed actuation comes with a big trade-off in terms of reduced dynamic load capability. However, the higher speed capability was deemed necessary for the application of actuating a steering fork, and hence chosen. Michini and Torrez used a motor with a 1.4 Nm torque capability, which also justifies the priority of actuating speed over power [22]. Table 5 shows a comparison of the actuating speed and load capabilities of the two MPC actuator variants.

Table 5: Linear actuator specification comparison.

	MPC LAD-HD10 (High Torque Model)	MPC LAD-HS10 (High-Speed Model)
Dynamic Load Capability	385 lb.	45 lb.
Actuation Speed	0.8 cm/sec (unloaded)	6.5 cm/sec (unloaded)

The linear actuator functions on a 12-volt system. A separate small 12-volt motorcycle battery was mounted behind the rear seat to supply power to the linear actuator. Like the rear hub electric motor, the linear actuator was connected to one of the Arduino microcontroller's PWM ports via a DC motor controller. The motor controller is required to step up the 5 Volt signal from the Arduino's PWM port. It also connects to the battery to provide the current necessary to power the actuator. Figure 12 displays the linear actuator mounted to the bicycle.

Figure 12: Arduino microcontroller wired to linear actuator.

Before installing the linear actuator to the bicycle, tests were performed with the linear actuator to verify actuating speed. With the linear actuator unloaded and fixed on a table, a pulse width modulated (PWM) signal was sent from the linear actuator to make it actuate at various speeds for a given time. The end of the linear actuator was marked to record the initial and final extension length of the actuator, and the extension rate was calculated. Figure 13 shows the measured actuation response to the PWM signal from the Arduino.

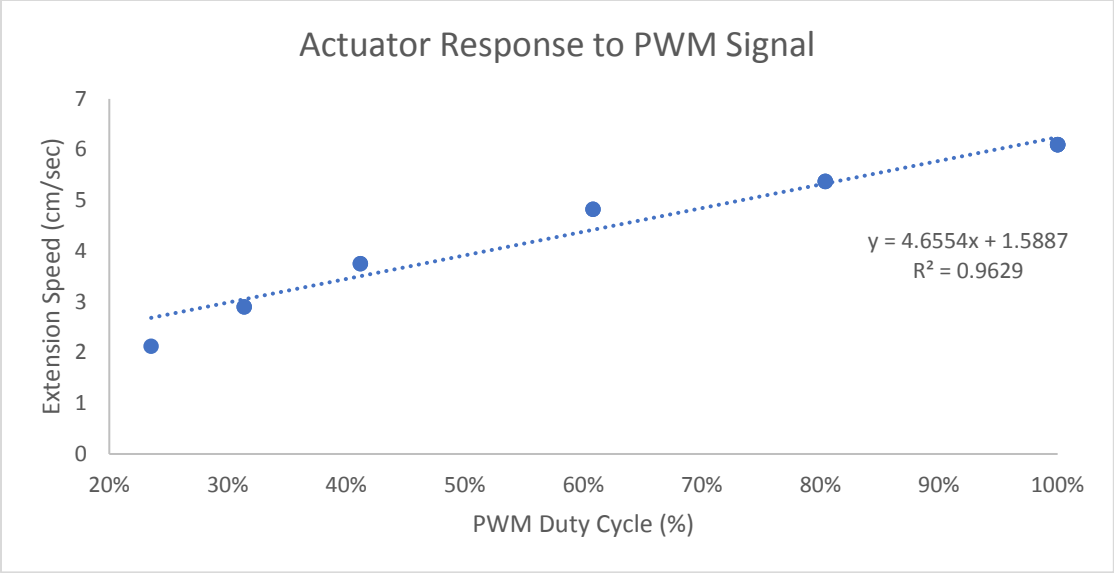

Figure 13: Linear actuator PWM signal response.

As would be expected with a DC motored system, the linear actuator exhibited a linear response to the modulated signal. It was observed that the PWM duty cycle had to be maintained above 22% to extend/retract the actuator, the actuator was unresponsive to a lower duty cycle. At 100% duty cycle, the linear actuator achieved a maximum calculated actuation speed of 6.1cm/sec. Any deviation in the linearity of the calculated extension speed with respect to the PWM duty cycle can be attributed to measurement error, as the measurements were performed manually.

After the linear actuator was installed in the bicycle, a static test was performed to evaluate the actuator's capability turning the handlebar. It was suspected that the maximum actuating speed of the linear actuator would be a little slower when coupled with the resistance of the handlebar. The test was performed by starting the actuator in its fully retracted condition. It was then extended for a given time period (at full speed). The actuated length was then measured, similarly to how it was measured in the table test. The actuation speed stabilized at 5.8

cm/sec. Figure 14 displays the measured response speed with the actuator installed on the bicycle.

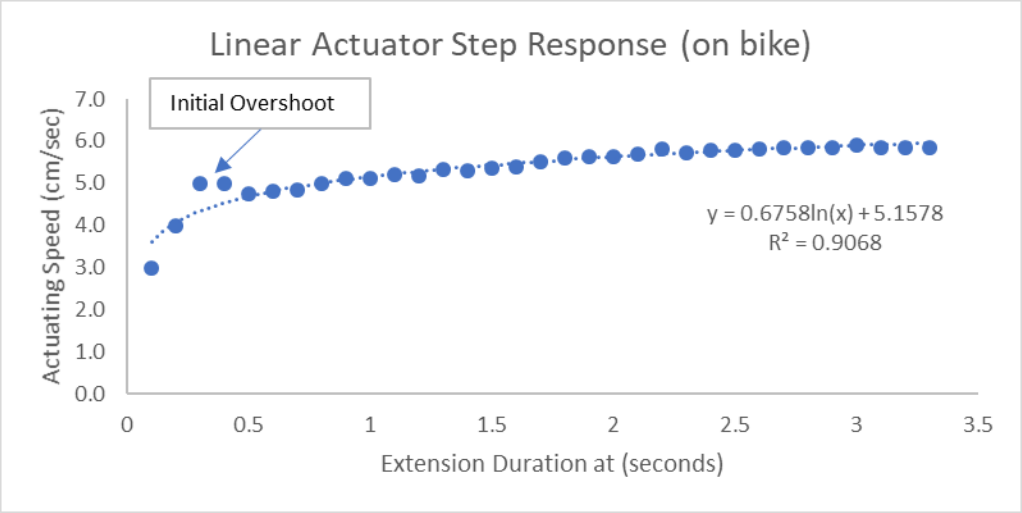

Figure 14: Step response – linear actuator extension speed (installed on bicycle).

It is expected that the actuation speed is slightly lower on the bicycle because of the load of the handlebar (the mass of the handlebar and tire contact force with the ground) that must be overcome to extend and retract the actuator. When the test was performed on the bicycle, a higher number of measurement points were taken to gain more certainty in the speed capability of the linear actuator. There is a small overshoot observed in the initial 0.5 second data range. It is possible that the overshoot is due to measurement error, rather than it being a characteristic of the motor or control system response. Whether the overshoot is due to measurement error or not, the system response appears acceptable for stability control of the bicycle.

The linear actuator is not perfectly in plane rotation of the handlebar, however; using the law of cosines, the measurement extension length was converted to a steering angle rate measurement over time. Figure 15 shows the steering rate over the measured time. The actuating system can rotate the handlebar at a maximum speed of 30-35 deg/sec. This steering rate was deemed acceptable for the purpose of proving the system's ability to balance the bicycle.

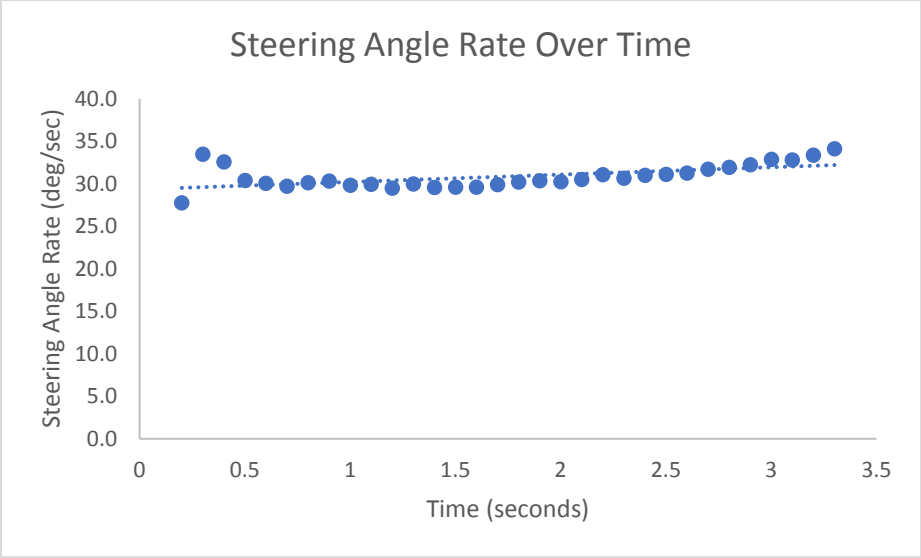

Figure 15: Step response - steering angle rate.

Figure 16 shows how the steering rate variation compares with angle of the front fork. The figure shows that the steering rate is consistent across steering angles +40 deg (clockwise) and -40 deg (counterclockwise), a span large enough to be capable of balancing the bicycle.

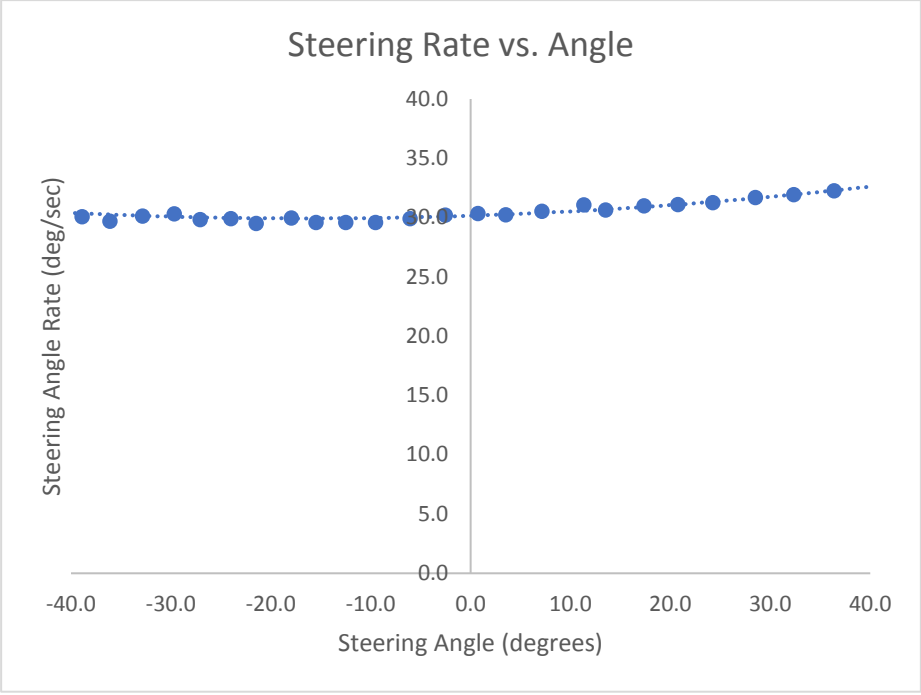

Figure 16: Steer angle rate compared to steering angle.

Chapter 3 Bicycle Stability Control

3.1 Controls and Sensing

The microcontroller used in building the prototype for the thesis was an Arduino Uno. The Arduino platform was selected as the microcontroller was already available (no cost), and because of familiarity with the software IDE. With the controller software being open source, a vast library of materials and support are also available online. Although there are many microcontrollers and software controller platforms that are more capable, the familiarity with the IDE, and the well documented support materials online made it easier to get the project started, spend less time debugging for code errors, and to focus on the testing and data analysis.

The Arduino Uno is an entry-level microcontroller integrated PCB board. The Uno uses an Atmel ATmega328P microcontroller with 32 kilobytes of memory, 2 KB of SRAM, and 1 KB of EEPROM. On the board are 14 digital I/O pins (6 of which can output a PWM signal), 6 analog input pins, as well as power connectors (5-Volts) and a USB connector port. The Arduino Uno board can interface with other microelectronics using I2C, SPI, UART, and USB communication protocols [48], [49].

To effectively actuate the handlebar and use steering control to balance the bicycle, it is necessary to know the roll angle and the roll rate of the bicycle. To do so, an InvenSense MPU 6050 inertial measurement unit (IMU) was mounted to the bicycle. The IMU has an integrated 3-axis gyroscope, and 3-axis accelerometer, and an integrated digital motion processor (DMP). The purpose of the DMP is to take the raw output data from the gyroscope and to perform motion measurement calculations. The IMU has programmable settings for angle rate and acceleration

ranges. The MPU 6050 uses I2C communication protocol to communicate with other devices. Detailed specifications of the MPU 6050 are available online [50].

