Proceedings of the ASME 2018
Dynamic Systems and Control Conference
DSCC2018
September 30-October 3, 2018, Atlanta, Georgia, USA

DSCC2018-9244

TOWARDS AUTOMATED BICYCLES: ACHIEVING SELF-BALANCE USING STEERING CONTROL

Wenhao Deng, Skyler Moore, Jonathan Bush, Miles Mabey, Wenlong Zhang*
The Polytechnic School
Ira A. Fulton Schools of Engineering
Arizona State University
Mesa, Arizona 85212
Email: [wdeng16, sdmoore9, jonbush, miles.mabey, wenlong.zhang]@asu.edu

ABSTRACT

In recent years, researchers from both academia and industry have worked on connected and automated vehicles and they have made great progress toward bringing them into reality. Compared to automated cars, bicycles are more affordable to daily commuters, as well as more environmentally friendly. When comparing the risk posed by autonomous vehicles to pedestrians and motorists, automated bicycles are much safer than autonomous cars, which also allows potential applications in smart cities, rehabilitation, and exercise. The biggest challenge in automating bicycles is the inherent problem of staying balanced. This paper presents a modified electric bicycle to allow real-time monitoring of the roll angles and motor-assisted steering. Stable and robust steering controllers for bicycle are designed and implemented to achieve self-balance at different forward speeds. Tests at different speeds have been conducted to verify the effectiveness of hardware development and controller design. The preliminary design using a control moment gyroscope (CMG) to achieve self-balancing at lower speeds are also presented in this work. This work can serve as a solid foundation for future study of human-robot interaction and autonomous driving.

1 INTRODUCTION

Connected and autonomous vehicles have become a hot research topic recently due to the potential benefits of reduced vehicle fatalities and injuries, reduced carbon dioxide emissions, increased vehicle energy efficiency, improved accessibility to transportation and so on [1]. Many automakers and self-driving companies, such as Waymo and Tesla, have announced plans to have driverless cars on the road by 2021. Compared to cars, bicycles have many advantages in both commercial and research cases. For example, bicycles have more maneuverability in cities, which makes the bicycle a good tool for solving modern mobility challenges in smart cities. Bicycles are also more affordable and environmentally friendly, which has led to a rapid global development in bike sharing systems [2]. Moreover, bicycles are safer for pedestrians and other road users due to their light weights and relatively low speeds compared to cars and motorcycles. Besides the potential advantages in commercial world, bicycles also have great potential in research and academia. For instance, a bicycle is a unique platform to study physical human-machine interaction [3] because there are many contacting points between a bicycle and its rider, such as the peddles, seat, and handle bars, which enables great potential to attach different and multiple sensors to track human activities. Therefore, one can use riding to study physical human-robot interactions (pHRIs) and human-in-the-loop control systems [4, 5]. This study is also beneficial for understanding how to apply artificial intelligence techniques on motion planning and human intention detection [6]. Consequently, bicycles are an effective tool for rehabilitation with a great potential for wearable assistive devices to be integrated in this loop. Researchers can also use this platform to study interactions between bicycles and bicycles, pedestrians, cars, and even unmanned aerial vehicles for different civil and military applications. This study can also help researchers gain a better understanding of human intent in order to predict human rider behavior

*Address all correspondence to this author.

in advance [7], which will help achieve safer and more convenient bicycle riding. The researchers chose to not investigate a tricycle type bike because it does not have the same dynamics as a two-wheeled bike and as such is not a suitable platform for human-bike interaction research based on balance. To make bicycles driverless, the biggest challenge is to solve the inherent problem of balancing since bicycles are not able to balance by themselves [8]. In this study, a hardware prototype is designed and built, and reliable control strategies to make the bicycle self-balance are explored.

2 REVIEW OF RELATED WORK

Related work about driverless bikes can be classified into study of how to self-balance bicycles, how to make bicycles driverless and understanding of human-bicycle interactions. The goal of this section is to introduce related work in these topics.

2.1 Bicycle Self-balancing

Methods to achieve a self-balanced bicycle can be classified into four types based on the literature review:

Control Moment Gyroscope: A CMG is an attitude control device often used in spacecraft attitude control [9]. A typical CMG is made of one spinning rotor and one motorized gimbal that can change the rotor's angular momentum, which can generate precessive torque to balance the bicycle. Many researchers have utilized CMGs to balance bicycles [10,11] since CMGs can produce effective moments on a body to provide balance at low forward speed. While a CMG can provide a large amount of torque, the energy required to maintain a constant high flywheel speed is a major drawback of this method.

