Controlling two-wheeled self-balancing robot moving on inclined plane

Nguyen Duc To

***Abstract*—This research not only describes the design and implementation of two - wheeled self - balancing robot, but also shows how to simulate the platform in Simulink Matlab using Proportional–Integral–Derivative (PID) cascade control rule, then compares it with reality platform. A Kalman filter is used for state reconstruction in the final implementation. A cascaded PID control algorithm was proposed to combine the balancing and movement. The movement of the robot is controlled by using a distance controller that use rotary encoder sensor to measure its traveled distance. Besides the robot is able to move forward, backward, turning and reach the desired angle position by calculating the body's tilt angle. The experiment shows that the robot is likely to climb up slope with upon 25 degrees. Last but not least, this research also shows how to control the robot by using smartphone and C# form on laptop.**

***Index Terms*— Algorithm, PID cascade, C#.**

1 INTRODUCTION

The two-wheeled self-balancing robot has the characteristics of flexible and simple structure, working well in a small space, especially in the poor working conditions and complexity task places, such as exploration and dangerous field. Therefore, the robot not only has become an essential and classical experiment facility but also is a good platform for researchers to prove various kinds of control theory and control method. The research on two wheels balancing robot is based on inverted pendulum model. Thus, two wheels balancing robot needs a good controller to control itself in upright position without the forces from outside.

Contrary to this existing research, this study aims to make the robot platform with a cheaper budget, better results and providing feedback with all the angles and position values in detail for evaluation and comparison. After that, the platform can be used in lab rooms at HCM City University of Technology.

The structure of this paper is as follows: Section 2, an overview of the proposed method is presented to explain the whole procedure. This section also discusses each step of the proposed algorithm in detail. Design and construction of experimental models of the robot show in Section 3. Experimental results of the proposed system are presented in Section 4. Finally, Section 5 concludes the paper with suggestions for future research.

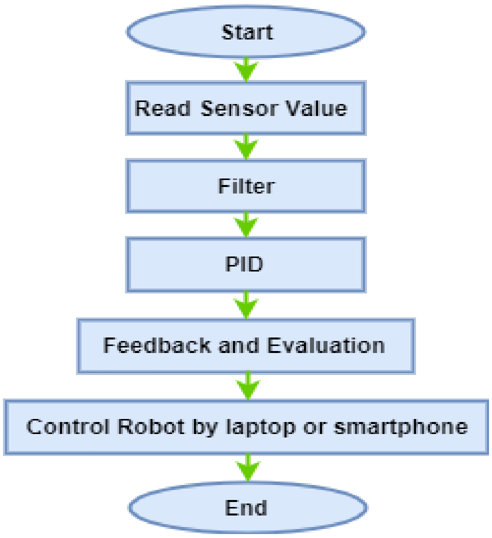


Fig 1. Block diagram of proposed method

2 PROPOSED METHOD

The block diagram of the proposed algorithm is shown in Fig. 1. As can be seen, the whole algorithm is comprised of five steps: (2.1) reading sensor value; (2.2) using filter to have the exactly value; (2.3) applying control rule to have the output to control the robot balancing and moving;

Manuscript Received on March 15th, 2017, Manuscript Revised on November 01st, 2017.

Nguyen Duc To, Ho Chi Minh City University of Technology, VNU–HCM, Viet Nam (e-mail: nguyenducto0123@gmail.com)

(2.4) feedback all the results for comparison and evaluation; (2.5) controlling robot by smartphone and laptop. Design and construction of experimental models of the robot show in Section 3. The experimental results in Section 4 will visualize clearer each step of the proposed processes achieved in this method.

2.1 Read Sensor

Established the I2C communication to start and read angle from the MPU6050 sensor.

At the point of the robot's balancing, the angle was read and set it into Control Angle.

The Gyroscope bias drifted so it had to be calibrated. First, when robot was keep standing still, the gyro values were read and set into GyroCalib.

After reading and calibrating the angle, the rest of this part concerns calculating the Angle. Accelerometer Angle:

$$Roll = \arctan 2(accY / accZ) \qquad (1)$$

Gyro Angle:

$$gyroAngle_k = gyroAngle_{k-1} + gyroRate_k * dt \quad (2)$$

The final Angle calculated by using Gyro Angle subtract to GyroCalib. After calculating the Gyro and Accelerometer Angle, the filter was used to combine it to compute the real and exact angle.