To take the raw outputs from the gyroscope and accelerometer and perform and calculating/tracking the motion of the accelerometer requires a lot of processing power. There are also important factors, such as temperature, that impact the sensor's readings. Therefore, the DMP was utilized to track the motion of the accelerometer and it was reported out to Arduino Uno using a FIFO buffer. The DMP reports out the acceleration and roll rates using a quaternion parameterization. An open-source library for I2C devices, which is recommended on the Arduino content contributor forum, included an extensive library of functions for utilizing the DMP on the MPU 6050 [51], [52], [53]. The library included functions such as initializing the DMP, checking for error states, querying the FIFO registry, setting the programmable options for the MPU 6050, sensor calibration, and converting the quaternion angles into Euler angles.

Utilizing the DMP and recommended source code and libraries for gathering data from the MPU 6050 saved time, computational resources on the Arduino Uno, and allowed for confidence that measurement output values from the accelerometer and gyroscope were being appropriately reported to adjust for temperature offsets, signal noise, drift, etc. [50]. Table 6 lists the specifications of the InvenSense MPU.

Table 6: InvenSense MPU specifications.

Gyroscope Programmable Ranges	±250, ±500, ±1000, ±2000°/sec
Gyroscope Data Output Rate	4 to 8,000 Hz
Gyroscope Data Tolerance	± 3% error
Accelerometer Programmable Ranges	±2g, ±4g, ±8g, ±16g
Accelerometer Data Output Rate	4 to 1,000 Hz
Accelerometer Data Tolerance	± 3% error
Register Communication Protocol	I2C

Using the reported roll angle from the MPU 6050, the Arduino continually adjusts the PWM inputs that actuated the bicycle handlebar. In this way, the IMU completes the closed-loop control system, whereby the Arduino microcontroller can control the balance of the bicycle via steering inputs. A high-level description of the processes performed by the microcontroller are shown below. The amount of time for the microcontroller to complete one control loop was recorded to be 14 to 16 milliseconds (15ms average).

Figure 17 shows the electrical systems diagram and what signals are used to communicate and power each portion of the electrical system.

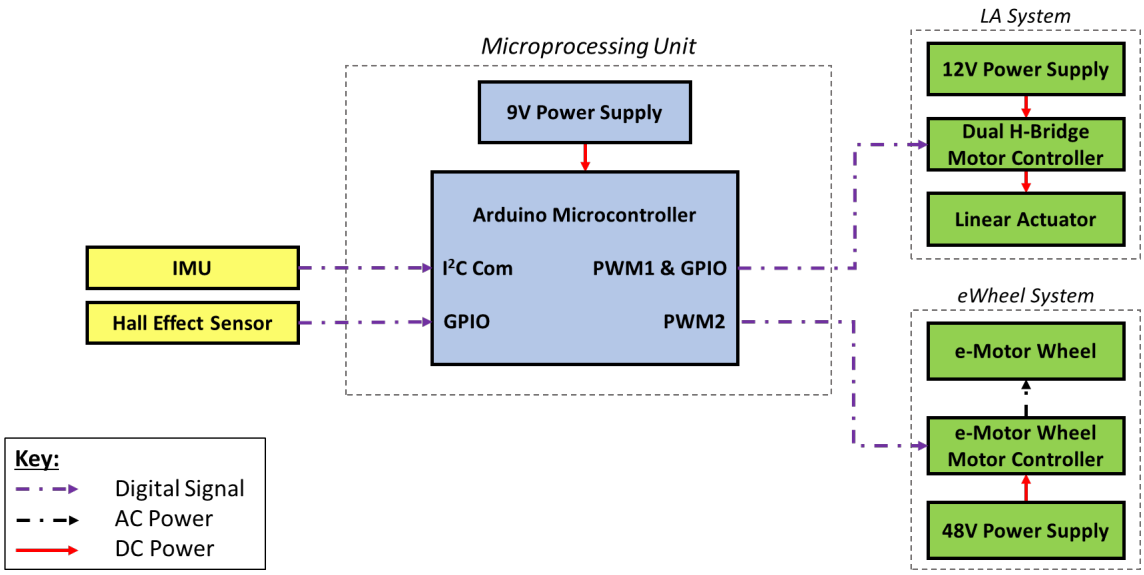

Figure 17: Embedded controller system diagram.

Figure 18 is a control flow diagram that shows the primary operations of the Arduino microcontroller to maintain stability of the bicycle. The Arduino microcontroller operating system partitions operations into an initialization period, and a continuous loop. During the initialization phase, the Arduino begins communication with the accelerometer and initiates power to the motor. The continuous loop portion of the code is focused on measuring the roll

angle of the bicycle and appropriately changing the actuation speed of the linear actuator to maintain the stability of the bicycle.

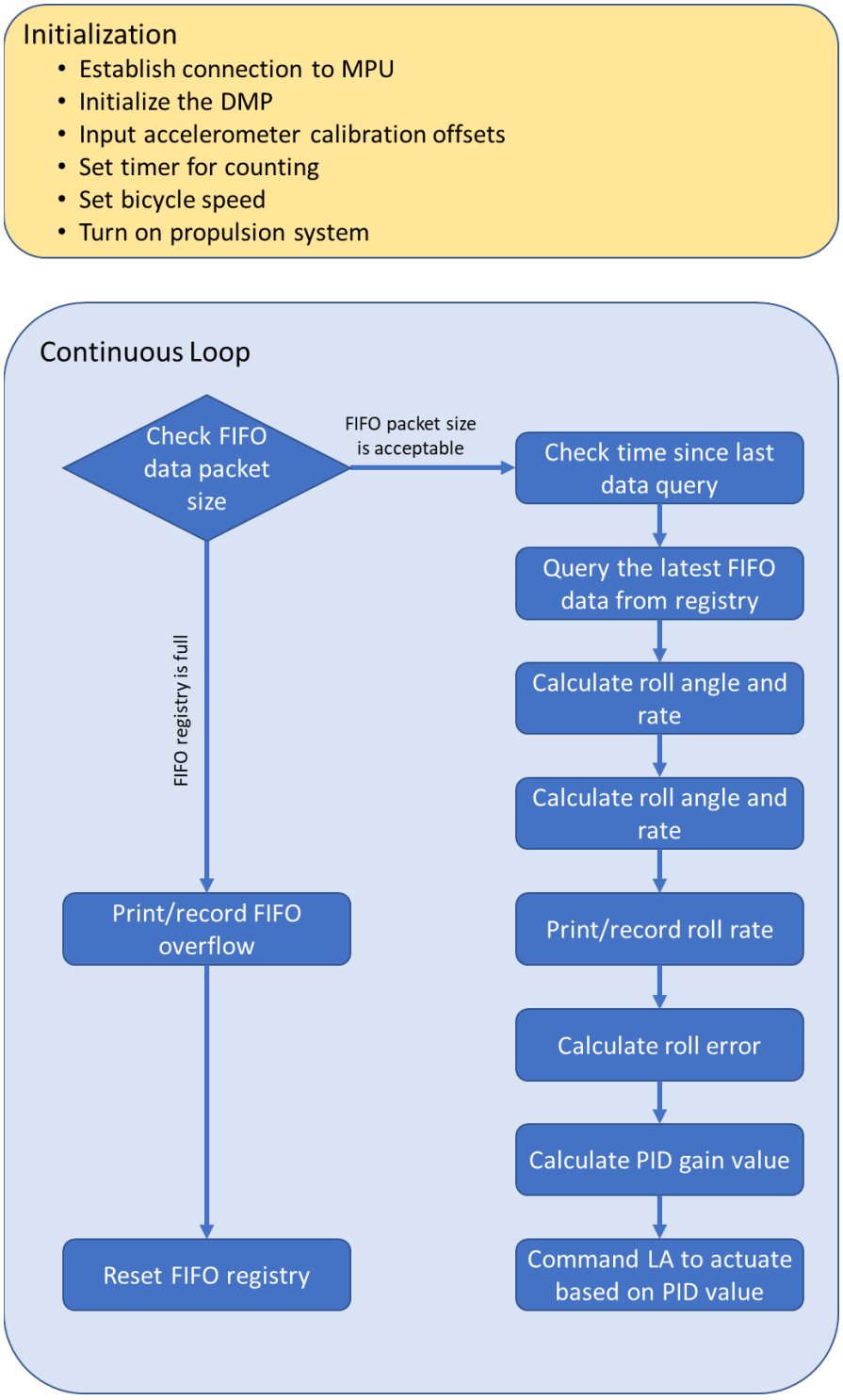

Figure 18: Microcontroller control flow diagram.

3.2 Bicycle Dynamics

As referenced in Chapter 2, there is some prior work on the stability control of bicycles and other single-track vehicles. A simple linearized model of the bicycle was first presented by Carvalho over a century ago [54]. Over the years, the topic bicycle dynamics has been explored in many dissertations and papers. Many models and experiments have been developed to understand the design and dynamics factors that impact bicycle stability and maneuverability. As the modern bicycle has evolved, so has the modeling of the system. The earliest models of the bicycle were linearized second-order models. Throughout the 1900's, modeling evolved in complexity to include both linear and non-linear fourth-order models. Modern computational techniques have also been implemented to bicycle dynamics. Bicycle dynamics models have been implemented both in numerical simulation software (such as MATLAB and Maple) [22] as well as object oriented and multi-body dynamics software (such as ADAMs) [55]. All modern models agree that the bicycle is, at a minimum, a non-linear, nominally unstable, 2^{nd}-order dynamics system.

Figure 19 is a diagram of the geometry of a bicycle. The wheelbase and head angle of the bicycle have a large impact on the system dynamics.

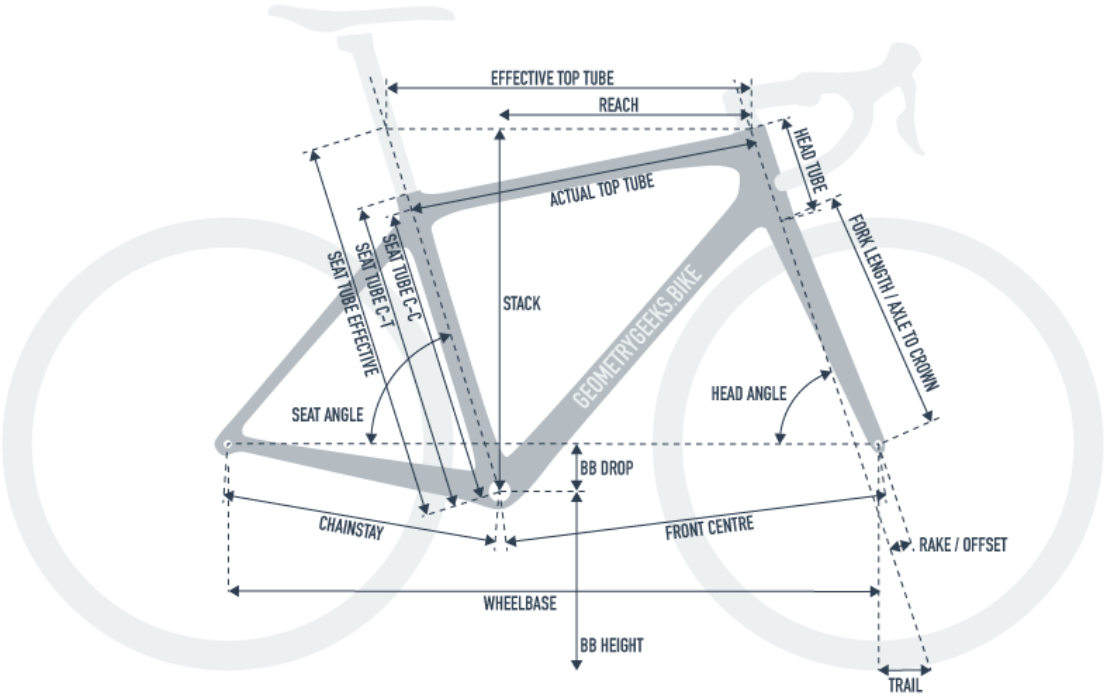

Figure 19: Bicycle geometry [56].

The second-order models make simplifying assumptions such as no tire slip (pneumatic rubber tire affects are ignored), the steering fork being perpendicular to the ground (a 90-degree head angle with no offset, or trail), constant velocity, a fixed center-of-gravity (relative to the bicycle frame), and a rigid rider and frame [12]. Small angle assumptions are also used to linearize the model. The second-order linearized models describe the bicycle system very closely to that of an inverted pendulum (two poles mirrored across the imaginary axis from one another). The input variable is usually modeled as a steering angle input and creates a direct relationship between the steering angle and the roll angle of the bicycle [12].

More complicated models include geometric additions (such as the angling of the front fork), the gyroscopic effects of the wheels, and substitutions for a steering torque input (rather than steering angle) [12]. Models for human riders (or leaning mass systems) also include a leaning mass. Fourth-order models include the leaning mass as a dynamic input, and non-linear models also include the effects of pneumatic tire slip and tire friction [12].

Åström provides a detailed inspection of various aspects that impact bicycle dynamics. Citing the work of other authors, Åström notes that the effect of the gyroscopic affects from the tires are relatively small compared to the centripetal forces on a bicycle during stability and turn maneuvers. He also notes that the front fork design on a bicycle has a large impact on the dynamics of the system. For example, a bicycle with a greater head angle, has greater self-stabilizing characteristics [12, see pages 19&20].