Mass Balancing: Bicycles can also balance by using a massive balancer on the bicycle to achieve self-balance [12, 13]. The mechanical structure required for mass balancing is significantly simpler compared to CMG, but it can only provide a small amount of torque.

Steering Control: Steering control is a method to achieve real time control of bicycle balance by adjusting the steering system in real time [14, 15]. Steering control consumes significantly lower energy compared to a CMG, but this method cannot adequately provide balance at low forward velocities.

Reaction Wheel: Murata Manufacturing utilizes a reaction wheel to balance a bicycle and produces the well-known self-balancing robot bicycle called Murata Boy. The response time of the reaction wheel is short while the output torque is limited, so it is only suitable for balancing a small bicycle.

Table 1 is a summary of advantages and disadvantages for these four self-balancing techniques. In this study, the bicycle is massive and its center of mass is high, resulting in the necessity of a large amount of torque with quick response times. Therefore, the steering control method was chosen to achieve fast response while moving with sufficient forward velocity and a CMG system will be incorporated in future designs in order to balance the system at low forward velocities.

TABLE 1: Summary of advantages and disadvantages for different self-balancing techniques.

	CMG	Steering Control	Reaction Wheel	Mass Balancing
Advantages	Large torque	Low energy consumption, Real time response	Short response time	Simple mechanical structure
Disadvantages	High energy consumption	No balance at low velocities	Limited torque output	Small torque

2.2 Driverless Bicycles

The research about autonomous driving has raised a lot of interest recently, especially in autonomously driving cars. However, there is limited research that focuses on driverless bicycles. Based on a survey of existing literature, several typical projects about driverless bicycles are discussed below. A power-assisted electric bicycle was designed by researchers from University of California, Berkeley for uphill riding. Control algorithms in the system can provide flexible assistive power through estimating and compensating for the changing environmental disturbances [16]. Automotive collision avoidance systems have been developed by many automotive manufacturers and a sensing system for bicyclist-motorist crash prediction is developed to accurately track rear vehicles that can have two-dimensional motion [7]. In addition, vision-based motion planning systems for an autonomous motorcycle has been designed for desert terrain, where uniform road surface and lane markings are not present [17]. These works only address some part of the real self-driving bicycle systems. Cameras, millimeter wave radars, ultrasonic sensors and low-cost laser radar (LiDAR) are typically used in research and commercial self-driving cars [18]. A LiDAR, an RGB camera, and other sensors have already been installed on the prototype developed in this study and will be applied in the future to achieve environmental awareness, allowing for detection and modeling of areas that could not be accessed by cars. Finally, deep learning algorithms about self-driving and human robot interaction can be developed and studied based on the huge amount of data the bicycle can collect.

3 HARDWARE DEVELOPMENT

The hardware for this project was selected and designed to create a platform that a robust control algorithm can be implemented on and that can be expanded for future work. A pre-built electric bicycle, the X-Treme Trail Maker Elite Mountain Bike, was selected based on several features. The hub motor on this bicycle is a 300 W electric motor. This power will allow the bicycle to carry the extra weight, besides the human, required for actuators and research equipments [16]. The location and orientation of the battery was an important consideration as bicycles were benchmarked. The X-Treme Elite bike includes a battery placed between the seat and the back tire, which leaves the center triangle and back rack free for research hardware. The battery pack is also removable, which allows researchers to conduct longer studies by having multiple batteries. The overall design of the modifications to the original bicycle are illustrated in Figure 1.

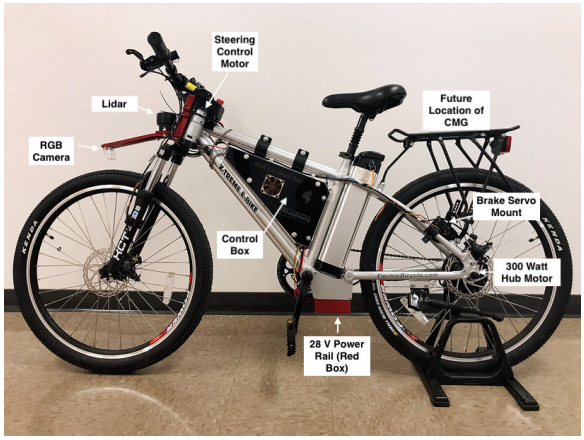


FIGURE 1: Self-balancing bike prototype.

3.1 Bicycle Modifications

In order to implement a control algorithm for self balancing, certain parts of the selected electric bicycle need to be modified. The bike's throttle signals are transferred from the handlebar to the microcontroller and the bike is equipped with a servo attached to the rear brakes, allowing the platform to be remotely braked. In order to better understand the real time dynamics of the bicycle, hall effect sensors are mounted on the fork of each wheel and three magnets affixed to the spokes. This allows the microcontroller to accurately calculate the speed of the bike.