2.2 Filter

2.2.1 Complementary filter:

The complementary filtering is needed because the accelerometer is accurate in long term not short term. The gyroscope is just the reverse. Therefore, it's necessary to combine and filter the output of MPU6050 sensor.

The complementary filter is the combined low pass filter and high pass filter.

This filter combines accelerometer and gyroscope value to have final angle following this formula:

$$AngleC = A* \;\; AngleC + gyro * dt \;\; + \; 1 - A \; * Angle \qquad (3)$$

T value is the boundaries of trust about gyro and accelerometer, so the recommend for value of A is 0.98 to have the suitable T.

2.2.2 Kalman filter:

The Kalman filter is an algorithm which uses a series of measurements observed over time; in this context, it uses an accelerometer and a gyroscope. These measurements will contain noise that will contribute to the error of the measurement. The Kalman filter will then try to estimate the state of the system, based on the current and previous states, that tend to be more precise that than the measurements alone. It is a good way to deal with the accelerometer and gyroscope noise.

The Kalman filter includes two main steps: Time Update and Measurement Update

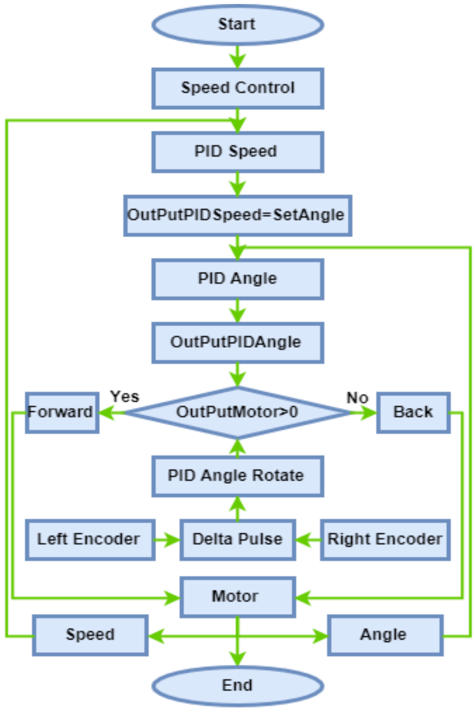


Fig 2. PID cascade

2.3 Proportional–Integral–Derivative (PID) Control System

PID control method has been widely used in the feedback control system. It is a very typical way of control strategy in industry [1]. PID is composed of three parts [2]: The proportional part which is used in the purpose of error elimination, the integral part is used to average past error, the derivative part is to predict the further error through past error variation. The final control value can be calculated by simply adding these three terms together.

Three PID controller loops are used to make the robot stand still and move [3]. The first PID was used to let the error between Angle and Angle control down to zero.

If the robot still moved forward and backward in balancing mode, the second PID controller was used to deal with this problem. It keeps the speed of robot to reach the speed control. The output of this speed PID is angle and became the input of the first PID Angle.

Unfortunately, if the hardware is not perfect, the speed of two motors was not the same so the third PID was applied [3]. The output of these PID controllers will bonus with the first two PID above to have the final output to motors. The Fig. 2 will show it.

2.4 *Feedback for evaluation and comparison*

After using PID, all of the PID parameters and result values display and setup through C form. Therefore, we can more easily to find the suitable PID parameter and control the response. The Fig. 3 has shown the method.