In this body of work, the stability of the bicycle is controlled solely by the actuation of the front steering fork. Also, the trail on the prototype bicycle frame is quite steep (greater than 75 degrees). Therefore, a simple second-order model was chosen to describe the dynamics of the system, and for stability analysis.

While more detailed models exist, the Karnopp bicycle model was chosen as the baseline dynamics model for this work [57]. The primary advantage of the Karnopp model over other models that were considered is that it takes the complexity of a fully described bicycle system (bicycle fork stability features, non-linear tire modeling, gyroscopic effects of spinning wheels, etc.) and yet simplifies it to a 2nd-order linear model. As the intent of this research is to test the control model experimentally, it was necessary to choose a bicycle model that could accurately be monitored by a micro-controller, without becoming too resource intensive (due to a high number of sensors, or algorithmic complexity).

For the interested reader, a summary of the Karnopp dynamics model is described below.

3.2.1 Model Derivation

The dynamics of the bicycle can be understood by considering the traditional "bicycle model" used for automotive applications. Figure 20 shows the ground plane geometry for a two-wheeled system vehicle in a turn.

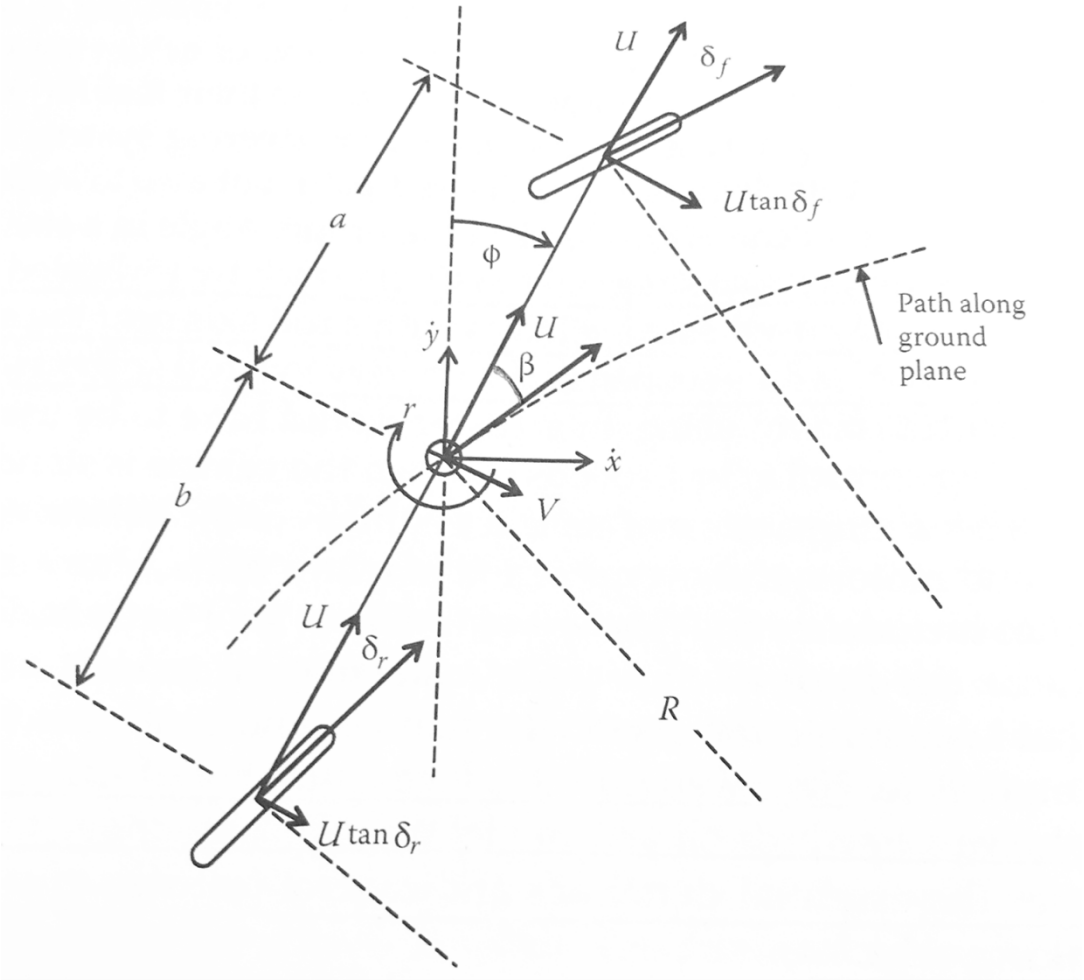

Figure 20: Ground plane geometry for a two-wheeled vehicle in a turn [57].

By assuming the bicycle traveling at a velocity (U), that the steer angle is small relative to the bicycle's turn radius (R), and that there is no tire slip, it is possible to create a kinematic relationship that directly relates the handlebar steering angle to the path of travel of the vehicle. Using a small angle assumption for the steering angle (δ_f) and (δ_r), the turn radius of the bicycle is as follows:

$$R = \frac{a+b}{\delta_f} \tag{1}$$

Those familiar with using the bicycle model for automobile applications may find it strange to see the tire slip angles ignored, as they have a large impact on the stability of an

automobile. However, as Karnopp and Åström explain [57],[12], under small and even moderate lateral acceleration, the dynamics of a tilting bicycle are heavily governed by the tilt angle of the vehicle and are less affected by tire slip angles (so long as the bicycle wheels do not skid). Acting under the same assumptions, the yaw rate of the bicycle (r) is directly proportionate to its forward velocity and the turn radius which it is following. The lateral velocity (V) of the bicycle (at center of mass projected onto the ground plane) is described below:

$$r \cong \frac{U}{R} = \frac{U\delta_f}{a+b} \tag{2}$$

$$V \cong U(b\delta_f)/(a+b) \tag{3}$$

Figure 21 shows center of mass of the bicycle leaning in during a steady-state turn. The center of mass of the bicycle is suspended at a height (h) above the ground and at a roll angle (θ) relative to the tilt axis of the bicycle in the ground plane. The velocity components from the ground plane (U and V) and the angular rotations (r and θ) are displayed at the CG.

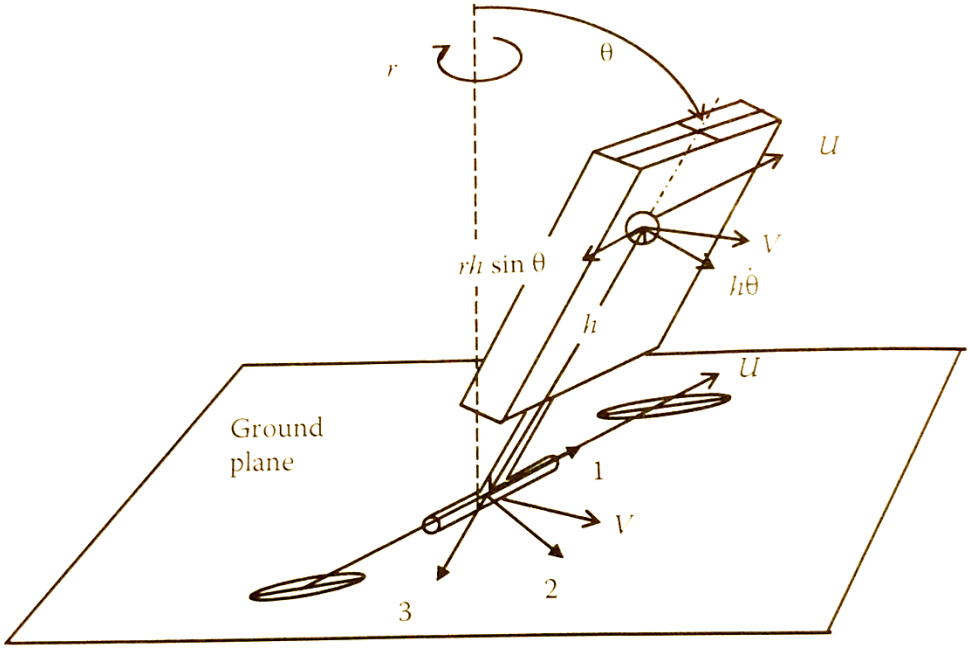

Figure 21: The vehicle body tilting around the ground plane axis [57].

A detailed derivation of the equation of motion of the bicycle via the LaGrange Method can be referenced in [57]. Suffice it to say that the total sum of the kinetic energy of the bicycle is made up of velocity components described at the center of gravity, as well as the angular velocities of the bicycle about the 1-2-3 axes. The resulting equation of motion is described below:

$$(I_1 + mh^2)\ddot{\theta} + (I_3 - I_2 - mh^2)r^2 cos\theta sin\theta - mghsin\theta = -mhcos\theta(\dot{V} + rU) \quad (4)$$

By using the small angle assumption, the linearized version of the equation of motion is as follows:

$$(I_1 + mh^2)\ddot{\theta} - mgh\theta = -mh(\dot{V} + rU) \quad (5)$$

Noting that the right-hand side of the equation is governed by inputs from the ground plane, a bicycle can be considered as an inverted pendulum mounted on a laterally moving platform [20]. By relating the lateral acceleration and yaw of the bicycle back to the steering angle, it is shown that the stability of the bicycle can be completely controlled by the steering angle and forward velocity of the bicycle:

$$(I_1 + mh^2)\ddot{\theta} - mgh\theta = -\frac{mh}{a+b}(b\dot{\delta}_f U + \delta_f U^2) \quad (6)$$

Where:

I_1 – Inertia of the bicycle about the roll-axis of the bicycle
m – Mass of the entire autonomous bicycle system
h – Height of the center of gravity (above the ground plane)
$\ddot{\theta}$ – Second derivative of the roll angle
a – Distance from the front axle to the center of gravity (ground plane projection)
b – Distance from the rear axle to the center of gravity (ground plane projection)
δ_f – Front fork steering angle
U – Forward velocity of the bicycle

Note in the stability equation that if the bicycle is stationary (the speed is zero), the actuation of the has no bearing of the roll angle on the bicycle. Also, the higher the speed bicycle, the more sensitive the roll rate is to the steering angle and steering rate.

3.3 Stability Analysis

To simplify the stability analysis, Karnopp defined the following constants:

$$\tau_1^2 = \frac{(I_1 + mh^2)}{mgh}, \tau_2 = \frac{b}{U}, \tau_3 = \frac{a}{U}, K = \frac{U^2}{g(a+b)} \tag{7}$$

The transfer function is therefore shown as follows:

$$(\tau_1^2)\ddot{\theta} - \theta = -K(\tau_2\dot{\delta_f} + \delta_f) \tag{8}$$

Thus, the characteristic equation for the system is of the form $\tau_1^2 s^2 - 1 = 0$. With two poles in on the real axis mirrored at $\pm \frac{1}{\tau_1}$ (the same as an inverted pendulum). The stability of the system can be improved and by increasing the height of the center of gravity, the mass, or increasing the rotational inertia of the bicycle. This model of the bicycle explains how the steering angle and roll angle of the bicycle are interrelated and explains the steady-state motion of a bicycle. The transfer function produced, uses the steering angle as the input lean of the system.

It is important to note that in this simple model of the bicycle, the bicycle is not capable of self-stabilizing. The primary reason for this is that the front fork dynamics have been ignored. As noted in [12], when a bicycle is in a lean, the tire-road contact forces exhibit a torque on the bicycle front fork. On a bicycle with a positive trail, the contact forces exhibit a counter-torque on the front fork. As the speed of the bicycle increases, so do the tire-road contact forces. Since the steering angle and roll angle of the bicycle are interrelated, the counter-torque on the front fork results in negative feedback to the roll angle of the bicycle, provided that the torque is large

enough to change the steering angle. Therefore, all bicycles with a positive trail can be self-stabilizing under certain speed and leaning conditions. This explains why a skilled bicycle rider can ride the bicycle hands free.

Is it appropriate to use a bicycle model that ignores the self-stabilizing impact of the front fork geometry?

First, the control method proposed utilizes a linear actuator which is mechanical joined to the front steering fork. The linear actuator has minimal back-drivability; therefore, any counter torque effects from the front fork tire contact will be mitigated because they will not be large enough to change the steering angle of the front fork. While the counter-torque forces on the handlebar will still impact the path of the vehicle (they still impact the effective steering angle of the bicycle), it will not improve the bicycle lean stability. Second, the bicycle used for experimental validation of the control method is a "neutral" bicycle, meaning the trail on the front fork is minimal. Finally, the front fork geometry improves (not decreases) the stability of the system, it is a conservative assumption to ignore its impact on the steady-state stability of the system.

3.3.1 Stability Modeling

The prototype bicycle system was measured and weighed. Table 7 shows the input parameters which were used to model the system in MATLAB.

Table 7: MATLAB input parameters for stability analysis.

Parameter	**Value (units)**
Inertia (about the ground plane roll axis)	13.48 kgm^2
Mass	34.27 kg
CG Height	0.575 m
A (front axle to CG)	0.721 m
B (rear axle to CG)	0.351 m
U (forward velocity)	*See figures below*

Using the MATLAB Control Toolbox, the bicycle transfer function was modeled by defining Karnopp's coefficients (see equation 7 and 8).

