PSU–Journal of Engineering, Technology, and Computing Sciences (JETCS)
(Vol. No. 2, Issue 2 pp. 21-26, December 2020)

Autonomous Self-Balancing Bicycle with Line Follower Prototype

Engr. Mark Jecel J. Rapir
School of Engineering, Colegio De Dagupan
mjrapir@cdd.edu.ph

Abstract *– This paper aims to design and build a bicycle prototype capable of driving, balancing, and following a particular path. The robot detects tilt angle and correctly reacts to maintain a steady vertical position to support balancing. Input data from the sensors trigger a control system that outputs a balancing torque to a motor spinning the reaction wheel. The requirements include that the bicycle should accelerate, driving in a straight line without falling.*

Keywords *- Reaction wheel, Gyroscopic effect, Balancing control, Kalman's Filter, PID Control*

INTRODUCTION

Learning how to ride a bicycle for the first time is one of the typical childhood achievements that most of us are fond of reminiscing. However, for those people, children and adults alike, who never had the opportunity to learn how to ride a bicycle because of being physically or mentally challenged, riding a bike is still a skill they yearn to possess. A system that can assist them without compromising the experience of riding a bicycle could be an empowering tool for these individuals.

The act of balancing a bicycle is one of the phenomena that baffles many scientists ever since. It is so mysterious that only a few had a concrete explanation of how humans balance a bicycle. The closest theory, which the researchers had considered, is that of an inverted pendulum. A pendulum is a mass suspended at the one end of a string or rod whose other end served as the fixed pivot. When the mass is below the pivot, the pendulum is pretty stable. However, if the mass is above the pivot, the inverted pendulum becomes very unstable, and it was almost impossible to keep the mass still above the other end of the string or rod.

There have been several known methods for robot bicycles. Beznoz et al. (1998) and Gallaspy (1999) used gyroscopic stabilization. Lee and Ham (2002) moved the Center of Gravity (COG), also called the mass balancing method. Tanaka and Murakami (2004) utilized steering control.

Murata (2005) developed the famous self-balancing robot bicycle called Murata Boy, which uses the robot's reaction wheel to serve as a torque generator, which served as an actuator to balance the bike. The spinning rotor in the reaction wheel usually has a spin rate of zero. Its spin is fixed, and its speed is varied to create reaction torque around the spin axis. Relative to other momentum-exchange actuators, reaction wheels are the simplest and least expensive. However, it is less energy-efficient, and it can only produce a relatively low amount of torque. Gallaspy also proposed another method that controls the torque exerted on the steel handlebar to balance the bicycle. It requires low mass and is energy efficient; however, it also involves ground reaction force and lacks robustness versus significant roll disturbances.

Prasad and Nirwan (2016) utilized the flywheel design that implements a flywheel rotating about an axis parallel to the bicycle's frame. The method treats the bike as an inverted pendulum where the contact point between the bicycle wheel and the floor served as the pivot. Relatively high stability and less complex mathematical model are the significant advantages of this design, making it more straightforward to implement. However, it does not allow easy steering, and some features of the frame need redesigning to distinguish it from a usual bicycle.

This study aims to build a Self-balancing bicycle with a line-following feature using different balancing techniques. The researchers want to determine the initial requirement set in implementing the Self-Balancing Bicycle with a line-following feature. Moreover, the researchers want to evaluate what features a Self-Balancing Bicycle should have to maintain balance and assess why a self-balancing bicycle needs a feedback controller using a steering wheel, momentum wheel, and sensors.

MATERIALS AND METHODS

Conceptual Framework

The central processor is composed of Arduino Mega 2560 as the microcontroller where the Wi-Fi module is connected to control it wirelessly via phone using an application made. It also contains the two smart serial servos for the back wheel and steering wheel, Dynamixel, with a fully integrated DC motor, reduction gearhead, controller, driver, network in one module Arduino Nano controls the momentum wheel and sensors. For the momentum wheel, an encoder motor is an electromechanical device that provides an electrical signal for speed and position control. It is connected with a motor driver that acts as a current amplifier. On the other hand, the sensors used are the IMU-9D0F module, which has 3- axis gyroscope, triaxial accelerometer, and

ISSN 2599-462X (Print) www.psurj.org/jetc ISSN 2599-4638 (Online)

3-axis magnetic field; and Infrared sensors for line following function by detecting reflected light coming from its infrared LED.