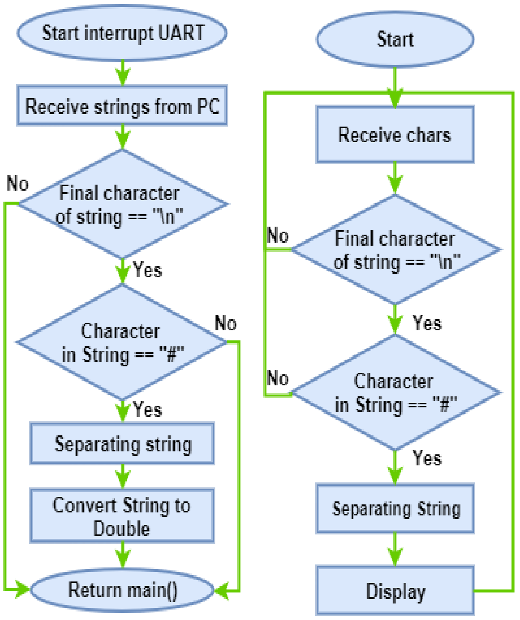
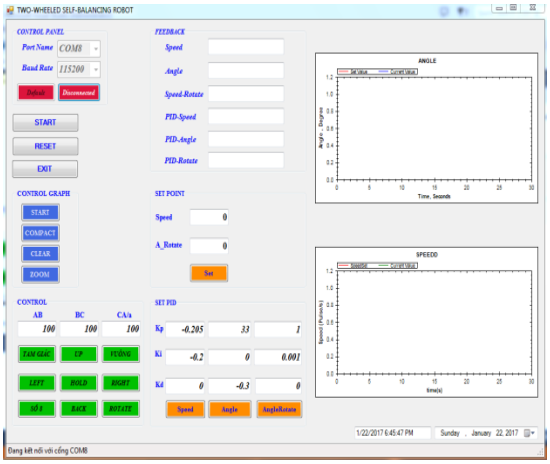


Fig 3. Transmit and receive algorithm

2.5 *Control robot*

The robot was not only controlled by using form load C# Fig. 4 but also by android app. When clicked the button, the "character" will be sent to the microcontroller and the robot will operate in suitable mode depending on the character it received.

Transmission and reception of signal between laptop or smartphone with robot via Bluetooth. So we can control it in a short distance about 10 meters radius.



Port Name COM8
Baud Rate 115200

SET POINT
Speed 0
A_Rotate 0

CONTROL
AB 100
BC 100
CA/a 100

SET PID
Kp -0.205 33 1
Ki -0.2 0 0.001
Kd 0 -0.3 0

1/22/2017 6:45:47 PM
Sunday , January 22, 2017

Đang kết nối với cổng COM8

Fig 4. Form C# on laptop

3 DESIGN AND CONSTRUCTION OF EXPERIMENTAL MODELS

3.1 *Mathematical model of the robot*

θ	rad	Raw angle
δ	rad	Yaw angle
M_W	kg	Mass of wheel
M_B	kg	Mass of body
R	m	Radius of wheel
L	m	Distance between the center of the wheels and the robot's center of gravity.
D	m	Distance between the contact patches of the wheels
g	m/s^2	Acceleration of gravity

Equation describes object:

$$\begin{cases} \left(\frac{3}{4}\frac{(M_W R+M_B L\cos\theta)\cos\theta}{2M_W+M_B\ \ L}-1\right)\ddot{\theta}=\frac{3}{4}\frac{M_B L\sin\theta\cos\theta}{2M_W+M_B\ \ L}\dot{\theta}^2+\left(\frac{3}{4}\frac{1+sin^2\theta}{M_B L^2}+\frac{3}{4}\frac{\cos\theta}{2M_W+M_B\ \ RL}\right)C_\theta \\ \qquad -\frac{3}{4}\frac{g\sin\theta}{L} \\ \left(2M_W+M_B-\frac{3}{4}\frac{(M_W R+M_B L\cos\theta)\cos\theta}{L}\right)\ddot{X}=M_B L\sin\theta\,\dot{\theta}^2-\frac{3}{4}\frac{g(M_W R+M_B L\cos\theta)\sin\theta}{L} \\ \qquad +\left(\frac{3}{4}\frac{(M_W R+M_B L\cos\theta)(1+\sin^2\theta)}{M_B L^2}+\frac{1}{R}\right)C_\theta \end{cases} \tag{4}$$

Linearizing nonlinear model of two-wheeled self-balancing robots

With conditions:

$$\theta \ll 1[rad]$$

$$\Rightarrow \sin\theta \approx \theta;\sin^2\theta \approx \theta;\cos \approx \theta$$