The pole plot for the bicycle is shown below:

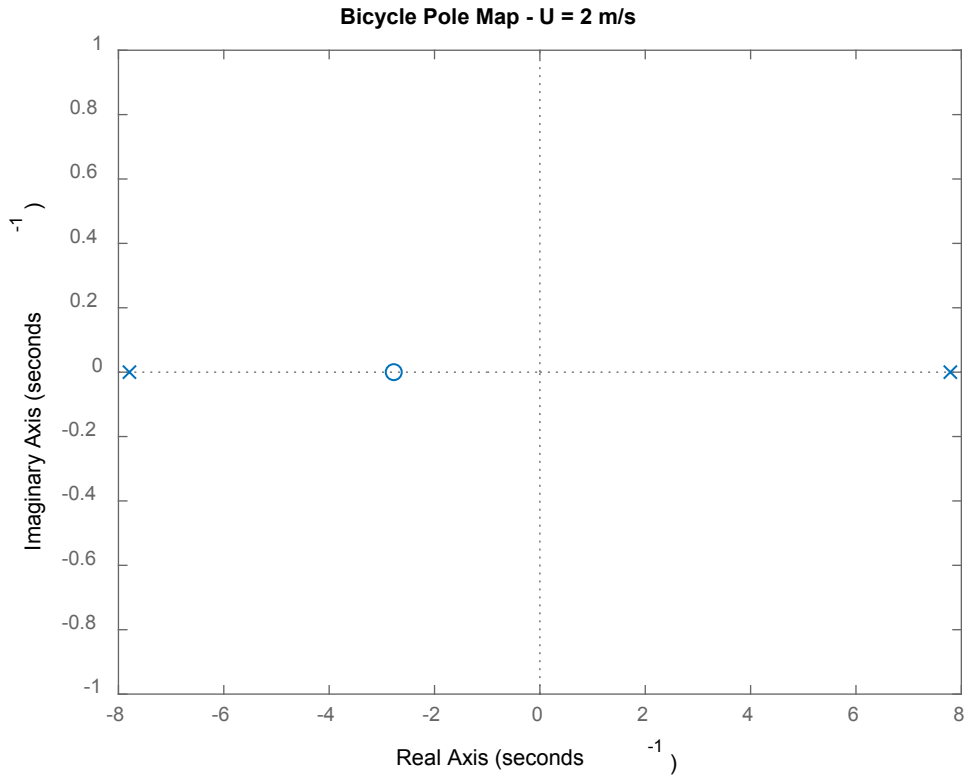

Figure 22: Pole plot for a bicycle plant (uncontrolled).

As explained previously, the system is unstable with a pole in the right half plane.

3.4 Controller Design

As outlined in 3.3, the geometry and inertial properties of the bicycle impact the system stability. However, the only controllable variables are:

- Actuation of the front fork (steering angle and steering rate), and
- Forward velocity of the bicycle

Chapter 2 notes various works on single-track vehicle stability control through front fork actuation. Schwab gives a good summary of control methods that have been successfully implemented [7, see Section 3]. These include optimal (LQR) controllers, PID and LPV controllers, fuzzy logic controllers, neural network controllers, inverse and forward dynamics controllers, and intermittent controllers. Some have been validated experimentally though many (especially the more computationally complex control methods) have only been simulated. Even in the application of the fuzzy logic and neural network controllers, the controllers are ultimately PID controllers. The fuzzy logic and neural networks are used to intelligently select the PID gains applied to the control signal [18].

Many of the controllers and experiments reported use full-state feedback models, requiring state observers and state estimation models. Also, many of them use a fourth-order dynamics model of the bicycle. From an applications perspective, higher fidelity (more complex) dynamics and controller models are computationally more expensive and require more sensors for state estimation. Since the objective of this research is to develop a simple and computationally efficient controller, a simple PD controller was chosen to control the actuation of linear actuator.

It will be shown that through gain scheduling, the controller can successfully balance the bicycle at different speeds. As noted in Chapter 2.3.2, the speed of the bicycle is maintained an open-loop controller (the Arduino sends a command signal for wheel speed, but does not actively

measure it); therefore, the only states necessary to measure are the roll angle, and the roll rate of the autonomous bicycle system.

3.4.1 Proportional Control

First, it must be understood that proportional feedback alone (based on the roll angle of the bicycle) is not sufficient for stabilizing the system. A proportional gain controller was added to the MATLAB bicycle dynamics model. The pole plot in Figure 23 shows the poles of the system with various proportional gain constants. As shown in Figure 23, the poles remain in the right half plane, and increasing the gain only serves to push the pole towards positive infinity.

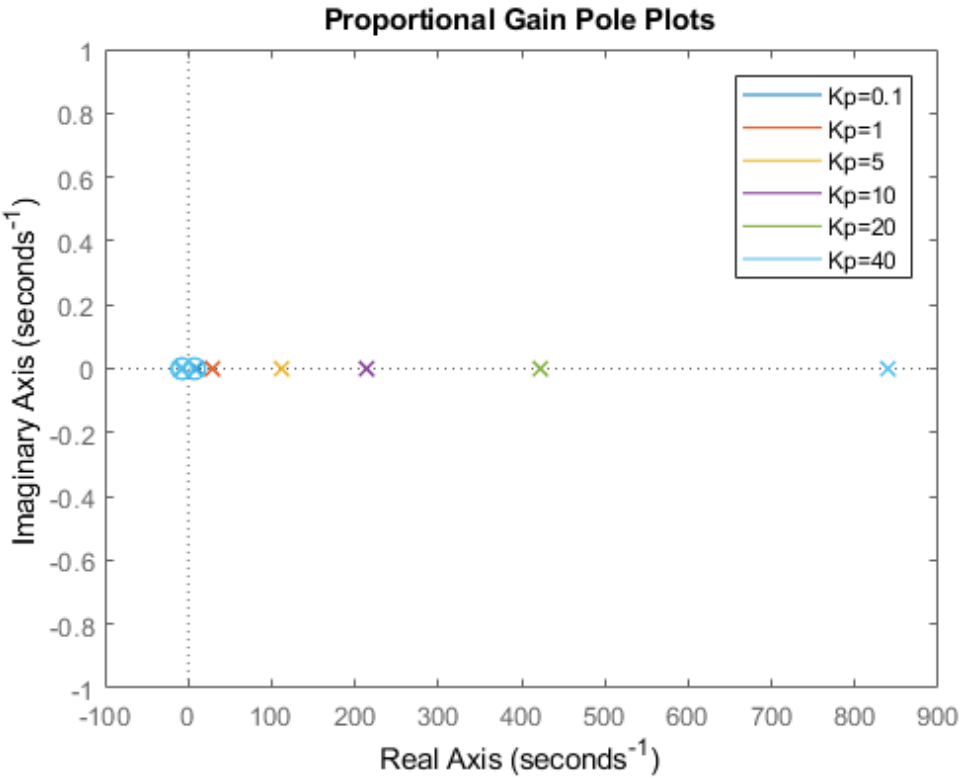

Figure 23: Proportional gain plot.

This was observed experimentally as well. With a low gain value, the steering angle did not increase quickly enough to counter the fall of the bicycle. With a larger gain, the linear actuator would counter-steer quick enough to counter the roll angle, however, even with the

bicycle returning to the upright position, the steering angle would continue to increase. The resulting action is that the bicycle would then roll in the opposite direction and, with the steer angle now moving the wheels in the wrong direction, the bicycle would overshoot the upright position and then quickly fall to the ground.

3.4.2 PD Control

By understanding from a physics perspective what happens when a controller is based on roll angle alone (proportional feedback), it becomes clear that the controller must be able react to changes to the roll rate of the bicycle as well. Adding derivative feedback to the controller allows the controller to assess changes in the roll angle rate. This improves the controller in 2 ways: 1) When the roll rate is in the same direction as the roll angle (when the bicycle is falling), the gain is increased. 2) When the roll rate is in the opposite direction of the roll angle (when the bicycle is returning to the neutral position), the gain is decreased.

As can be seen in Figure 24, adding derivative control not only moves the poles of the system to the left half plane, but it also provides a tunable parameter that controls the damping ratio of the system.

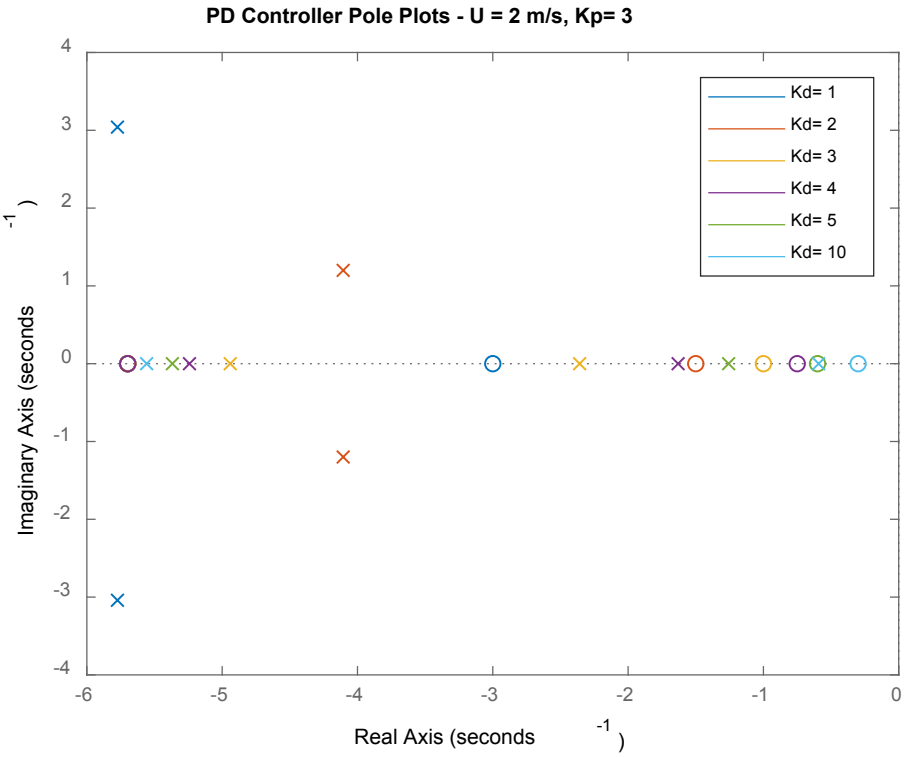

Figure 24: Pole plot of a bicycle with a PD controller.

Derivative feedback allows the controller to adjust the command signal to the linear actuator as soon as the bicycle begins to return to the upright direction. Figure 24 shows that the PD controller can stabilize the bicycle system, but the system response is highly dependent on the proportional and derivative constants that are shown. The proportional and derivative gain constants can be tuned to give the system any desired system response. From an applications perspective, a faster system response usually comes at the cost of more expensive hardware and computational power.

Since the system's roll and steering angles impact the tracking of the vehicle, it is desirable to increase the response time as much as possible. If damping ratio and settling time is too long, the oscillations of the steering angle would cause the vehicle to deviate from its desired path. For the bicycle speed and proportional gain selected, a Kd value between 2 and 3 would yield a critically damped or slightly underdamped system response.

There are three parameters to consider when setting the proper gain constants for the PD controller: vehicle speed, the proportional gain constant, and the derivative gain constant. The impact of each will be shown below.

3.4.2.1 Impact of the Proportional Gain Constant

Changing the proportional gain changes the system response for a given set of derivative gain values, and at a given vehicle speed. As an example, in Figure 25 the bicycle system response is shown with the velocity and derivative gain held at a constant value. The proportional gain value is varied from a value of 1 to 8.

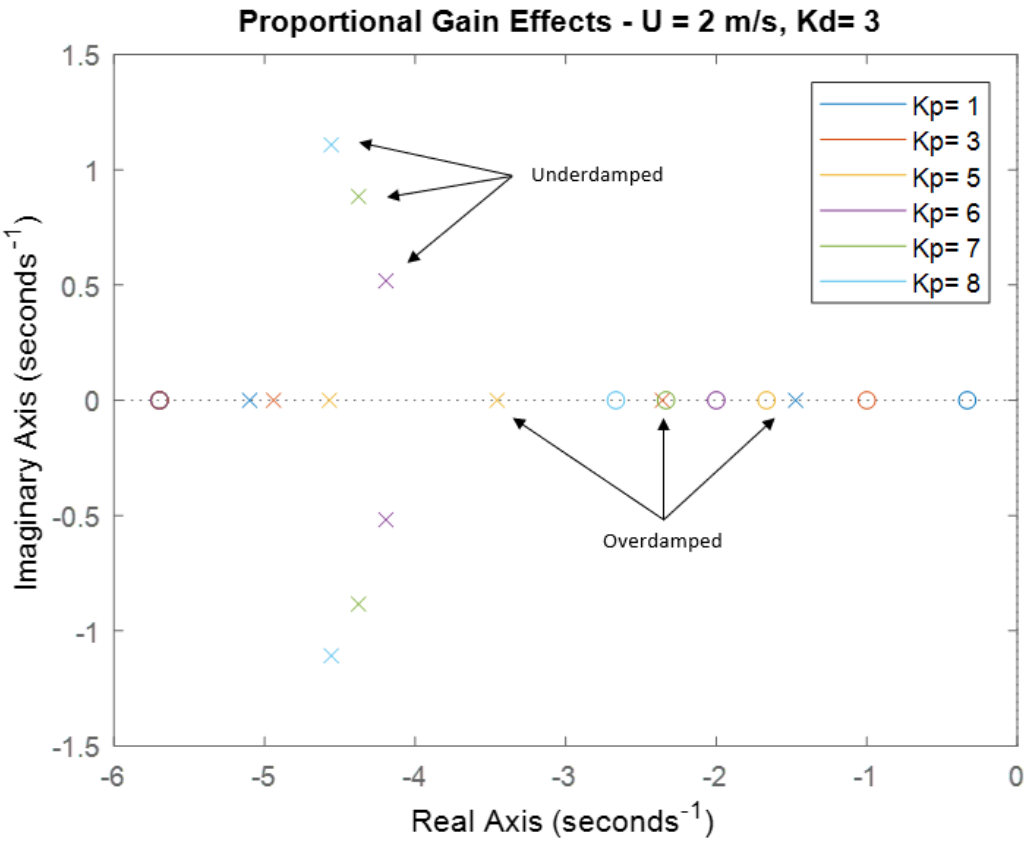

Figure 25: The impact of the proportional gain constant on system response.