Control View

The control of the bicycle is simple and straightforward. It includes reading data from the sensors, filtering the data, applying control algorithms on the data, and sending control signals to the actuators. A microcontroller does all the filtering of data and the generation of a control signal.

The line follower system uses a flow chart defining how it follows its designated line in its path. This flow chart contains the method, how the robot is making decisions on its pathway. Line follower bicycle is mainly dependent on the sensor system, and its process is slow. The bike reaches faster towards its destination so that it becomes speedier and more effective for its work.

Balancing Technique

Various researchers have several bicycle stability approaches, including physical system designs, plant models, and underlying control strategies. Steering control, gyros, a moving mass, and a combination of these are the most commonly used techniques to achieve stability.

This study uses a reaction wheel and steering control. The reaction wheel design employs a flywheel that rotates about an axis parallel to the bicycle's frame. This design converts the bicycle into an inverted pendulum with a fixed pivot where the bicycle wheels meet the floor. As the bicycle falls to one side, a motor mounted on the bicycle exerts a torque on the flywheel, generating a reactionary torque on the bicycle, which restores the bicycle's balance. There are several advantages of this design. This design is very stable, giving the bicycle capability to balance even in a stationary position. Due to the simplicity of the autonomous bicycle model's design, the controller's implementation is relatively straightforward. This design also allows the bicycle to travel in a straight line with little or no deviations.

The following equation gives the value of the angle of inclination

$$\theta p(i) = \theta p(i-1) + \frac{1}{6}(vali\text{-}3 + 2vali\text{-}2 + 2vali\text{-}1)$$

Equation 1

PID Control

There are two distinct categories of control techniques in control systems. The first technique, being a linear control model of the system. A linear control method models the process of an operating desire point. The linear method is usually very sufficient in balancing the system and bringing it to a stable vertical position. On the other hand, a nonlinear controller uses the system's unrealistic dynamics model to design a controller. Nonlinear controllers would provide a more robust system implementation. The implementation and complexity difficulty associated with the nonlinear method causes most control researchers to utilize the linear controller approach.

The self-balancing bicycle in this study utilizes a linear controller applied through a Proportional, Integral, and Derivative, also referred to as the PID.

In this algorithm, the error signal received is the input. And the following equation is applied to the error.

$$U(t) = (K_P) \bullet e(t) + K_d\left(\frac{d}{dt}e(t)\right) + K_i \int e(t)$$

Equation 2

Kalmann's Filter

Digital control engineering often uses Kalman's filter. It is an essential addition to the proposed project's stability to provide sensor fusion between accelerometer and gyroscope. The digital filter provides reliable sensor data to get tilt angle information and improves the measured angle's overall accuracy.

A set of equations performs Kalman's Filtering Algorithm.

$$\hat{x}_{k|k-1} = F\hat{x}_{k-1|k-1} + B\dot{\theta}_k$$

$$\begin{bmatrix}\theta \\ \dot{\theta}_b\end{bmatrix}_{k|k-1} = \begin{bmatrix}1 & -\Delta t \\ 0 & 1\end{bmatrix}\begin{bmatrix}\theta \\ \dot{\theta}_b\end{bmatrix}_{k-1|k-1} + \begin{bmatrix}\Delta t \\ 0\end{bmatrix}\dot{\theta}_k$$

$$= \begin{bmatrix}\theta - \dot{\theta}_b\Delta t \\ \dot{\theta}_b\end{bmatrix}_{k-1|k-1} + \begin{bmatrix}\Delta t \\ 0\end{bmatrix}\dot{\theta}_k$$

$$= \begin{bmatrix}\theta - \dot{\theta}_b\Delta t + \dot{\theta}\Delta t \\ \dot{\theta}_b\end{bmatrix}$$

$$= \begin{bmatrix}\theta + \Delta t(\dot{\theta} - \dot{\theta}_b) \\ \dot{\theta}_b\end{bmatrix}$$

