

PAPER • OPEN ACCESS

A Design for Self Balancing Scale Model Bicycle

To cite this article: Umar Tsani Abdurrahman *et al* 2021 *J. Phys.: Conf. Ser.* **1764** 012171

View the [article online](#) for updates and enhancements.

You may also like

- [Detecting fine scratches on smooth surfaces with multiscale wavelet representation](#)
Li Yao, Yan Wan, Ming Yao et al.
- [Ultrasonic Sensor Application As A Performance Enhancement Of Robot Two Wheels](#)
Yeni Irdyanti, Rd Kusumanto, Masayu Anisah et al.
- [Justification of Filter Selection for Robot Balancing in Conditions of Limited Computational Resources](#)
M V Momot, E V Politsinskaia, A V Sushko et al.



240th ECS Meeting

Digital Meeting, Oct 10-14, 2021

We are going fully digital!

Attendees register for free!

REGISTER NOW



A Design for Self Balancing Scale Model Bicycle

Umar Tsani Abdurrahman^{1*}, Pria Sukamto¹, Mohamad Anas Sobarnas¹ and Mujiarto²

¹ Department Informatics Engineering, Sekolah Tinggi Teknologi Muhammadiyah Cileungsi, Cileungsi, Bogor, Indonesia 16820

² Department of Mechanical Engineering, Universitas Muhammadiyah Tasikmalaya, Tasikmalaya, Indonesia

*umar.tsani@sttmcileungsi.ac.id

Abstract. This research will describe the development of a self-balancing scale model bicycle. The main reason for just using a scale model instead of a full-size bicycle is to reduce development costs while maintaining the same design principle and software coding. For the hardware we are using a 1:10 metal bicycle model, Arduino Uno board for real-time balance control, GY-521 & MPU 6050 for real-time gyro and 3 axis accelerometers, and servo motor for weight balancing movement. 5-volt Li-ion battery power bank used for a power source. For coding, we use Arduino IDE for windows.

1. Introduction

Bicycle is an old invention, more than 125 years ago. But the whole balancing process of a bicycle was not completely known until recent years. Indeed the mechanism is complicated and includes balancing [1] action from the driver himself along with a couple of other forces acting from the bicycle.

To make modeling and abstraction simple we will reduce the problem onto balancing bicycles based on the inverting pendulum model. This model used with the assumption that bicycles in a rest condition. This is the same condition as an inverting pendulum where the centers of mass is high and cause an instability equilibrium. In the inverting Pendulum, this instability can overcome by moving the lowest part of the pendulum so that the entire mass changes its central mass in dynamic equilibrium.

The self-balancing apparatus today can be made possible by the availability of 3 axis accelerometer sensor chips based on MEMS (Micro Electro Mechanical Systems) [2]. The availability of MEMS sensors improve the quality of both and price / performance, due to Moore's law of microelectronics. The calculations required for this system are relatively low and available. The DC motor implemented in this model can saved from toys. Only power electronics and the pendulum has to be custom-made.

2. Literature Review

One of the successful implementations for a model bike balancing is done by Japanese company Murata named Murata Boy figure 1, a robotic bicycle [3]. With a gyro sensor and a large disc to correct any slant.





Figure 1. Murata Boy, bike and robot as one system [4]

It consists of an accelerometer/gyro sensor under the robot and a reaction wheel pendulum in the middle of the robot (figure 2.).

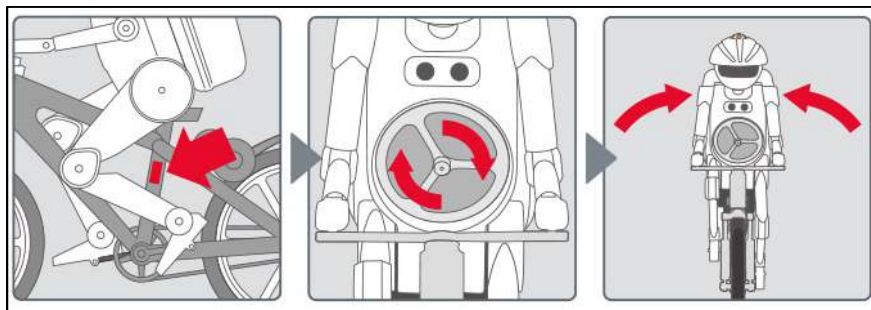


Figure 2. Sensor (left) and reaction wheel pendulum to balance the bike [5]

3. Modeling

Although bicycle stability is complicated, for zero velocity [6] this stability can be achieved by only use an inverting pendulum principle as a figure.3, we only use one axis x here for consideration.