With the smaller proportional gain constants, the system is overdamped. However, if the proportional gain constant is increased to a value greater than 5, the system becomes underdamped.

3.4.2.2 Impact of the Derivative Gain Constant

Changing the derivative gain also the system response. Figure 26 shows how the system response changes with the derivative gain constant. The velocity of the bicycle and proportional gain constant are held constant. The proportional gain value is varied from a value of 1 to 10.

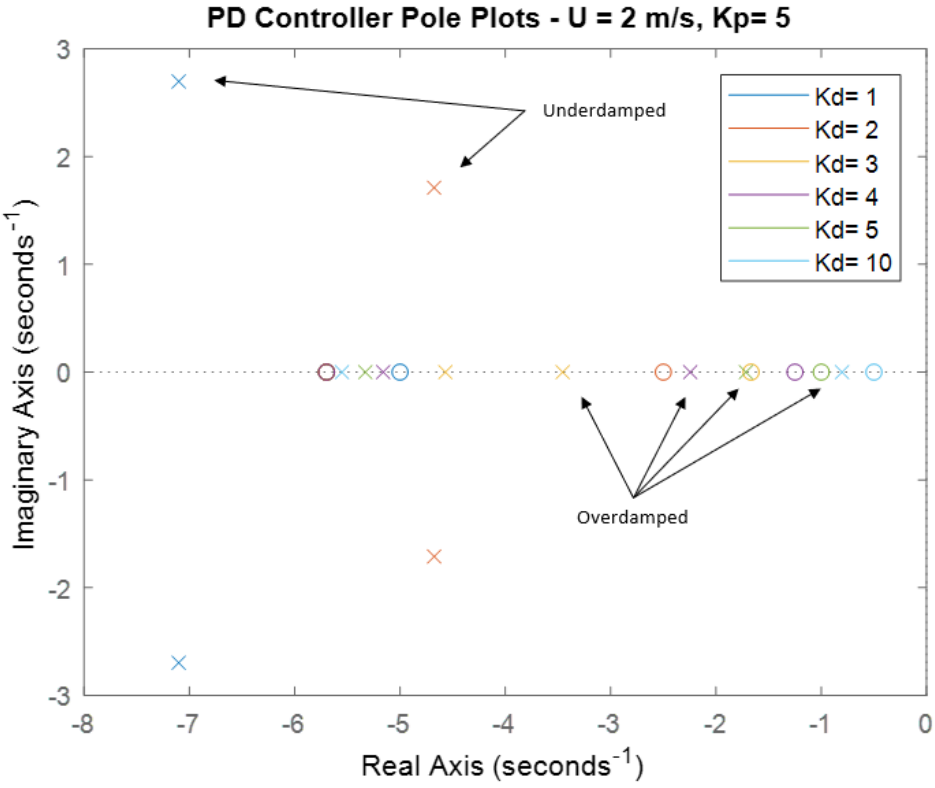

Figure 26: The impact of the derivative gain constant on system response.

With a derivative gain constant of 1 or 2, the system is underdamped. However, the system is overdamped when $K_d \geq 3$. Therefore, the ideal value for the derivative gain constant lies somewhere between 2 and 3 for this particular speed and proportional gain constant.

3.4.2.3 Impact of Vehicle Speed

The bicycle speed also changes the system response. Figure 27 shows the system response across a speed range of 0.5 to 3 meters per second. The proportional and derivative gain constants are fixed at a value of 3.

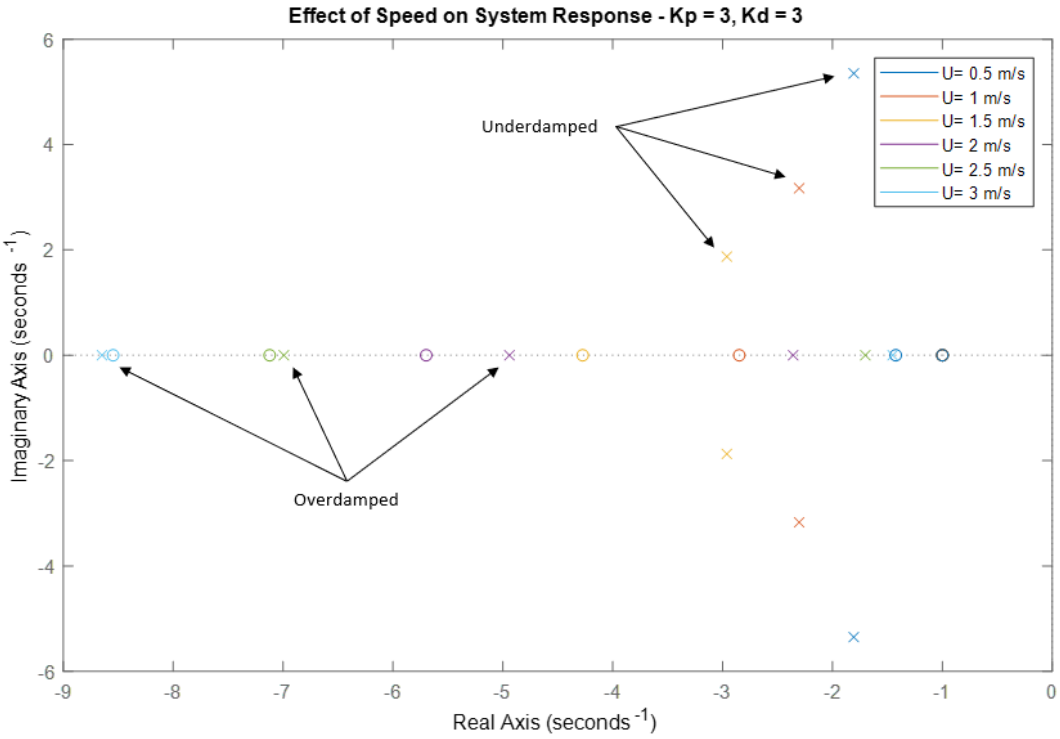

Figure 27: Effect of vehicle speed on system response.

With the fixed gain values selected, the bicycle must maintain a speed between 1.5 and 2 meters per second to remain a critically or slightly underdamped system and maintain its fastest response time. Figure 27 helps demonstrate that gain scheduling is necessary to tune the PD controller for different speed regimes.

3.4.2.4 Need for Gain Scheduling

The proportional gain, derivative gain, and vehicle speed all have a substantial impact on the behavior of the system. It is necessary to tune and modify all of three of these parameters in concert with one another. A gain scheduling approach appears to be appropriate since an

autonomous bicycle system will be expected to travel at various speeds. The proportional and derivative gain constants can be tuned to deliver a desired system response for a given speed range. The Bicycle Control Platform would then select the appropriate gains to use based on the speed of travel.

3.4.2.5 The PD/PID Control Law

An important note on the tuning of the PD controller is that this model ignores any dynamics of the linear actuator, as well as any sampling and time delay of the microcontroller. Also, this 2nd Order Model of the bicycle system is very much a simplified model. Therefore, the necessary real-world constants could be different from what is shown in the figures in this section.

After recognizing that steady-state angle errors can persist when the vehicle is close, but not perfectly positioned at the neutral position, integral control was also added also. To help mitigate concerns of integral windup, the following control law was proposed and studied during experimental testing:

$$U_{PID}(t) = K_p e_\theta(t) + K_d \dot{e}_\theta(t) + \int_0^t K_i e_\theta(\tau) d\tau, \quad for\ |\theta(t)| < 2° \tag{9}$$

$$U_{PD}(t) = K_p e_\theta(t) + K_d \dot{e}_\theta(t) \qquad for\ |\theta(t)| \geq 2° \tag{10}$$

With the following parameters:

U_{PID} – The control signal output from the microcontroller at each time step
U_{PD} – The control signal output from the microcontroller at each time step
$\theta(t)$ – Roll angle at each time step (measured)
e_θ –Roll angle error at each time step (desired - measured)
$\dot{e}_\theta$ – Rate of change of roll angle error at each time step (calculated from e_θ)
K_p – Proportional gain constant
K_p – Derivative gain constant
K_i – Integral gain constant

The controller could be considered somewhat of a hybrid controller since it used a different control strategy (PD vs. PID) depending on the state of the system.

Chapter 4 Experimental Results

The experimental objectives were to achieve the following milestones:

1. Demonstrate the use of a linear actuator to control the front handlebar as a viable method for front fork steering actuation.
2. Validate the control laws outlined in the previous section.
3. Show that through gain scheduling, the system could be stabilized at a range of speeds under power.

4.1 Test Setup and PID Tuning Process

The first round of testing was performed without the propulsion system installed. With less mass on the bicycle, it was possible to give the bicycle enough momentum for it to travel a measurable distance with a strong push and give the stabilizing controller time to (re)act. An entire catalog of PID tuning processes is described from both industry and academic sources [58],[59],[60]. Given the high number on non-linearities in a single-track vehicle system, this thesis determined to perform a trial-and-error method that is comparable (in some ways) to the Ziegler-Nichols tuning method. The process is outlined below:

1. Set the derivative and integral constants to zero.
2. Increase the proportional gain constant until the controller can correct an initial fall to one side.
3. Increase the derivative gain constant until the controller sufficiently dampens oscillatory motions and can achieve a steady-state motion.
4. Add the integrator constant to correct for small steady-state errors.
5. Trial higher derivative gain constants to increase response time and minimize overshoot.

The Zeigler-Nichols method and other methods recommended for manual tuning generally begin by isolating a single constant (usually the proportional gain constant) and increasing it until the targeted step response is achieved [60],[61]. In the case of stabilizing a single-track vehicle, it has not been proved possible to stabilize the system with proportional gain alone. Therefore, it was thought that once the proportional controller was able to catch the bicycle system from falling to one side, it would be a sufficient starting point to add in the derivative control. Improvements from steps 2 through 5 were usually assessed by subjectively observing how quickly oscillations were dampened out and measuring the distance the bicycle was able to travel before losing momentum. Figure 28 shows the prototype system being launched for a PID parameter tuning trial.

Figure 28: Initial trial with the bicycle launched with a push.

The process outlined above proved sufficient for stabilizing the unpowered bicycle. The bicycle was found to stabilize itself with $Kp = 10, Kd = 10, Ki = 0.003$. It was also trialed on two separate surfaces (a basketball gym and a sloped parking garage). The plot of the bicycle model in Figure 29 shows that the bicycle system would have its fastest response time (be

critically damped) in the 1m/s to 1.5m/s (2.2-3.4mph) speed range, which is achievable with a strong push.

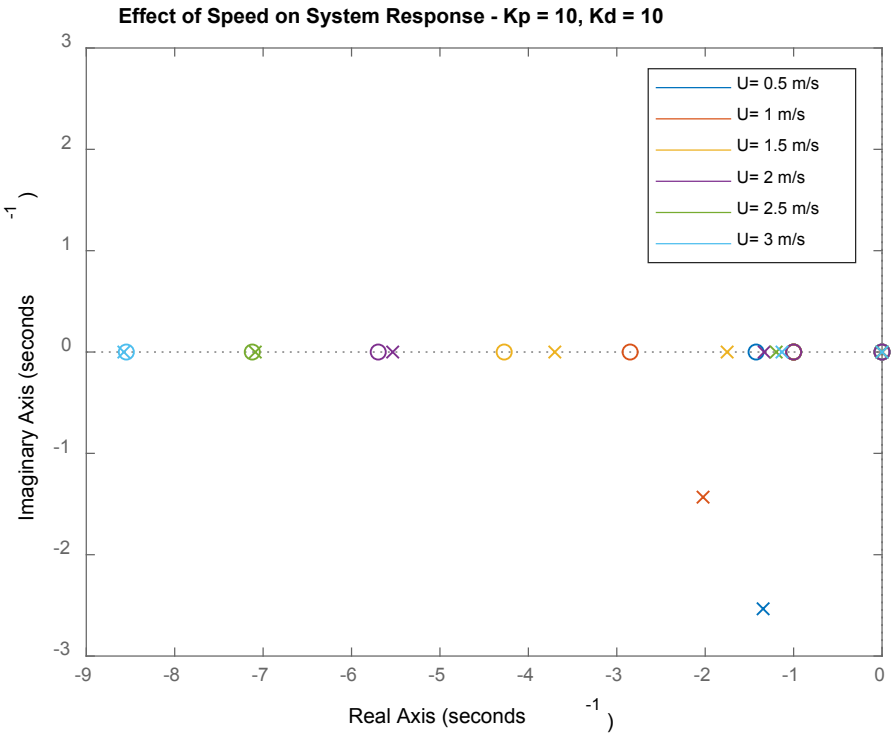

Figure 29: System response with the experimental gain values.