Equation (5) becomes:

$$\begin{cases} \ddot{X}=g\theta-\frac{4}{3}L\ddot{\theta}-\frac{C}{M_B L} \\ 2M_W+M_B\ \ \ddot{\theta}=\frac{2M_W+M_B}{M_W R+M_B L}\ddot{X}+\frac{C}{R\ \ M_W R+M_B L} \end{cases} \tag{5}$$

Solve the system of equations (6) we obtain:

$$\begin{cases} \ddot{\theta}=\frac{gM_B}{X}\theta-\frac{Y}{X}C_\theta \\ \ddot{X}=\left(-\frac{4}{3}L\frac{gM_B}{X}+g\right)\theta+\left(\frac{4LY}{3X}-\frac{1}{M_B L}\right)C_\theta \end{cases} \tag{6}$$

With:

$$\begin{cases} X=\frac{4}{3}M_B L-\frac{M_B}{2M_W+M_B}\ \ M_W R+M_B L \\ Y=\frac{M_B}{2M_W+M_B\ \ R}+\frac{1}{L} \end{cases}$$

3.2 Block diagram of system

The system block diagram is show in Fig. 5. Functions of the blocks in the diagram:

- Transmit / Receive Universal Asynchronous Receiver/Transmitter (UART): transmit code, signal via serial communication protocol with the 8 bits data.
- Acceleration and Gyroscope: implementing measure the angle and angular velocity of the robot.
- H – bridge: control velocity and direction of motor.
- Encoder: read pulse from the encoder and transmit position, velocity of the robot to the microcontroller.
- Microcontroller: read the signals from acceleration, gyroscope, encoder, transmits signals to the UART, processing the input signal and provides output signals

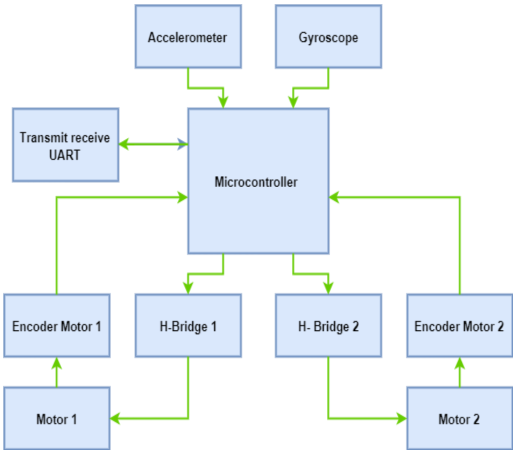


Fig 5. Block diagram of system

3.3 Hardware architecture of robot

Hardware includes modules [4, 5]: read encoder of two DC servo motors, PWM and DIR to control two DC servo motors, read angle value of the sensor - MPU6050, communicate with Bluetooth HC-05, source... to connect peripheral devices.