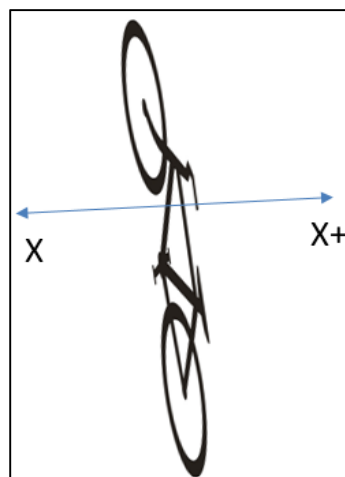


Figure 3. Simplified zero speed bike balancing model

Since the most unstable condition of a bike is at velocity 0, this would tackle the biggest and foremost factor of bike balancing problem, and the dynamic can be simplified as the figure 4.

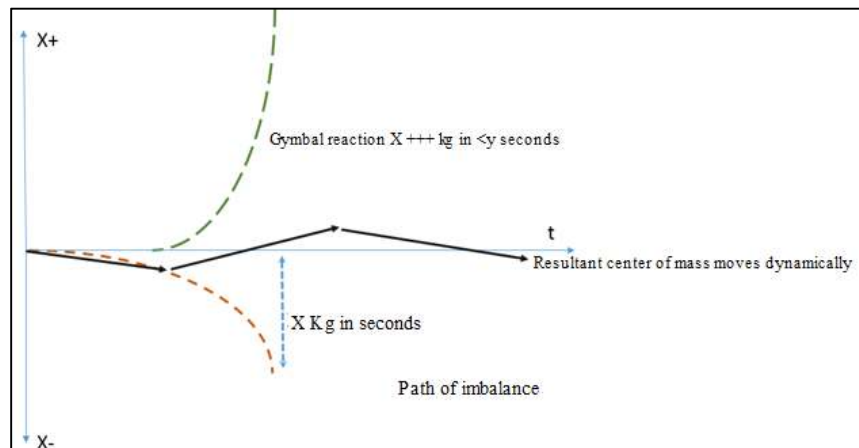
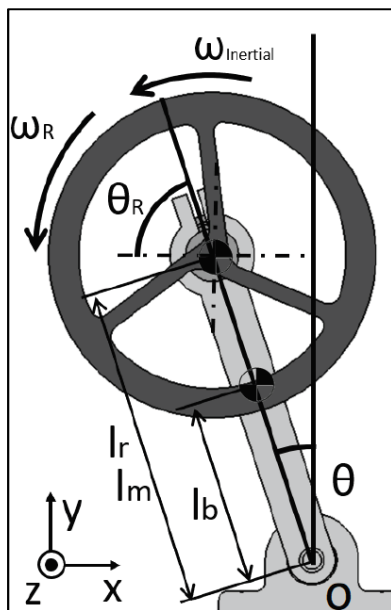


Figure 4. Dynamics of x-axis balancing stability process [7]

Dynamic balancing can be provided through actions on the reaction wheel. The reaction wheel [8] itself is modeled over an inverted pendulum [9] where it has no inherent stability, this stability is provided via a dynamic process of adjusting reaction wheel inertia as in Figure 5 [8].



M_m : Motor Mass
 M_R : Reaction Wheel Mass
 M_b : Pendulum Arm Mass
 b_b : Pendulum Arm axis viscous friction
 l_b : Pendulum Arm center of mass distance to the axis
 l_m : Motor center of mass distance to the axis
 l_r : Reaction wheel center of mass distance to the axis
 g : Gravitational acceleration.
 τ_c : Effective Motor Control torque
 Inertia:
 I_{bo} : Pendulum concerning point o.
 I_{ro} : Reaction concerning respect to point o.
 I_{mo} : Motor with respecting to point o.
 I_R : Reaction wheel concerning its center of mass.
 Angular Speed:
 ω_R : Wheel speed concerning the pendulum arm
 $\omega_{Inertial}$: Wheel Inertial speed.

Figure 5. Reaction wheel model.

One factor that we need to consider is how high is the position of the axis of the reaction wheel from the base (l_m), this actually can have a direct effect on the maximum recovery angle (θ) as can be seen in figure 6.