$$P_{k|k-1} = FP_{k-1|k-1}F^t + Q_k$$

$$\begin{bmatrix}P_{00} & P_{01} \\ P_{10} & P_{11}\end{bmatrix}_{k|k-1} = \begin{bmatrix}1 & -\Delta t \\ 0 & 1\end{bmatrix}\begin{bmatrix}P_{00} & P_{01} \\ P_{10} & P_{11}\end{bmatrix}_{k-1|k-1}\begin{bmatrix}1 & 0 \\ -\Delta t & 1\end{bmatrix} + \begin{bmatrix}Q_\theta & 0 \\ 0 & Q_{\dot{\theta}_b}\end{bmatrix}\Delta t$$

$$= \begin{bmatrix}P_{00} - \Delta t P_{10} & P_{01} - \Delta t P_{11} \\ P_{10} & P_{11}\end{bmatrix}_{k-1|k-1}\begin{bmatrix}1 & 0 \\ -\Delta t & 1\end{bmatrix} + \begin{bmatrix}Q_\theta & 0 \\ 0 & Q_{\dot{\theta}_b}\end{bmatrix}\Delta t$$

$$= \begin{bmatrix}P_{00} - \Delta t P_{10} - \Delta t(P_{01} - \Delta t P_{11}) & P_{01} - \Delta t P_{11} \\ P_{10} - \Delta t P_{11} & P_{11}\end{bmatrix}_{k-1|k-1} + \begin{bmatrix}Q_\theta & 0 \\ 0 & Q_{\dot{\theta}_b}\end{bmatrix}\Delta t$$

$$= \begin{bmatrix}P_{00} - \Delta t P_{10} - \Delta t(P_{01} - \Delta t P_{11}) + Q_\theta \Delta t & P_{01} - \Delta t P_{11} \\ P_{10} - \Delta t P_{11} & P_{11} + Q_{\dot{\theta}_b}\Delta t\end{bmatrix}$$

$$= \begin{bmatrix}P_{00} + \Delta t(\Delta t P_{11} - P_{01} - P_{10} + Q_\theta) & P_{01} - \Delta t P_{11} \\ P_{10} - \Delta t P_{11} & P_{11} + Q_{\dot{\theta}_b}\Delta t\end{bmatrix}$$

Line Detection

The path to be detected is predefined and visible as a black line on a white surface with a high contrasting color, or the path can be complex such as magnetic markers or laser guide markers. Various sensors can be employed to detect the line which the bicycle has to follow.

This study uses an IR sensor. If the left sensor detects a black line only, the bicycle turns left for the right motor to work. When the left motor stops and the right motor rotates in a clockwise direction, the bicycle turns left.

Line following is an intelligent communication and which the bicycle detects a visual line embedded on the floor and follows it. Moreover, the microcontroller is the

ISSN 2599-462X (Print) www.psurj.org/jetc ISSN 2599-4638 (Online)

controlling unit that takes the sensor's input, matches the exact action needed, and then sends the output to the actuator.

Software

Matlab. It is a high-performance language mainly used for technical computing. It incorporates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notations. In this study, a simulation is carried out by solving the equations, simulating the implemented design and calculations to test them in theory, and testing if components available in the market would have adequate performance using this program.

Arduino IDE. It is a user-friendly cross-platform application written in the Java programming language. IDE stands for Integrated Development Environment; writing and uploading programs to Arduino and other development boards are done in Arduino IDE. The boards Arduino mega 2560 and Arduino nano are used throughout the study, acting as microcontrollers and uploading the codes.

App Inventor. This program lets you develop Android phone applications using a web browser and either a connected phone emulator. Data is stored in the AppInventor, where you can regularly track the progress of the project.

An android application is made through this program to communicate with a Self-balancing bike with a Line-Following feature via Wi-Fi.

Auto-CAD. It is a Computer-Aided Design tool that allows many different designers to create adverse kinds of drawings and designs. This program helps designers create their designs much more quickly than by hand and offers many quick, easy, and valuable features. It is used for designing the bicycle frame and structure of the prototype.