In addition, a future modification will automate the bicycle kickstand using custom 3D printed parts and a micro servo. This is intended so the bicycle can "rest" upright without the CMG activated. In order to accomplish this, the factory kickstand has been modified so the mechanism requires less force to move to allow the stand to be actuated by a microservo. This servo will then be implemented in the overall bike code so the bike could be set onto the kickstand autonomously.

3.2 Microcontroller and Electronics Mounting

A box to house the electronics was designed and 3D printed. The box is designed to fit in the center triangle of the bicycle without interfering with pedaling. The placement of this box allows for central control of all components and allows the CMG to be mounted on the rear rack of the bicycle. The box has a hole at each corner of the triangle to allow wires to be routed easily. The four points of contact (two upper clamps around the frame and two screws into the lower portion of the frame) have proven to be sufficient to hold the box securely even on uneven terrain or in the event of a crash. This box includes mounting points for all of the electronics that act as standoffs to allow for clean wiring to all of the components resulting in easier troubleshooting and reliable operation. The lid of this box has mounting points for a 60mm fan which provides cooling for the components inside. Figure 2 shows the model of the central logic mounting box detailing each of the components.

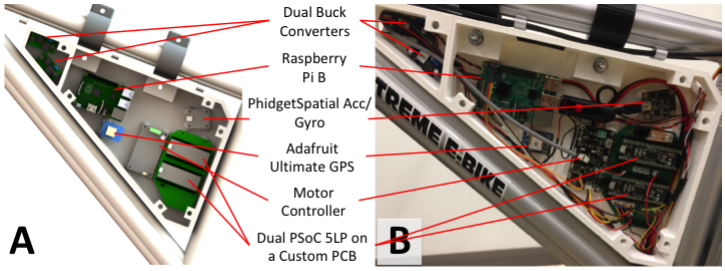


FIGURE 2: View A shows the CAD model with components labeled of the control box. View B shows the real world implementation.

3.3 Electrical Components

The high-level control of the bike system and image processing is done on a Raspberry Pi Model 3 B (Raspi) programmed in Python. The Raspi uses WiFi to communicate with a custom user interface on an Android tablet. Once complete, this interface will allow researchers to see real-time information about the bike such as a speed, direction, and GPS location on a map. Researchers can also tune certain control parameters such as the PID control gain. The Raspi communicates over USB with a PhidgetSpatial 3/3/3 IMU, which measures the specific force (acceleration), angular rates, and magnetic field in the bicycle frame. In order to enhance the navigation of the bike, the Raspi gets data

from an Adafruit GPS module over UART. For the low-level control, a Cypress PSoC 5LP microcontroller communicates with the Raspi over a serial connection. The PSoC controls the servos and motor controller for the steering motor. All the electronic components run on 5 V, so it was necessary to create a 5 V rail. Two buck converters are used to supply enough amperage. All of these connections are illustrated in Figure 3.

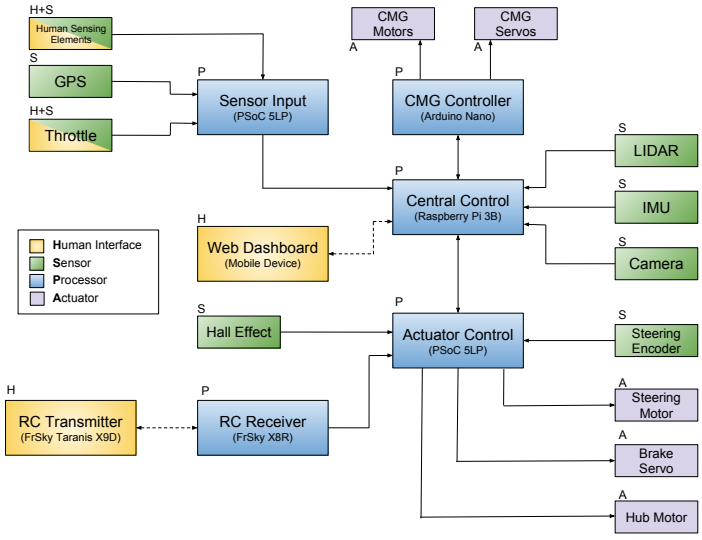


FIGURE 3: Block diagram for the smart bike's electrical systems.