The initial tests and PID tuning process proved that the bicycle could be stabilized by a linear actuator and that the PD/PID hybrid control law proposed functioned as intended.

4.2 Testing Under Self-Propulsion

After the initial control testing was accomplished with the bicycle unpowered, the motorized wheel and propulsion power system were added to the bicycle. The propulsion system added significant weight to the rear of the bicycle which shifted the center of gravity closer to the rear axle and increased the overall inertia of the system.

The bicycle system was now sufficiently heavy that it was difficult for a person to catch the bicycle if it started to fall. Modified training wheels were installed to prevent the bicycle from falling over. When system testing began, the control parameters from the previous testing

were used as a baseline. However, the system dynamics were different enough that the previous PID calibration was not able to maintain the bicycle within the roll angle range allowed by the training wheels. The result was that the bicycle would travel in an s-shaped curve and would oscillate from one side to the other while resting on the training wheels. After making a few modifications to the vehicle speed and derivative constant, it was determined that increasing the proportional gain constant would be necessary as well.

When returning to step #1 of the PID tuning process, the proportional gain was increased until the controller was able to catch the bicycle in a fall. Unfortunately, the bicycle rolled so quickly in the opposite direction that the rear wheel left the ground and the bicycle stood up on the opposite training wheel. The angle momentum carried the bicycle further and it ultimately fell to the ground. Figure 30 shows initial lean of the bicycle (in position 1), and the effect of a sever angle overshoot leading to a fall (in position 2).

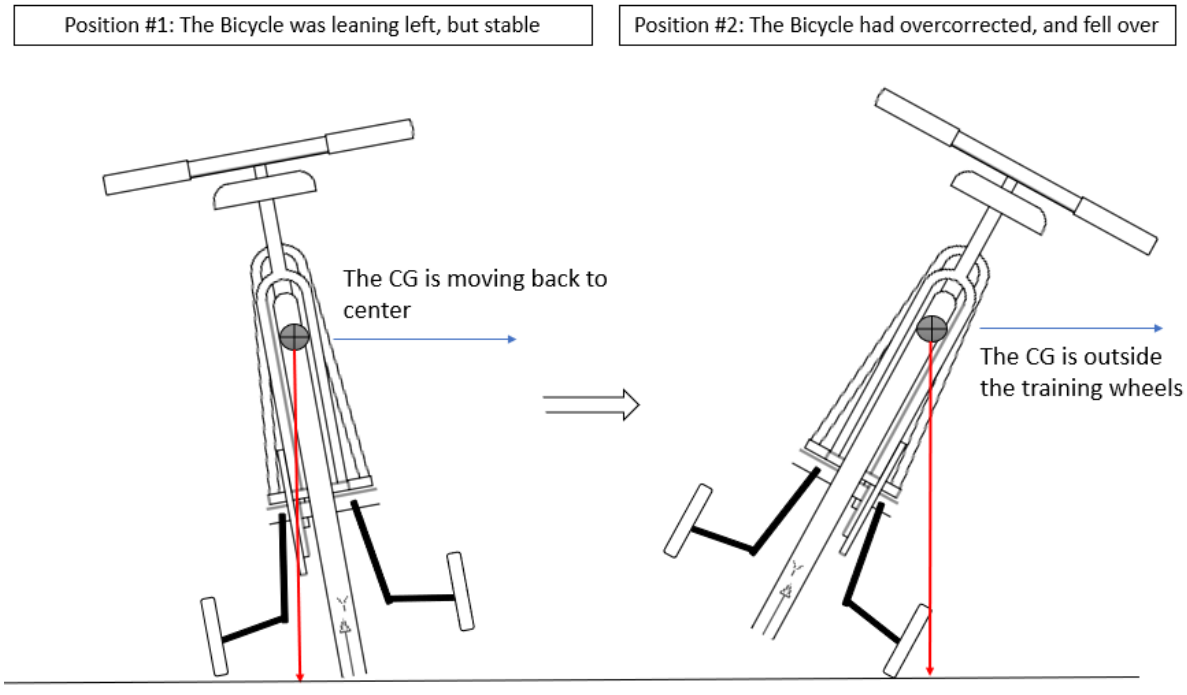

Figure 30: A bicycle in a fall due to a large angle overshoot.

The fall caused significant damage to the support structure on the bicycle, and some of the wires were torn as well. The wooden mounting post securing the linear actuator to the seat post had broken off. Also, the battery rack over the rear axle (which holds the motor control, 12V battery, and 48V battery) broke free as well. The right-side training wheel bracket that the bicycle had rotated on had also bent under the weight of the bicycle. The wooden mounting block and some stripped fasteners were replaced, as well as the wires that had been damaged. The training wheel bracket was also bent back into shape. However, it was noted that even after the repairs, the rear axle battery rack was not as rigid as it had been before the crash.

Due to concerns with the strength of the rear axle battery rack, and that the training wheel bracket may yield again under too much weight, testing with the full propulsion system was postponed (and eventually cancelled due to the COVID pandemic) and plans were made to make a future upgrade to the prototype bicycle structural design.

Testing did continue with the bicycle system (minus the weight of the 48V battery). However, in subsequent tests, the bicycle was launched from a running start at various speeds.

4.3 LPV Control Conceptual Demonstration

A third round of tests were performed with the 48V battery removed (to lessen the weight of the system). The bicycle was launched at two different speeds and the controller response was observed on a sidewalk and closed road in front of the IAVS building on the University of Michigan-Dearborn campus.

The objective was to launch the bicycle at a slow-jog and fast-jog and demonstrate a unique set of PID constants for each speed regime that leads to proper balance control of the bicycle.

The PID tuning process was followed again. A new set of PID constants was generated for the slow and fast jogging speeds. The controller was able to successfully balance the bicycle system. Table 8 shows the unique PID gain parameters used in the two sperate speed regimes. Figure 31 and Figure 32 show the roll angle and measured bicycle speed during the slow and fast speed trials.

Table 8: PID values for two speed regimes.

	Trial #1	Trail #2
Speed regime (m/s)	3.0 – 3.5	3.5 – 4.0
Reference speed	Slow jog	Fast jog
Proportional gain constant (Kp)	18	13
Derivative gain constant (Kd)	28	20
Integral gain constant (Ki)	0.003	0.003

Shown below is the data recorded during both tests:

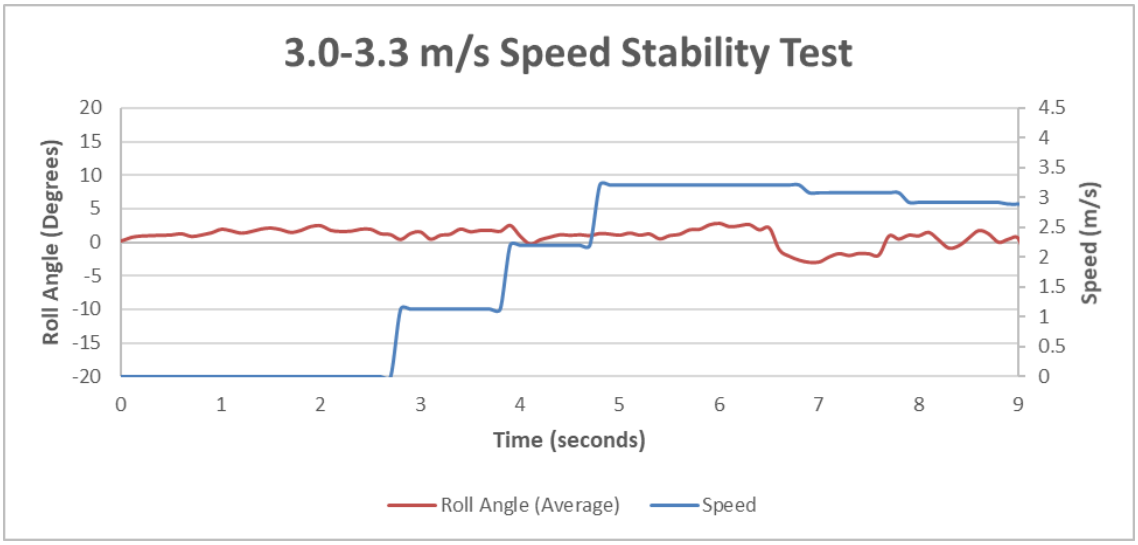

Figure 31: Roll angle measurements from lower speed trial.

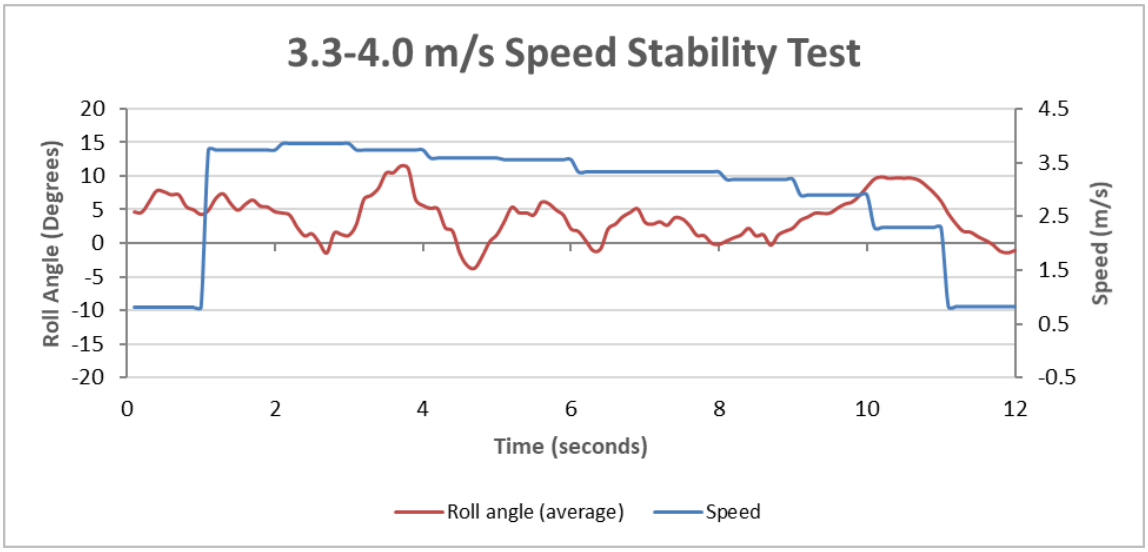

Figure 32: Roll angle measurements from higher speed trial.

Both datasets show that the oscillatory response in the system is not entirely tuned out, although; the lower speed PID constants do appear to be slightly more refined. It is also important to note that the initial conditions affect the quality of the test run.

The initial conditions are as follows:

- The launch/release of the bicycle (how the operator released the bicycle)
- The initial steering angle at the time of the launch (which is never perfectly dead ahead)
- The exact orientation of the accelerometer (if it is positioned a few degrees off its calibrated orientation, measurement error will be present in the data)

Even though the test was performed with a manual launch of the prototype bicycle system. It was still demonstrated how gain scheduling can be used to change the PID values for different speed ranges to keep the controller performing as designed. However, future work should be done to experimentally validate the control laws over a wider speed range.

4.4 Recommendations for Future Prototype Testing

Throughout the testing process several challenges were encountered with setting up each test run, as well as with the electrical and mechanical hardware. The test planning was also

difficult since usually more than one person was required to run the tests. The following opportunities and recommendations are made for future prototypes and testing:

- **A permanent wiring setup**. All wiring to and from the Arduino was an on small 22-gauge wiring connected to and from a breadboard. This worked well for initial signal identification and for subsystem tests. However, once the entire electrical system was connected, the high density of wire connections often led to one or more wires been bumped and losing proper continuity. This resulted in 15-20 minutes of wire inspection before each batch of testing was performed. It is highly recommended that future testing be done with at least having the wires soldered into a joint on a breakout board, so that the connections do not come lose after each day of use.
- **Proper electrical component housing.** The InvenSense IMU unit was installed directly on top of the bicycle seat on top of piece of double-sided tape. Although testing was achievable, great care had to be taken to position the IMU in the exact location both during the IMU calibration and testing process. More than once it was found that the bicycle system would unexpectedly performing as was as it had just a few minutes prior. Many times, it was discovered that the accelerometer was positioned of center, or at an angle. Also, many other control components were also installed on other unique spots on the bicycle. It is recommended that a housing box is made for the components. Especially for the accelerometer, the housing box should have defined locations and features to properly hold down each motor controller, IMU, battery, and the Arduino.
- **Efficient software coding and data logging.** As mentioned previously, the software code for the accelerometer was copied in from a library and little effort was made to make the system perform more efficiently. The control-loop processing time was about 15ms, which seemed sufficient for interacting with the IMU and controlling the traction motor and linear actuator. However, once a data recording device (an SD card with a chip reader) was connected to the Arduino as well, the processing time dropped substantially, and it took an additional 0.5-0.75 seconds for each iteration of the stability code. The Arduino was programmed to completely shut down communications on each control iteration. Thus, each time the Arduino needed to write data to the SD card, it had to re-initialize communications with the SD card reader. The data write time slowed down the

system too much and the microcontroller was no longer able to iterate fast enough to stabilize the roll of the bicycle. Therefore, the roll angle and velocity data of the bicycle was captured with an iPhone 7 that was strapped to the bicycle during the test. The overall code structure of the controller should be evaluated. Any unnecessary functions or commands should be removed. Also, the SD card reader should be initialized at the very beginning of the test and stay linked to the Arduino throughout the test to enable fast and efficient data logging.