Results and Discussion

Prototype

The motorcycle project is composed of a two-wheeled robot that can balance and move using a rotating disc, which was called the inertia wheel, to compensate if the motorcycle loses balance. The motorcycle was controlled by an Arduino MKR1000, the Arduino MKR Motor Carrier, a DC motor to move the back wheel, an encoder, a DC motor to control the inertia wheel, a 6 axis IMU, a standard servo motor to steer the motorcycle's handle, a distance sensor, and a tachometer. In this project, the simulation of the vehicle's overall behavior and the created models for programming the hardware components and improve the control algorithms' quality, where it is programmed with Simulink, controls its balance algorithm and makes it move in a straight line were involved.

Modeling and Program Simulation

A simplified version of the system derivation was examined below to understand the motorcycle-inertia wheel system's equations of motion and create a functioning simulation that was not too complex.

The following were the assumptions made out of this system:

- The Motorcycle was free to move only about the wheel-ground axis.
- There was no rotational friction between the motorcycle and the ground or between the motorcycle and the inertia wheel.
- The motorcycle wheels' thickness was negligible.
- Air resistance was negligible.

The physics of the project is easily understood by looking at the motorcycle from behind. The most critical coordinate is the lean angle, θ, which is equal to 0 degrees when the motorcycle is perfectly upright, positive when the motorcycle leans counterclockwise when viewed from behind, and negative when the motorcycle leans clockwise. Another critical angle is the rotational displacement of the inertia wheel, φ. As the inertia wheel rotates relative to the motorcycle's rest, φ varies in value, where positive displacement equals counterclockwise.

Finding the center of mass of the entire motorcycle system (including the inertia wheel) was complicated due to some fixed factors. But this location can be measured experimentally, but it is approximately just below the inertia wheel. The center of mass location is essential to motorcycle dynamics because the gravitational force acts directly through this point. The height of the center of mass from the ground when the motorcycle is upright ($\theta = 0$) is hcm. The motorcycle system is free to rotate around the wheel-ground axis, an imaginary line connecting the rear and front wheels to the ground (into the picture). The inertia wheel can rotate along its axis, directly through the middle of the inertia wheel.

A torque occurs when a force reacts on a system at some point in the rotational axis. The torque's magnitude is given by using the following equation:

$$\tau=|F\cdot r\cdot\sin(\alpha)|\tau=|F\cdot r\cdot\sin(\alpha)|$$

Equation 3

The motorcycle system includes the rear and front wheels, the steering column, the motorcycle body (along with all its embedded electronics), and the inertia wheel. The axis of rotation for this system is the wheel-ground axis, and the measure of rotation is the lean angle θ, defined earlier. The torque acting on the motorcycle (denoted with M) about the wheel-ground axis has 3

ISSN 2599-462X (Print) www.psurj.org/jetc ISSN 2599-4638 (Online)

major components:

- Gravitational torque (τg,Mτg,M)
- Inertia wheel torque (τIW, MτIW, M)
- External torque (τext, Mτext, M)

The gravitational torque is due to the gravitational force pulling down on the motorcycle system center's center when it is not directly above the wheel-ground axis (i.e., when it is leaning). Based on Equation 3, the gravitational torque is:

$$\tau g,M=MM\cdot g\cdot hcm\cdot \sin(\theta)\tau g,M=MM\cdot g\cdot hcm\cdot \sin(\theta)$$

Equation 4

The motorcycle system uses other torques to prevent it from falling. When the inertia wheel motor actuates, the motor shaft exerts a torque on the inertia wheel, causing it to accelerate (rotationally). In turn, the inertia wheel exerts an equal and opposite torque on the motor shaft due to angular momentum conservation.

The motion of the inertia wheel is defined by:

$$IW\cdot \phi\ddot{}=\tau motor,IWIIW\cdot \phi\ddot{}=\tau motor,IW$$

Equation 5

Testing of Physical Mode in Simulink

This study uses a Simulink model that implements the motorcycle's physical behavior. This Simulink model aids in the implementation of the control mechanism to balance the motorcycle vertically. This model contains a Motorcycle subsystem block, an External Torque block to generate a data array, a Gain block that is a signal multiplier, and a Constant block to provide data input. The first test is torque-free, i.e., no torque is applied to the inertia wheel.