3.4 Steering Control Assembly

All of the parts used to mount the steering hardware are 3D printed in order to allow for rapid prototyping and iteration. The mounting hardware for the steering motor and encoder includes a housing that covers the gears, thus increasing safety for the end user and providing a mounting surface and wiring support for the motor, encoder, and emergency stop button. The design as seen in Figure 4, is much more compact compared to that of other bicycles with steering control. Other designs typically replace the handle bars with a large motor mounted concentrically with the steering shaft—preventing any rider from being able to control the bike manually. Our design features a motor mounted parallel to the steering column and below the gear attached to the steering column. This allows a rider to ride the bike as though it were unmodified. In order to effectively control the steering, modified steel gears, with a 20° pressure angle, are attached to the motor shaft and to the steering column just below the handle bars. The steering gear is attached securely using a set screw. The 3D printed mounting hardware also includes a brace to mount the LiDAR unit and the Raspberry Pi camera extended on an armature. The LiDAR will be used to detect obstacles in the path of the bike and the camera will be used to identify lines on the road allowing for automation on paved roads. The lower support for the LiDAR mount also attaches to the bottom of the motor in order to help constrain the motor to the proper angle and provide stability. The steering motor is a 24 V DC motor with an integrated hall effect encoder, which provides 1472 oz-in of torque at 143 rpm. With our gear ratio of 1:0.6 this setup equates to the motor adjusting the angle of the front tire by 0.516 degrees per millisecond. This is the second iteration of the motor as the first did not have sufficient torque and only operated on 12 volts. It was observed that the first motor, which had a rated torque of 583 oz-in at 100 rpm, could move the tire when it was suspended and had no weight on it. The second iteration of motor was selected because it has nearly triple the torque and a faster speed as well as 24 V operating voltage, which allows it to be powered directly from the main battery. Several of these components such as the LiDAR and RGB Camera are not in use at the time of writing but have been integrated and mounted for future use by the researchers.

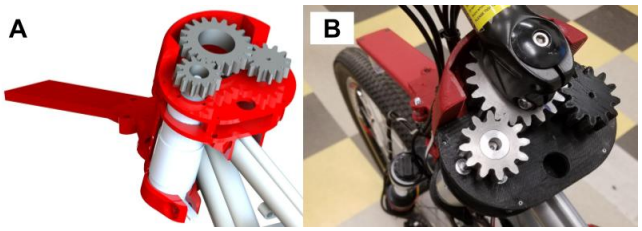


FIGURE 4: View A depicts the CAD model of the steering assembly, View B depicts the real world implementation of the model. These views do not include the designed gear guard

3.5 CMG Development

Above the back wheel of the bike, a box containing a pair of CMGs will be mounted. These CMGs will provide the balancing actuation for the bike at low forward speeds. The use of two CMGs allows the system to produce a larger precessive torque than using a single CMG, allowing the system to not only balance itself, but also to assist with balancing with a human rider. The design of the CMGs can be seen in Figure 5 which displays an exploded view (A), a cross sectional view through the middle of the assembly (B), and an assembled view (C). The box and gimbals are made from sheet metal bent and welded into shape. The gimbals will be turned by a sprocket and chain which will be actuated using a high torque servo. The motors and microcontroller for the CMG will be cooled using two 60mm intake fans near the front and a 140mm exhaust fan on the rear of the box. Each of the CMG's gimbals will be equipped with a 15 pound (6.8 Kg) flywheel that will be spun at 8000 rpm in order to produce the torque needed for the CMG to provide the balancing

force. This was calculated using standard equations developed for mini-CMG's in satellites [9]. It was found that with a 250 lb human on the bike a restoring force of 236 Newtons would be needed. In addition to the CMGs, the box will also contain the circuitry necessary to run the gimbal servos and motors as well as an additional bank of three four-cell 10 Ah Li-Po batteries to power both the circuitry and motors. Adding these additional batteries will allow the bike to be used for longer periods without needing to be recharged. These batteries will also decrease the spin up time of the flywheels due to the increased draw rate available from these batteries. The design of the box also allows for expansion on the lid, which may allow for future work to be performed.

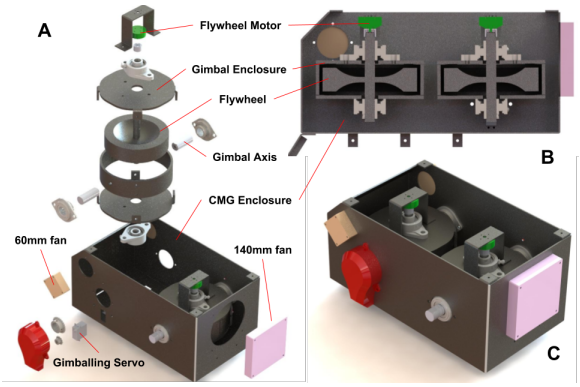


FIGURE 5: Model of the CMG assembly, view A is the exploded view of the assembly, View B is a section view of the center of the assembly, and View C is the isometric view of the assembly.