- **Higher performing electrical hardware and measurement equipment.** One of the objectives of this research was to maintain a small budget and develop a low-cost stability control system, if possible. While the team achieved this goal with the hardware and sensors selected, investing in something more than an entry-level microcontroller and IMU could yield higher performance and a more robust control solution. It would also provide the processing power to handle more complex control algorithms and handle auxiliary assignments that may be needed if the bicycle stability and propulsion system is integrated into a full autonomous vehicle control platform.
- **Dedicated test surfaces**. It was beneficial to have more than one person present when testing so that someone could set up the cameras and record observations while the other person launched the bicycle. Coordinating schedules was compounded by not have a dedicated test space. Sometimes it was difficult to find a smooth road or sidewalk surface where the traffic was low enough to be able to perform testing. During one test morning, the wider sidewalk and service road were close for resurfacing, and we were forced to run tests in the parking lot. The uneven nature of the asphalt and small cracks and holes made it difficult to know if the stability challenges were due to the controller or due to noise input from the road surface. If a dedicated test surface were available, it would make the testing time more efficient and controlled.

Besides the general recommendations listed above, the current prototype bicycle should be reinforced (possibly replaced by a manufactured e-bike) so that trials may continue with a functional propulsion system. A key next step would be to trial the control laws across a wider speed range. It would also be possible to test the control theory through CAE before more

experimental testing is performed. Although LPV control appears viable at this time, there are also a wide variety of other, more complex, control methods that may achieve similar, or improved, results.

Chapter 5 Conclusions and Considerations for Future Work

In this paper, I reported on the design evolution of the bicycle and other single-track systems and how they have become a key tool for people and goods transportation worldwide. The form factor, carrying capacity, maneuverability, and cost of single-track vehicles makes them advantageous in a variety of circumstances and justifies their use case in the 21st Century. As autonomous double-track vehicles arrive on public roads, it is natural that single-track autonomous systems will also be considered. Many of the functions of autonomous systems for automobiles can be directly applied to an inline wheel system as well. The unique challenge lies in the dynamics of the single-track vehicle.

The inherent instability and non-minimum phase dynamics of single-track vehicles poses a challenge from a controls perspective and has intrigued scientists and engineers for over a century. Although many researchers have provided commentary on the stability and tracking control of a riderless bicycle, relatively few bodies of work have validated their analysis through experimental testing. When a human rides on a bicycle, they learn to intuitively use a combination of lean and steering actuation to conduct the bicycle in a stable manner. The issue becomes more complicated when the rider is removed. Using solely steering or lean control comes has tradeoffs in maneuverability and each proves unreliable in certain scenarios (for example, steering actuation when the bicycle is at rest). Constructing an integrated controller has its own set of challenges as well.

Future applications of this research can expand beyond bicycle stability analysis. The stability methods here apply equally to motorcycles, mopeds, scooters, and other inline wheel systems. The μSMET project at the University of Michigan – Dearborn is another vehicle where single-track stability method described in this thesis can apply. The vehicle is a tricycle hybrid which can adjust the distance between its rear wheel axles. In its narrow configuration, the vehicle will behave much more like a single-track vehicle rather than a tricycle. Figure 33 shows the μSMET in its narrow and expanded configuration. The same control law used in this thesis would be a good starting point for active roll stabilization of the μSMET [62].

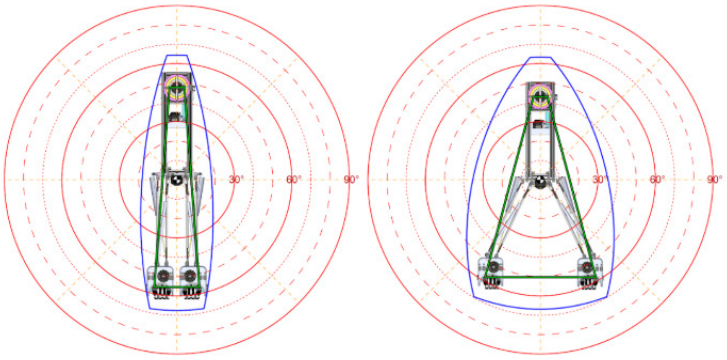

Figure 33: μSMET shown in its narrow (left) and expanded configuration (right).

In this thesis, we have successfully demonstrated that, through gain scheduling, a PID-type controller can achieve the self-balancing of an autonomous single-track vehicle by using a linear actuator to implement front-fork steering control. This control method is novel in the way in which the front fork is actuated. To the best of my knowledge, no other body of research outlines such use of a linear actuator. The manual PID tuning process outlined in this body of work is also unique, as well as the specifics of the control law (although others have used PID controllers).

The Linear-Parameter Varying gain scheduling approach has been shown to be successful at maintaining a single-track vehicle upright. Much work needs to be done to determine the path-

tracking capability of such a method. Some future research could include experimental validation of the controller across a wider speed range, trialing maintaining a bicycle in a steady-state turn, developing an accurate model of the full system so that less experimental testing is required, and exploring more advanced control methods.

Appendices

Appendix A Modeling Mass, Center of Gravity, and Inertia

The following appendix outlines how the bicycle system mass moment of inertia was calculated.

Assumptions:

- Assume all electronics components added to the bicycle are sufficiently small that their inertia may be modeled individually as a point mass, with the center of gravity at the center of the object.
- Additional components added to the bicycle may also be modeled as a point mass.
- The bicycle will be modeled as a rectangular plane (except for the powered wheel). The power wheel is considered a significant enough weight to be modeled separately as a point mass.
- The height of the center of gravity of the bicycle is approximately 70 percent of the height of the bicycle.

Process for finding the bicycle's center of gravity (ground plane projection):

- Measure the bicycle wheelbase.
- Measure the total weight of the bicycle.
- Measure the proportion of weight on each bicycle tire.
- Create a free-body diagram and a moment-balance equation to determine the CG location relative to the bicycle front and rear axles.

Process for calculating inertia:

- Each component of the bicycle was individually weighed.
- The height of each component (when installed) above the ground plane was measured.
- The standard engineering formula for the mass moment of inertia for point masses and beams was used to model the inertia of each system and the parallel axis theorem was used to calculate the mass moment of inertia about the ground plane [63], [64].
- Sum the mass and inertia of the bicycle and all the components.

Summary of values for the prototype bicycle:

Bicycle (no rear wheel)		
Mass	14.210	kg
CG Height	0.648	m
Assumptive Width of Rectangle	0.038	m
Assumptive Height of Cycle	0.925	m
Inertia	6.977	kg-m^2
Linear Actuator		
Mass	1.861	kg
Height above ground plane	0.850	m
Inertia	1.345	kg-m^2
48V Battery		
Battery Mass	5.130	kg
Height above ground plane	0.760	m
Inertia	2.963	kg-m^2
Powered Wheel		
Wheel Mass	7.500	kg
Height above ground plane	0.330	m
Inertia	0.817	kg-m^2
12V Battery		
Mass	1.362	kg
Height above ground plane	0.845	m
Inertia	0.973	kg-m^2
Motor Controller		
Mass	0.499	kg
Height above ground plane	0.830	m
Inertia	0.344	kg-m^2
Training Wheels		
Mass	2.724	kg
Height above ground plane	0.150	m
Inertia	0.061	kg-m^2
Total Mass:	**33.287**	**kg**
Total Inertia:	**13.480**	**kg-m^2**

Appendix B Ziegler Nichols PID Tuning Method

The original Ziegler Nichols paper was published in 1942 under the title "Optimum Settings for Automatic Controllers". An accounting of the presentation of the paper is in [65]. I was not able to find a direct link to the original paper. The process is described online in an open-source Wiki site maintained by the University of Michigan [66]. An excerpt is shown below:

Ziegler-Nichols Method [edit]

In the 1940's, Ziegler and Nichols devised two empirical methods for obtaining controller parameters. This was a popular manual method used for non-first order plus dead time situations; however, most manual methods are no longer used due to improved optimization software. Nonetheless, with computer aids, the following two methods are employed today and are the most common:

Ziegler-Nichols closed-loop tuning method [edit]

When using closed loop systems the ultimate gain value,K_u, and the ultimate period of oscillation,P_u, are used in order to calculate K_c. This method is used to obtain the controller constants K_c, T_I, and T_D, in a system with feedback. This technique allows for the tuning of processes that cannot run in an open-loop environment. The main objective of the Ziegler-Nichols closed-loop method is to find the value of the proportional-only gain that causes the control loop to oscillate indefinitely at a constant. This gain, which causes steady-state oscillations, is called the ultimate proportional gain (K_u). Another important value associated with this proportional-only control tuning method is the ultimate period (P_u). The ultimate period is the time required to complete one full oscillation once the response begins to oscillate at a constant amplitude. These two parameters, Ku and Pu, are used to find the loop-tuning constants of the controller (P, PI, or PID). To find the values of these parameters and to calculate the tuning constants, you must do the following:

Closed Loop (Feedback Loop) [ec

1. Remove integral and derivative action. Set integral time (T_I) to 999 or its largest value and set the derivative controller K_D to zero.
2. Create a small disturbance in the loop by changing the set point. Adjust the proportional, increasing and/or decreasing, the gain until the oscillations have constant amplitude.
3. Record the gain value K_u and period of oscillation Pu

Image:closedlooptune.jpg

Figure 1 System tuned using the Ziegler-Nichols closed-loop tuning method

4. Plug these values into the Ziegler-Nichols closed loop equations and determine the necessary setting for the controller.

Table 1. Closed-Loop Calculations of K_C, T_I, T_D

	K_c	T_I	T_D
P	$K_u/2$		
PI	$K_u/2.2$	$P_u/1.2$	
PID	$K_u/1.7$	$P_u/2$	$P_u/8$

References

[1] E. Salmeron-Manzano and F. Manzano-Agugliaro, "The Electric Bicycle: Worldwide Research Trends," *Energies*, vol. 11, 2018, doi: 10.3390/en11071894.

[2] Z. Liu, L. Ma, T. Huang, and H. Tang, "Collaborative Governance for Responsible Innovation in the Context of Sharing Economy: Studies on the Shared Bicycle Sector in China," *J. Open Innov. Technol. Mark. Complex. Artic.*, vol. 6, 2020, doi: 10.3390/joitmc6020035.

[3] "Technology, Media, and Telecommunications Predictions 2020," 2020. Accessed: Jun. 20, 2021. [Online]. Available: https://www2.deloitte.com/content/dam/insights/us/articles/722835_tmt-predictions-2020/DI_TMT-Prediction-2020.pdf.

[4] "Italian bicycle sales 'surpass those of cars' - BBC News," *BBC News*, 2012. https://www.bbc.com/news/world-europe-19801599 (accessed Jun. 20, 2021).

[5] "Bicycle," *Wikipeida*. https://en.wikipedia.org/wiki/Bicycle (accessed Apr. 03, 2021).

[6] "Timeline of bicycle transportation - Timelines," *Wikipedia*. https://timelines.issarice.com/wiki/Timeline_of_bicycle_transportation (accessed Apr. 03, 2021).

[7] "Bicycle suspension," *Wikipedia*. https://en.wikipedia.org/wiki/Bicycle_suspension#History (accessed Apr. 03, 2021).

[8] "Rover 'Safety' bicycle, 1885 | Science Museum Group Collection." https://collection.sciencemuseumgroup.org.uk/objects/co25833/rover-safety-bicycle-1885-bicycle (accessed May 14, 2020).

[9] "Sun Bicycles Skylar 5 - Bicycle Renaissance Bike Shop in New York City on the Upper West Side." https://www.bicyclerenaissance.com/product/sun-bicycles-skylar-5-263384-1.htm (accessed May 14, 2020).

[10] T. Hadland, *Bicycle Design: An Illustrated History*. Cambridge: MIT Press Books, 2014.

[11] "Trail-Breaker - Rokon." https://www.rokon.com/product/trail-breaker/ (accessed Apr. 03, 2021).

[12] K. J. Astrom, R. E. Klein, and A. Lennartsson, “Bicycle Dynamics and Control: Adapted bicycles for education and research,” *IEEE Control Syst.*, vol. 25, no. 4, pp. 26–47, 2005, doi: 10.1109/MCS.2005.1499389.

[13] S. Pendleton *et al.*, “Multi-class driverless vehicle cooperation for mobility-on-demand,” *21st World Congr. Intell. Transp. Syst. ITSWC 2014 Reinventing Transp. Our Connect. World*, no. September 2016, 2014.