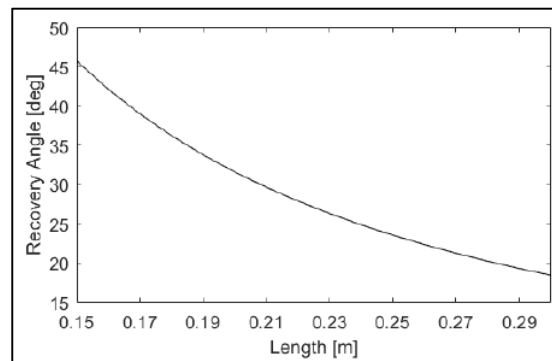


Figure 6. Fix moment of inertia effect to recovery angle vs length of the inverted pendulum

4. Discussion

The parameters are as below:

- M bike = 400 gr M motor = 110 gr - M pendulum = 95gr M electronics = 50gr
- M batt = 100gr so the total: M Total =705gr

The motor torque and specs are as below:

- Standard 130 Type DC Motor - Operating Voltage: 4.5 to 9 V.
- Recommended or Rated Voltage: 6V
- Current at No Load: 70 mA (max) - No-Load Speed: 900 rpm.
- Loaded current: 250 mA (approx) - Rated Load: 10g*cm
- Motor Size: 27.5 mm x 20 mm x 15 mm

The electronics is using Arduino Uno, IBT2 H Bridge, MPY 6050 Sensors with the diagram as below in figure 7.

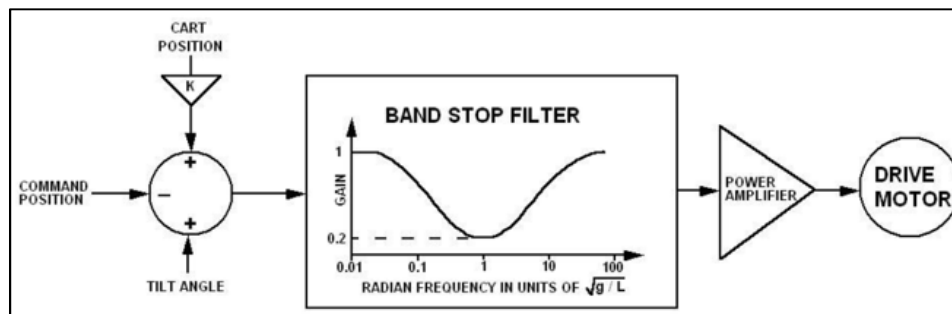
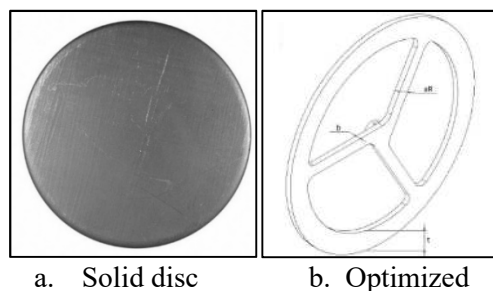


Figure 7. Electronic flow from sensing position to controlling the motor [10]

The reaction wheel pendulum has a diameter of 5 cm, was a solid disc at first (figure 8a), but according to the following equation, the ring is doubling the inertia at the same mass:

So the spoke which connects axis to the outer ring has to be made as thinly as possible just enough to make the wheel rigid as in figure 8b. The radius on the equation is the average radius /center mass of the ring.



a. Solid disc

b. Optimized

Figure 8. Reaction wheel solid vs optimized (minimum mass, max inertia) [11].

The H-bridge to drive the motor is L298N or MP6513 as a diagram in figure 9.

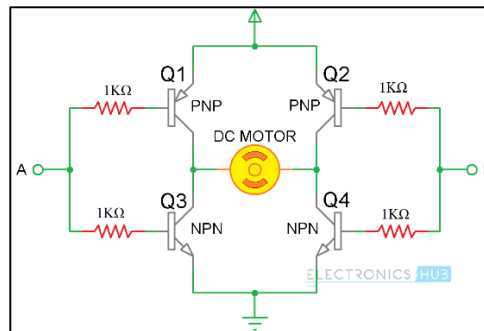


Figure 9. H Bridge to drive the DC motor [12]