4 Software Development

The software system is divided into four main subsystems: high-level control, actuator control, human sensing, and remote user interface. The modular design of the system improves the ease of modification and enables adding high-level control features without modifying other areas of the software. This software system is implemented entirely on low-cost commercial hardware, using the C and Python programming languages.

The high-level control computations performed on the Raspberry Pi 3 Model B are responsible for balancing and navigating the bicycle. The inputs to this system include data from the IMU, LiDAR, and RGB camera sensors, with additional inputs from the other three subsystems. These data are fed to several asynchronous processes which plan the bicycle path. The resulting roll angle setpoint is fed into the balance control program, which adjusts the target steering angle to achieve the desired roll angle corresponding to the turn.

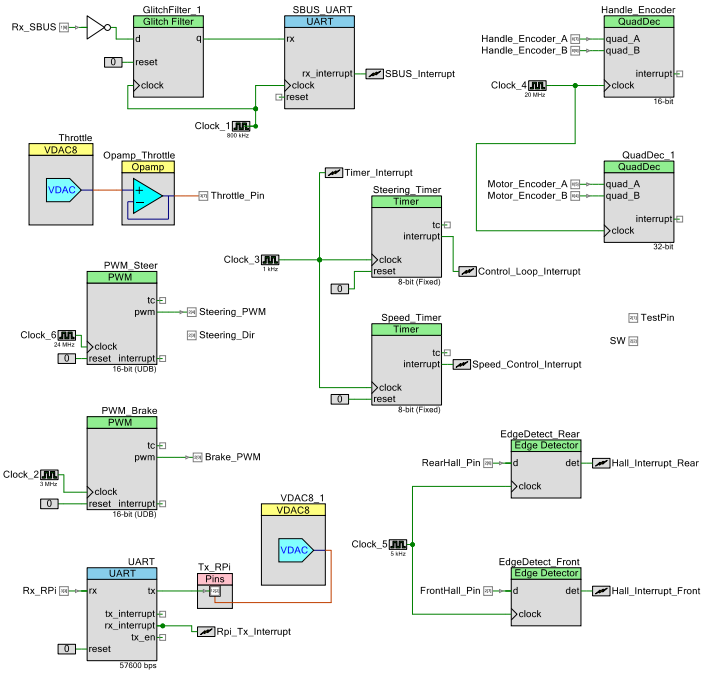


FIGURE 6: Components defined in the PSoC "Top Design" for the low-level actuator control.

Low-level control of the bicycle actuators is performed on one of the two Cypress PSoC 5LP microcontrollers. This microcontroller receives target values for the steering angle, bicycle speed, and braking amount over a serial connection to the Raspberry Pi, and adjusts the actuators to achieve these targets. The PSoC 5LP is an ideal microcontroller for this application because its hardware can be configured to perform tasks that would otherwise require additional computational resources. The 32-bit Arm Cortex-M3 CPU in the PSoC 5LP also has ample computational capacity to run the several interrupt-based control loops required to manage the bicycle speed and steering angle. The actuator PSoC also periodically sends data about the current conditions of the bicycle back to the Raspberry Pi for use in the high-level control algorithms. The "Top Design" defining the PSoC input/output structure is illustrated in Figure 6.

The user interface subsystem contains two primary components: an RC transmitter used to provide reliable manual input to the bicycle control system during operation, and a tablet or other computer used to display real-time information about the bicycle system. The RC transmitter communicates with the Smart Bike via a 2.4 GHz RC receiver connected to the actuator control PSoC via the Futaba S.Bus protocol. This direct connection, bypassing the Raspberry Pi, ensures reliability in case of problems with the high level control. If a dangerous situation should arise during testing, the actuators on the bicycle (throttle and brake) can be operated remotely. The Raspberry Pi in the bicycle pro-

vides a wireless network that mobile devices can connect to in order to display real-time data.

5 STEERING CONTROLLER AND HARDWARE EXPERIMENTS

The bicycle dynamics are fully defined by 25 parameters [19]. Many assumptions and linearizations need to be made in order to apply model-based control, and the effectiveness of the controller can not be guaranteed due to error in measuring the parameters on the bike, such as moments of inertia. The model-based controller is difficult to transfer to a different bicycle due to the difficulties of accurately measuring all the dynamic parameters.