Notice the subsystem's inputs are an external torque from the environment and a torque applied to the inertia wheel motor's motorcycle. The model inside the subsystem generates the physical quantities of interest: the lean angle θ, its time derivative θ', and the time derivative of the inertia wheel angle, ϕ'.

Using the Simulation Data Inspector (SDI) environment, logged data were viewed from simulations and examined in any visual axes, and compared the same signals across multiple simulations. The observations on the motorcycle system's behavior are recorded, and the control algorithm is fine-tuned to optimize performance.

The SDI window shows the motorcycle blocks' view (three outputs: theta, theta dot, and inertia). Notice that the signals present an oscillatory behavior when the torque is absent.

For the second test, the lean angle continues to oscillate after adding a component that made it decrease at a constant speed. Practically, this means that the motorcycle tends to fall, but since there is no "ground" such in the model, the motorcycle starts to rotate around its horizontal axis.

The inertia wheel's average speed increases indefinitely, which is practically impossible since the motorcycle breaks at some point: either the motor stops its acceleration because of mechanical conditions, or the motor driver breaks because of over-current. While neither of these situations is desirable, it is uncertain which happens first (i.e., mechanical or electrical damage). Therefore, a controller design is needed to avoid these circumstances.

Troubleshooting

This study uses the inertial measurement unit (IMU) to measure the lean angle θ and its time derivative θ'. It is possible to translate the information from a datasheet into a Simulink block to use in the programs. These models are used to test the IMU, understand its outputs, and configure it for the motorcycle balancing controller.

According to the configuration model, the block outputs its absolute orientation relative to the magnetic field and gravitational vector and the angular rate along with the local coordinates of the IMU sensor.

Preparation and Balancing

For the IMU to give accurate outputs, each sensor is initialized to locate the gravity vector and magnetic field and offset the coordinates accordingly. The elements of the calibration status vector respectively represent:

1. IMU system
2. Gyroscope
3. Accelerometer
4. Magnetometer

Each calibration status vector element has a value of 0 to 3, indicating the sensor's calibration degree. For the motorcycle, the gyroscope and the magnetometer need to be fully calibrated (3). To start the gyroscope calibration, leave the motorcycle entirely at rest while the model runs. Then, picking up the motorcycle and rotate it at least 90 degrees along each of the 3 spatial axes to calibrate its magnetometer.

The calibration process must be done every time the BNO055 IMU is powered on, which means that the model's strategies must figure out that it includes calibration when the motorcycle is power up.

In-Balance

The scope window displays a graph of theta, theta dot, torque command, and PD terms upon running the design system block and calibrating the IMU system, gyroscope, magnetometer, and accelerometer. In setting the center of balance, the motorcycle is in a vertical position while observing the Scope window. The motorcycle was pushed back and forth a few degrees in each direction to move freely across the balance point.

ISSN 2599-462X (Print) www.psurj.org/jetc ISSN 2599-4638 (Online)

The amplitude of the motion was gradually decreased until the balance point was identified within a few degrees.

There were five (5) main blocks called subsystems in the system in which Simulink performs computations. The five (5) blocks are IMU, Controller, Inertia wheel, rear wheel, and steer servo.

IMU is the source of all data regarding the angle position of the motorcycle. On the other hand, the rear wheel controls the motor at the back. Steer servo for the steering control, and inertia wheel for the reading of RPM or speed of the inertia wheel and Controller that serves as the central controller of the system where all the data gathered from all subsystems were analyzed to make the motorcycle functioned well.

The rest on the diagram were the battery reader, sliders, and measurement displays in diagnosing the output error per system.

The graph in the window was unstable without any calibration compared when there is calibration. And tuning the control gains was needed to find the optimal control law for balancing. Simulink Control Design was used to determine the optimal control gains by observing the state errors over time and characterizing the system's behavior under control. For the controller's initial tuning, it was helpful to attempt the tuning manually to get an intuition for the system behavior and each control gain's effect. In most PID tuning methodologies, the I and D gains are considered proportional to the P gain, and one should not adjust P without making equivalent proportional changes to I and D simultaneously. Otherwise, it would be difficult to assess the effect of a change in P alone. the motorcycle.