The code for constant monitoring of the sensor and adjusting the motor is provided at a high level as in Figure 10.

```

bicycle_balancing_01 | Arduino 1.8.7
File Edit Sketch Tools Help

bicycle_balancing_01.h

// Algorithm: sense accelerometer, send pulse to motor DC/stepper RC or LC
#define STEPPER_PIN_1 9
#define STEPPER_PIN_2 10
#define STEPPER_PIN_3 11
#define STEPPER_PIN_4 12
int step_number = 0;

// (c) UTA 2020
#include "Wire.h" // This library allows you to communicate with I2C devices.
const int MPU_ADDR = 0x68; // I2C address of the MPU-6050. If AD0 pin is set to HIGH, the I2C address will be 0x69.
int16_t accelerometer_x, accelerometer_y, accelerometer_z; // variables for accelerometer raw data
int16_t gyro_x, gyro_y, gyro_z; // variables for gyro raw data
int16_t temperature; // not use
char tmp_str[7]; // temporary variable used in convert function
char* convert_int16_to_str(int16_t i) { // converts int16 to string. Moreover, resulting strings will have the same
    sprintf(tmp_str, "%4d", i);
    return tmp_str;
}

void setup() {
    pinMode(STEPPER_PIN_1, OUTPUT);
    pinMode(STEPPER_PIN_2, OUTPUT);
    pinMode(STEPPER_PIN_3, OUTPUT);
    pinMode(STEPPER_PIN_4, OUTPUT);

    Serial.begin(9600);
    Wire.begin();
    Wire.beginTransmission(MPU_ADDR); // Begins a transmission to the I2C slave (GY-521 board)
    Wire.write(0x6B); // PWR_MGMT_1 register
    Wire.write(0); // set to zero (wakes up the MPU-6050)
    Wire.endTransmission(true);
    readsensordbg();
}

```

Figure 10. The Algorithm and high-level code on Arduino IDE

5. Conclusion

The design has been made for scale model bicycles however this design can be implemented also for a real bike or motorcycle was given a change in the DC motor with sufficient power and torque, sufficient inertia by bigger pendulum wheel and sufficient power from bigger H bridge power electronic. The inverted pendulum is sufficient to provide a self-balancing for scale model bicycle. However for a full-scale implementation a study has to be made if the design to be implemented with a human passenger

to provide sufficient data on how to provide more stabilization from other dynamic balancing factors such as steering wheel, velocity, gyro effect from the wheels, etc.

References

- [1] J. P. Meijaard, J. M. Papadopoulos, A. Ruina, and A. L. Schwab, "Linearized dynamics equations for the balance and steer of a bicycle: A benchmark and review," *Proc. R. Soc. A Math. Phys. Eng. Sci.*, vol. 463, no. 2084, pp. 1955–1982, 2007.
- [2] K. Dussanapudi, K. N. Raju, M. Preeti, and A. Sastry, "Genesis of MEMS Accelerometers for Select the Optimal Accelerometer for Bio Applications," *Int. J. Innov. Technol. Explore. Eng.*, vol. 9, no. 2, pp. 4962–4970, 2019.
- [3] J. D. G. Kooijman, A. L. Schwab, and J. P. Meijaard, "Experimental validation of a model of an uncontrolled bicycle," *Multibody Syst. Dyn.*, vol. 19, no. 1–2, pp. 115–132, 2008.
- [4] Murata, "Murata Boy," p. 2020, 2009.
- [5] M. Boy, "MURATA BOY ' s Capabilities," pp. 1–2, 2020.
- [6] G. Mohanarajah, "The Cubli: a cube that can jump up, balance, and 'walk,'" pp. 3722–3727, 2013.
- [7] J. R. Chagdes, S. Rietdyk, M. H. Jeffrey, N. Z. Howard, and A. Raman, "Dynamic Stability of a human standing on a balance board," *J. Biomech.*, vol. 46, no. 15, pp. 2593–2602, 2013.
- [8] G. Belascuen and N. Aguilar, "Design, Modeling and Control of a Reaction Wheel Balanced Inverted Pendulum," *2018 IEEE Bienn. Congr. Argentina, ARGENCON 2018*, pp. 1–9, 2019.
- [9] J. Meyer, N. Delson, and R. A. de Callafon, *Design, Modeling, and Stabilization of a Moment Exchange Based Inverted Pendulum*, vol. 42, no. 10. IFAC, 2009.
- [10] J. C. Gamazo-Real, E. Vázquez-Sánchez, and J. Gómez-Gil, "Position and speed control of brushless dc motors using sensorless techniques and application trends," *Sensors*, vol. 10, no. 7, pp. 6901–6947, 2010.
- [11] M. D. Hasha, "High-Performance Reaction Wheel Optimization for Fine-Pointing Space Platforms : Minimizing Induced Vibration Effects on Jitter Performance plus Lessons Learned from Hubble Space Telescope for Current and Future Spacecraft Applications On-Orbit Satellite," *NASA Tech. Reports*, no. Proceed dings of the 43 rd Aerospace Mechanisms Symposium, NASA Ames Research Center, pp. 373–400, 2016.
- [12] S. Kivrak, H. Power, and G. Systems, "H Bridge DC Motor Driver Design and Implementation with Using dsPIC30f4011," no. May 2017.