Instead of model-based control, model-free control methods design the controlled system without any explicit information about the model itself and it can be more easily applied on different hardware platforms once the effectiveness has been verified. Proportional-integral-derivative (PID) controllers are one of the model-free control approaches and have been widely used in control engineering practice for several decades. The biggest advantages of the PID controller are that it can be tuned and adjusted on-site by experiment on the controller plant and fine tuning of the controller can be achieved based on tuning rules. Figure 7 illustrates the overall control systems design, with the high level PID balance control and the low level PID actuator control on the bicycle. The input into the system is the roll angle of the bicycle measured by IMU and the output is the steering angle controlled by the steering motor.

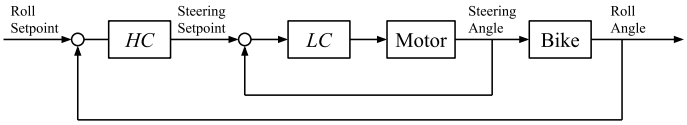


FIGURE 7: Block diagram for the balance control system. *HC* indicates the high level balance control, and *LC* indicates the low level actuator control.

The PID controller for balancing is implemented in Python on the Raspberry Pi. The current angular rates are captured from the PhidgetSpatial IMU sensor at a frequency of 200 Hz and are integrated, using the Runge-Kutta fourth-order method, to approximate the current roll angle. Sensor drift is reduced by passing the approximate roll angle determined from the high frequency IMU data through a complementary filter [20] with the roll angle calculated from the low-frequency accelerometer data. Readings from the accelerometer are filtered by discarding any data where the component of acceleration normal to the bicycle's direction of motion is outside (0.8, 1.2) g. The low-level steering motor controller is implemented on the actuator control PSoC and is driven by a timer at 200 Hz.

During the PID tuning process, the target roll angle is set to zero and the target speed is set based on the velocity to be tested. The balance controller is configured such that the control parameters can be adjusted from the RC transmitter during bicycle operation, and these parameters are initialized based on extrapolation of the values obtained from previous tests. For example, to determine the initial control parameters for 2.6 m/s motion of the bicycle, Kp was initialized to 1.0 and gradually increased until the bicycle could maintain balance for several seconds. However, introduction of the derivative term was necessary to achieve more stable operation of the bicycle for longer periods of running time.

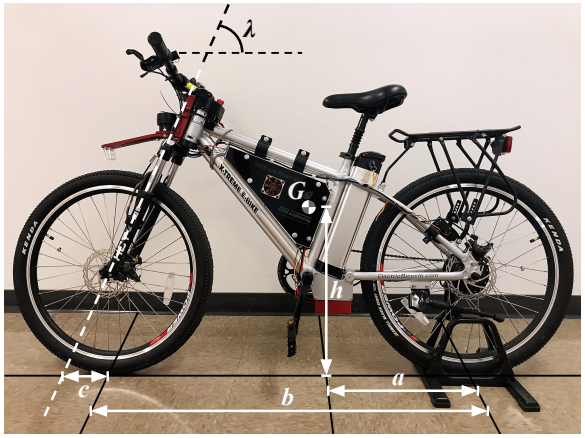


FIGURE 8: Bicycle with parameters.

TABLE 2: Summary of bicycle parameters.

a	b	c	h	m	λ
0.46m	1.15m	0.11m	0.48m	27kg	1.31°

The balance controller has also been verified by simulation based on MBC. Based on the dynamic model developed in [21], the transfer function of the bicycle is:

$$G(s) = \frac{\varphi(s)}{\delta(s)} = \frac{ahv \cdot sin(\lambda)s - mcag \cdot sin^2(\lambda)}{b(h^2s^2 - gh)}, \qquad (1)$$

where φ(s) is the roll angle of the bicycle and δ (s) is the steering angle. In equation (1), m is the mass of the bicycle, a is the horizontal distance between the center of gravity and the contact point of the rear wheel and ground, b is the horizontal distance

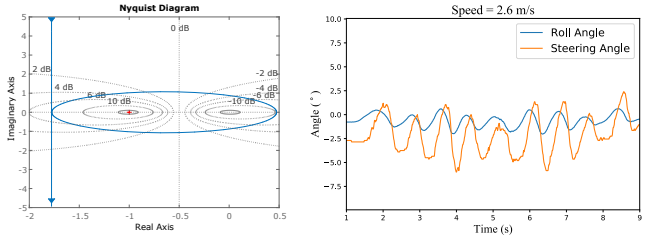
between the contact points of front and rear wheel and ground, c is the trail, h is the height of the center of gravity, λ is the fork angle and v is the velocity of the bicycle. Figure 8 is the bicycle structure with corresponding parameters labeled. The value of those measured parameters are summarized in Table 2. Based on this model, simulations have been conducted to verify the effectiveness of our approach. Nyquist plots for selected speeds have been presented in Figures 10(a)-12(a) and we can see that there are two counter-clockwise encirclements of -1. As a result, there are no right-hand closed-loop poles considering that the open-loop transfer function shown in (1) has two unstable poles. Therefore, the closed-loop system is stable.