CONCLUSION AND RECOMMENDATIONS

The bicycle can balance itself for two minutes; beyond this, overheating occurs. Wireless control can be an option however imposes possible propagation delays. The bicycle did not achieve the line tracing feature because it needs perfect balancing stability before proceeding in this feature which involves the steering control. There is also an observed limitation on its balancing while in a straight motion.

The project demonstrated that a fully functional self-balancing bicycle with a line following feature is not achieved. Because of low-cost components, time frames, and other factors, the project is an ideal demonstration of control theory for a classroom setting. The project can also serve as a test platform to write the necessary software control algorithms to implement commercial products.

REFERENCES

[1] Abhishek Chougule, Rohit Lade, Alok Bendale, Amol Gade, & Prof. M.B.Sorte (2018). Design, fabrication, and analysis of self-balancing electric bike. International Journal of Scientific & Engineering Research, 9(5). Retrieved from https://www.ijser.org/researchpaper/Design-and-Fabrication-of-Self-balancing-Electric-Bike.pdf

[2] Aphiratsakun, Narong & Techakittiroj, Kittiphan. (2013). Autonomous AU Bicycle: Self- Balancing and tracking control (AUSB²). Retrieved from https://www.researchgate.net/publication/286705989_Autonomous_AU_Bicycle_Self-Balancing_and_tracking_control_AUSB2

[3] Chi Ooi, Rich (2003). Balancing a Two-Wheeled Autonomous Robot (Bachelors Degree Thesis, University of Western Australia). Retrieved from https://robotics.ee.uwa.edu.au/theses/2003-Balance-Ooi.pdf

[4] Grönlund, A., & Tolis, C. (2018). Riderless self-balancing bicycle: Derivation and implementation of a time variant linearized state space model for balancing a bicycle in motion by turning the front wheel. Retrieved from http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1237256&dswid=4072

[5] He, Jiarui & Zhao, Mingguo. (2015). Control System Design of Self-balanced Bicycles by Control Moment Gyroscope. Lecture Notes in Electrical Engineering. 338. 205-214. 10.1007/978-3-662-46466-3_21.

[6] Huang, Yonghua, Liao, Qizheng, Guo, Lei, & Wei, Shimin (2013). Balanced motions realization for a MECHANICAL Regulators Free and Front-wheel Drive Bicycle Robot under Zero Forward Speed. International Journal of Advanced Robotic Systems, 10(8). Retrieved from https://journals.sagepub.com/doi/full/10.5772/56701

[7] Prasad, Mukeshkumar & Nirwan, Nilesh (2016). Design and Fabrication of Automatic Balancing Bicycle. International Journal of Science, Engineering, and Technology Reasearch, 5(12). Retrieved from http://ijsetr.org/wp-content/uploads/2016/02/IJSETR-VOL-5-ISSUE-2-532-536.pdf

[8] Ravi teja, T., & Krishna Chaitanya S. (2012). Computation of Natural Frequencies of Multi Degree of Freedom System. International Journal of Engineering Research & Technology, 1(10). Retrieved from https://www.ijert.org/research/computation-of-natural-frequencies-of-multi-degree-of-freedom-system-IJERTV1IS10544.pdf

[9] Suahil, Muhammad & Ishaqui, Ammar (2013). To Use Sensory Information as Feedback for High-Performance Real-Time Robotics Applications.

American Journal of Engineering and Technology Research 13(1). Retrieved from https://studylib.net/doc/18497556/volume-13--no.-1--2013-print-version

[10] Sung, Hsin-Chuan (2005). Balancing Robot Control And Implementation (Master's Thesis, Texas A&M University). Retrieved from http://oaktrust.library.tamu.edu/bitstream/handle/1969.1/155306/SUNG-THESIS 2015.pdf?sequence=1

ISSN 2599-462X (Print) www.psurj.org/jetc ISSN 2599-4638 (Online)