[14] S. Behere and M. Törngren, “A functional reference architecture for autonomous driving,” *Inf. Softw. Technol.*, vol. 73, pp. 136–150, 2016, doi: 10.1016/j.infsof.2015.12.008.

[15] A. Nardi and A. Armato, “Functional safety methodologies for automotive applications,” *IEEE/ACM Int. Conf. Comput. Des. Dig. Tech. Pap. ICCAD*, vol. 2017-Novem, pp. 970–975, 2017, doi: 10.1109/ICCAD.2017.8203886.

[16] Y. Jung, S. Park, and M. Han, “Automotive Hardware Development According to ISO 26262,” *13th Int. Conf. Adv. Commun. Technol.*, pp. 588–592, 2011.

[17] M. Nagai, “Analysis of Rider and Single-track-vehicle System,” *Automatica*, vol. 19, no. 6, pp. 737–740, 1983.

[18] J. D. G. Kooijman and A. L. Schwab, “A review on bicycle and motorcycle rider control with a perspective on handling qualities,” *Veh. Syst. Dyn.*, vol. 51, no. 11, pp. 1722–1764, 2013, doi: 10.1080/00423114.2013.824990.

[19] S. Miyagishi, I. Kageyama, K. Takama, M. Baba, and H. Uchiyama, “Study on construction of a rider robot for two-wheeled vehicle,” *JSAE Rev.*, vol. 24, no. 3, pp. 321–326, 2003, doi: 10.1016/S0389-4304(03)00045-6.

[20] A. L. Schwab and J. P. Meijaard, “A review on bicycle dynamics and rider control,” *Veh. Syst. Dyn.*, vol. 51, no. 7, pp. 1059–1090, 2013, doi: 10.1080/00423114.2013.793365.

[21] T. Saguchi, K. Yoshida, and M. Takahashi, “Stable running control of autonomous bicycle robot,” *Nihon Kikai Gakkai Ronbunshu, C Hen/Transactions Japan Soc. Mech. Eng. Part C*, vol. 73, no. 7, pp. 2036–2041, 2007, doi: 10.1299/kikaic.73.2036.

[22] B. Michini and S. Torrez, “Autonomous Stability Control of a Moving Bicycle,” *Am. Inst. Aeronaut. Astronaut.*, pp. 1–10, 2006.

[23] J. F. Lenkeit, “A servo rider for the automatic and remote path control of a motorcycle,” *SAE Tech. Pap.*, 1995, doi: 10.4271/950199.

[24] Z. Fawaz, R. Smith, P. Muench, S. Lakshmanan, and A. Mohammadi, “Design and benchtop validation of an autonomous bicycle with linear electric actuators,” p. 11, 2019, doi: 10.1117/12.2519189.

[25] M. Yamakita and A. Utano, “Automatic Control of Bicycles with a Balancer,” in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2005, pp. 24–28.

[26] L. Keo and M. Yamakita, “Controller Design of an Autonomous Bicycle with Both Steering and Balancer Controls,” in *18th IEEE International Conference on Control Applications*, 2009, pp. 1294–1299.

[27] V. Cerone, D. Andreo, M. Larsson, and D. Regruto, “Stabilization of a Riderless Bicycle: A Linear-Parameter-Varying Approach,” *IEEE Control Syst.*, vol. 30, no. 5, pp. 23–32, 2010, doi: 10.1109/MCS.2010.937745.

[28] D. Andreo, V. Cerone, D. Dzung, and D. Regruto, “Experimental results on LPV stabilization of a riderless bicycle,” *Proc. Am. Control Conf.*, pp. 3124–3129, 2009, doi: 10.1109/ACC.2009.5160397.

[29] I. I. Boldea and S. A. Nasar, “Linear electric actuators and generators,” *IEEE Trans. Energy Convers.*, vol. 14, no. 3, pp. 712–717, 1999, doi: 10.1109/60.790940.

[30] B. Na, H. Choi, and K. Kong, “Design of a Direct-Driven Linear Actuator,” *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 99, pp. 1–10, 2014, doi: 10.1109/TMECH.2014.2326696.

[31] M. Bergamasco, F. Salsedo, and P. Dario, “Linear SMA Motor as Direct-Drive Robotic Actuator,” in *1989 International Conference on Robotics and Automation*, 1989, pp. 618–623, doi: 10.1109/ROBOT.1989.100053.

[32] F. Sup, A. Bohara, and M. Goldfarb, “Design and control of a powered transfemoral prosthesis,” *Int. J. Rob. Res.*, vol. 27, no. 2, pp. 263–273, 2008, doi: 10.1177/0278364907084588.

[33] B. Na, H. Choi, and K. Kong, “Design of a Direct-Driven Linear Actuator for a High-Speed Quadruped Robot, Cheetaroid-I,” *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 99, pp. 1–10, 2014.

[34] P. Lucidarme, N. Delanoue, F. Mercier, Y. Aoustin, C. Chevallereau, and P. Wenger, “Preliminary Survey of Backdrivable Linear Actuators for Humanoid Robots,” *CISM Int. Cent. Mech. Sci. Courses Lect.*, vol. 584, no. February, pp. 304–313, 2019, doi: 10.1007/978-3-319-78963-7_39.

[35] J. Hollerbach, I. Hunter, and J. Ballantyne, “A Comparative Analysis of Actuator Technologies for Robotics,” in *The Robotics Review 2*, MIT Press, 1992, pp. 299–342.

[36] T. Ishida and A. Takanishi, “A Robot Actuator Development With High Backdrivability,” in *2006 IEEE Conference on Robotics, Automation and Mechatronics*, Jun. 2006, pp. 1–6, doi: 10.1109/RAMECH.2006.252631.

[37] P. M. Wensing, A. Wang, S. Seok, D. Otten, J. Lang, and S. Kim, "Proprioceptive Actuator Design in the MIT Cheetah: Impact Mitigation and High-Bandwidth Physical Interaction for Dynamic Legged Robots," *IEEE Trans. Robot.*, vol. 33, no. 3, pp. 509–522, Jun. 2017, doi: 10.1109/TRO.2016.2640183.

[38] "SUV(HATCHBACK) Soft Close Automatic Door Suction | Changyi." http://changyi-china.com.cn/product/automatic-tailgate-lift-for-suv-electric-suction/ (accessed Apr. 08, 2021).

[39] "2013-2019 Ford Flex Actuator DA8Z-14B351-A | Ford Discount Parts," *Ford Discount Parts*. https://www.forddiscountparts.com/oem-parts/ford-actuator-da8z14b351a?origin=pla&gclid=CjwKCAjw07qDBhBxEiwA6pPbHnfqR738Oxrm93h2sVUNYYI68GmP6INk4GajI_8Gq2Awk77cVZo8oBoC93YQAvD_BwE (accessed Apr. 08, 2021).

[40] Z. Fawaz, R. Smith, P. Muench, S. Lakshmanan, and A. Mohammadi, "Design and benchtop validation of an autonomous bicycle with linear electric actuators," in *Proceedings of SPIE - The International Society for Optical Engineering*, 2019, vol. 11021, doi: 10.1117/12.2519189.

[41] T. Newcombe, "Amid Cycling Surge, Sport Of Mountain Biking Is Seeing Increased Sales And Trail Usage," *Forbes Magazine*, 2020. https://www.forbes.com/sites/timnewcomb/2020/07/13/amidst-cycling-surge-sport-of-mountain-biking-seeing-increased-sales-trail-usage/?sh=53ce6e423ddf (accessed Apr. 03, 2021).

[42] B. Beacham, "5 Best Electric Bikes - Apr. 2021 - BestReviews," *Daily News*, 2021. https://reviews.nydailynews.com/reviews/best-electric-bikes (accessed Apr. 03, 2021).

[43] "Amazon.com : electric motorbike kit." https://www.amazon.com/electric-motorbike-kit/s?k=electric+motorbike+kit (accessed Apr. 03, 2021).

[44] R. P. B. H. Center, "Riding Rad with Regenerative Braking," *Rad Power Bikes*, 2021. https://radpowerbikes.zendesk.com/hc/en-us/articles/360045171734-Riding-Rad-with-Regenerative-Braking (accessed Apr. 03, 2021).

[45] "48V 1000W Front Wheel E-Bike Motor Conversion Kit," *Jaxpety.com*. https://www.jaxpety.com/bicycle-motor-kit-48v-1000w-rear-wheel-electric-conversion.html.

[46] "MPC 3620 Heavy Duty Linear Actuator, 12V DC, 10" Stroke, 770 lb. Max Load, Black Finish: Amazon.com: Automotive." https://www.amazon.com/MPC-3620-Linear-Actuator-Stroke/dp/B012YAYGVC (accessed Nov. 28, 2020).

[47] “MPC LAD-HS10 High Speed Linear Actuator with 10" Stroke, Heavy 12V, 65 mm/seconds Speed: Amazon.com: Automotive.” https://www.amazon.com/MPC-LAD-HS10-Linear-Actuator-seconds/dp/B00JYOIZ64/ref=cm_cr_arp_d_product_top?ie=UTF8 (accessed Nov. 28, 2020).

[48] “Arduino Uno Webpage.” https://store.arduino.cc/usa/arduino-uno-rev3 (accessed May 12, 2020).

[49] Martin Marino, “Arduino vs Microprocessor vs Microcontroller,” *StackExchange - Electronics*, 2014. https://electronics.stackexchange.com/questions/99434/arduino-vs-microprocessor-vs-microcontroller (accessed May 12, 2020).

[50] InvenSense, “MPU-6050 Product Specification Revision 3.4.” pp. 1–11, 2013.

[51] Arduino.cc, “MPU-6050 Accelerometer + Gyro.” https://playground.arduino.cc/Main/MPU-6050/ (accessed Feb. 10, 2018).

[52] J. Rowberg, “MPU-6050 6-axis accelerometer/gyroscope.” https://www.i2cdevlib.com/devices/mpu6050#source.

[53] J. Rowberg, “I2C Device Library,” *GitHub*, 2014. https://github.com/jrowberg/i2cdevlib (accessed Feb. 10, 2019).

[54] M. Carvallo, “Theorie De Mouvement Du Monocycle Et De La Bycyclette.” Gauthier-Villars, Paris, 1901, [Online]. Available: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Theorie+du+mouvement+du+Monocycle+et+de+la+Bicyclette#0.

[55] T. Tun, L. Rothenbusch, P. Ingenlath, A. Brezing, and B. Corves, “Modelling , Implementation and Analysis of the Carvallo-Whipple Bicycle Model in Msc Adams,” in *IASTEM International Conference*, 2018, no. August, pp. 19–23, doi: 10.13140/RG.2.2.22633.75369.

[56] “Understanding Bike Geometry,” *GeometryGreeks.BIKE*. https://geometrygeeks.bike/understanding-bike-geometry (accessed Jun. 20, 2021).

[57] D. Karnopp, *Vehicle Dynamics, Stability, and Control*, 2nd ed. Boca Raton: CRC Press, 2013.

[58] K. J. Åström and T. Hägglund, “Revisiting the Ziegler-Nichols step response method for PID control,” *J. Process Control*, vol. 14, no. 6, pp. 635–650, 2004, doi: https://doi.org/10.1016/j.jprocont.2004.01.002.

[59] V. Bobál, J. Macháček, and R. Prokop, “Tuning of Digital PID Controllers Based on Ziegler - Nichols Method,” *IFAC Proceedings Volumes*, vol. 30, no. 21. pp. 145–150, 1997, doi: 10.1016/s1474-6670(17)41430-3.

[60] R. Sen, C. Pati, S. Dutta, and R. Sen, “Comparison Between Three Tuning Methods of PID Control for High Precision Positioning Stage,” *MAPAN*, vol. 30, no. 1, pp. 65–70, 2015, doi: 10.1007/s12647-014-0123-z.

[61] Felix, “control - What are good strategies for tuning PID loops? - Robotics Stack Exchange,” *Robotics Stack Exchange*, 2013. https://robotics.stackexchange.com/questions/167/what-are-good-strategies-for-tuning-pid-loops (accessed Apr. 02, 2021).

[62] C. Adam, T. Kleinow, K. Grenn, B. Mason, O. Sapunkov, and P. Muench, “µSMET : A Lightweight Transport Robot,” in *SPIE Defense + Commercial Sensing*, 2021, vol. 11758, no. 1175807.

[63] “Mass Moment of Inertia Equations | Engineers Edge | www.engineersedge.com.” https://www.engineersedge.com/mechanics_machines/mass_moment_of_inertia_equations_13091.htm (accessed Jun. 21, 2021).

[64] “Parallel & Perpendicular Axis Theorem: Statement, Formula, Derivation.” https://byjus.com/physics/parallel-perpendicular-axes-theorem/ (accessed Jun. 21, 2021).

[65] J. G. Ziegler and N. B. Nichols, “Optimum Settings for Automatic Controllers,” *J. Dyn. Syst. Meas. Control*, vol. 115, no. 2B, pp. 220–222, Jun. 1993, doi: 10.1115/1.2899060.

[66] “PIDTuningClassical - ControlsWiki.” [Online]. Available: https://web.archive.org/web/20080616062648/http://controls.engin.umich.edu:80/wiki/index.php/PIDTuningClassical#Ziegler-Nichols_Method.