TABLE 3: Summary of velocities and corresponding control parameters.

Speed(m/s)	Kp	Ki	Kd
2.0	5.00	0.01	0.35
2.6	3.59	0.01	0.23
3.0	3.22	0.01	0.23
3.6	2.77	0.01	0.23
4.0	2.28	0.01	0.23
4.6	2.25	0.01	0.23
5.2	2.23	0.01	0.208

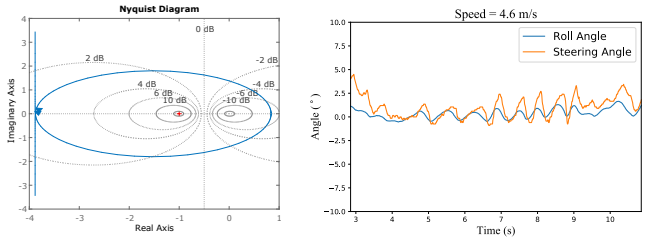
The self-balancing bike prototype was built in order to verify the effectiveness of the hardware development and steering controller. The experiment was performed at seven different constant forward speeds, ranging from 2 m/s to 5 m/s. Table 3 is the summary of the velocities and corresponding control parameters. The bicycle can run stably under those control parameters with a constant forward speed. Figures 10, 11, and 12 show the Nyquist diagrams and the actual performance of the balance controller at selected bicycle speeds, and Figure 9 is a picture showing how actual tests were conducted. The performance is reduced at lower velocities, resulting in the bicycle eventually falling to the side. At velocities above 3.0 m/s the bicycle was able to maintain balance for the entire length of the test track. A video detailing the bicycle balance tests at different speeds is available at `https://youtu.be/Dk2YER1uE_k`.

FIGURE 9: The bike balancing itself with a 4.6 m/s forward velocity in a test. Modified training wheels are not contacting the ground.



(a) Nyquist plot (b) Steering and roll angles

FIGURE 10: Close-loop analysis and experimental results at 2.6 m/s.

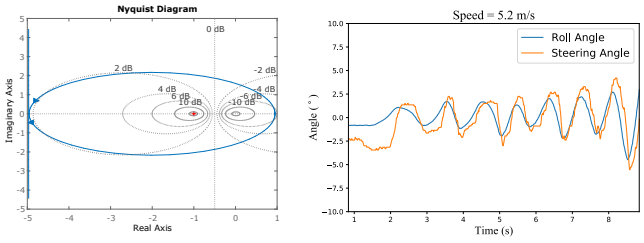


(a) Nyquist plot (b) Steering and roll angles

FIGURE 11: Close-loop analysis and experimental results at 4.6 m/s.

6 CONCLUSIONS AND FUTURE WORK

In this paper, a design for a self balancing bicycle was presented and discussed. The effectiveness of a data-driven based controller (PID controller) has been verified by both simulation and implementation on a hardware prototype. The controller can balance the prototype bicycle at various constant forward speeds with a reliable performance. The primary goals of this study were to design and implement the hardware for a self balancing bicycle and to build a research platform for the further study of automated driving of bicycles, balance algorithms, and human bicycle interaction. Both hardware and control algorithms for



(a) Nyquist plot (b) Steering and roll angles

FIGURE 12: Close-loop analysis and experimental results at 5.2 m/s.

self-driving bicycle have been developed, and a variety of sensors were applied on the system to achieve environmental awareness.

For future work, this system will be further developed to incorporate the CMG for low forward speed balance control to improve self-balancing actuation. In addition, the sensors implemented on the bike will be further utilized to gain a more complete environmental awareness, allowing for increased autonomy. Beyond improving the self-balancing actuation of the system and its environmental awareness, the bicycle will be used as a platform for other forms of research. Some of these research topics could include environmental modeling of areas unreachable by cars, bike sharing applications, rehabilitation applications, bicycle-car interaction, bicycle-UAV interaction, and research on bicycle-human interaction. This study has large potential to generate data which can be used in both self driving algorithms and human robot-interactions.

REFERENCES

[1] Bimbraw, K., 2015. "Autonomous cars: Past, present and future a review of the developments in the last century, the present scenario and the expected future of autonomous vehicle technology". In Informatics in Control, Automation and Robotics (ICINCO), 2015 12th International Conference on, Vol. 1, IEEE, pp. 191–198.

[2] Fuller, D., Gauvin, L., Kestens, Y., Daniel, M., Fournier, M., Morency, P., and Drouin, L., 2011. "Use of a new public bicycle share program in montreal, canada". *American journal of preventive medicine,* ***41***(1), pp. 80–83.

[3] Zhang, Y., Chen, K., Yi, J., and Liu, L., 2014. "Pose estimation in physical human-machine interactions with application to bicycle riding". In Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on, IEEE, pp. 3333–3338.

[4] Goodrich, M. A., and Schultz, A. C., 2007. "Human-robot interaction: a survey". *Foundations and trends in human-computer interaction,* ***1***(3), pp. 203–275.

[5] Meyer, D., Zhang, W., and Tomizuka, M., 2015. "Sliding mode control for heart rate regulation of electric bicycle riders". In ASME 2015 Dynamic Systems and Control Conference, pp. V002T27A003–V002T27A003.

[6] Rani, P., Liu, C., Sarkar, N., and Vanman, E., 2006. "An empirical study of machine learning techniques for affect recognition in human–robot interaction". *Pattern Analysis and Applications,* ***9***(1), pp. 58–69.

[7] Jeon, W., and Rajamani, R., 2017. "Two-dimensional active sensing system for bicyclist-motorist crash prediction". In American Control Conference (ACC), 2017, IEEE, pp. 2315–2320.

[8] Kooijman, J., Meijaard, J. P., Papadopoulos, J. M., Ruina, A., and Schwab, A., 2011. "A bicycle can be self-stable without gyroscopic or caster effects". *Science,* ***332***(6027), pp. 339–342.

[9] Roser, X., and Sghedoni, M., 1997. "Control moment gyroscopes (cmg's) and their application in future scientific missions". In Spacecraft Guidance, Navigation and Control Systems, Vol. 381, p. 523.

[10] Yetkin, H., Kalouche, S., Vernier, M., Colvin, G., Redmill, K., and Ozguner, U., 2014. "Gyroscopic stabilization of an unmanned bicycle". In American Control Conference (ACC), 2014, IEEE, pp. 4549–4554.

[11] Beznos, A., Formal'Sky, A., Gurfinkel, E., Jicharev, D., Lensky, A., Savitsky, K., and Tchesalin, L., 1998. "Control of autonomous motion of two-wheel bicycle with gyroscopic stabilisation". In Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on, Vol. 3, IEEE, pp. 2670–2675.

[12] Lee, S., and Ham, W., 2002. "Self stabilizing strategy in tracking control of unmanned electric bicycle with mass balance". In Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on, Vol. 3, IEEE, pp. 2200–2205.

[13] Yamakita, M., and Utano, A., 2005. "Automatic control of bicycles with a balancer". In Advanced Intelligent Mechatronics. Proceedings, 2005 IEEE/ASME International Conference on, IEEE, pp. 1245–1250.

[14] Getz, N. H., and Marsden, J. E., 1995. "Control for an autonomous bicycle". In Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on, Vol. 2, IEEE, pp. 1397–1402.

[15] Tanaka, Y., and Murakami, T., 2004. "Self sustaining bicycle robot with steering controller". In Advanced Motion Control, 2004. AMC'04. The 8th IEEE International Workshop on, IEEE, pp. 193–197.

[16] Fan, X., and Tomizuka, M., 2010. "Robust disturbance observer design for a power-assist electric bicycle". In American Control Conference (ACC), 2010, IEEE, pp. 1166–1171.

[17] Song, D., Lee, H. N., Yi, J., and Levandowski, A., 2007. "Vision-based motion planning for an autonomous motorcycle on ill-structured roads". *Autonomous Robots,* ***23***(3), pp. 197–212.

[18] Cho, H., Seo, Y.-W., Kumar, B. V., and Rajkumar, R. R., 2014. "A multi-sensor fusion system for moving object detection and tracking in urban driving environments". In Robotics and Automation (ICRA), 2014 IEEE International Conference on, IEEE, pp. 1836–1843.

[19] Meijaard, J. P., Papadopoulos, J. M., Ruina, A., and Schwab, A. L., 2007. "Linearized dynamics equations for the balance and steer of a bicycle: a benchmark and review". In Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, Vol. 463, The Royal Society, pp. 1955–1982.

[20] Euston, M., Coote, P., Mahony, R., Kim, J., and Hamel, T., 2008. "A complementary filter for attitude estimation of a fixed-wing uav". In Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on, IEEE, pp. 340–345.

[21] He, J., Zhao, M., and Stasinopoulos, S., 2015. "Constant-velocity steering control design for unmanned bicycles". In Robotics and Biomimetics (ROBIO), 2015 IEEE International Conference on, IEEE, pp. 428–433.