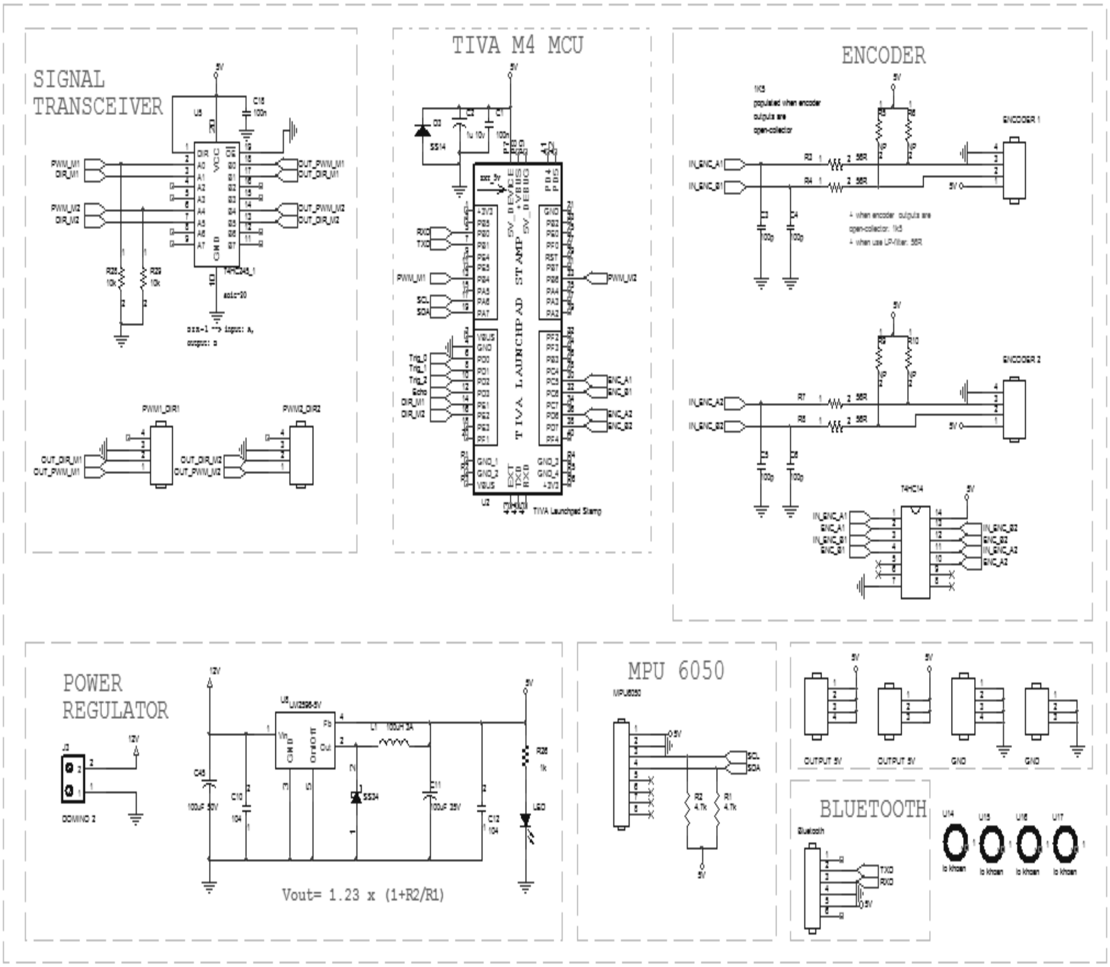


Fig 6. Schematic

3.4 *Mechanical design*

In this model, the structure of a self-balancing robot is expressed. Mechanical parts of the car are designed by AutoCAD software and simulated 3D by SketchUp software.

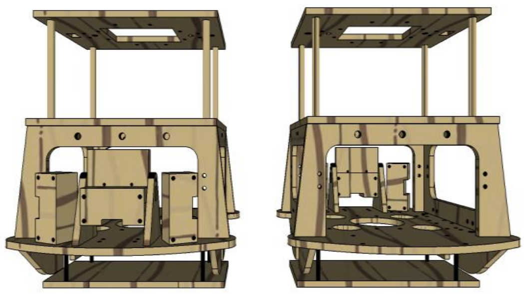


Fig 7. Simulate 3D of robot

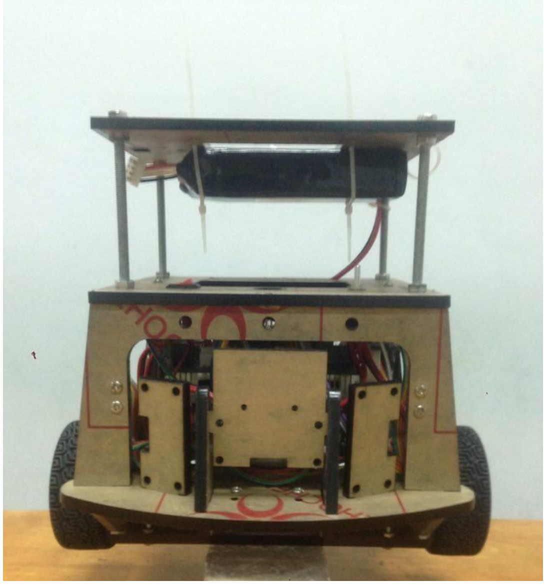


Fig 8. Real model

4 EXPERIMENTAL RESULTS

4.1 *Complementary file and Kalman filter*

The Fig. 9 and Fig. 10 show the angle output after Kalman filter. In general, the angle output is stable and good.

The Fig. 11 shows the Complementary filter is not as accurate as the Kalman filter once when the Angle began changing continuously.

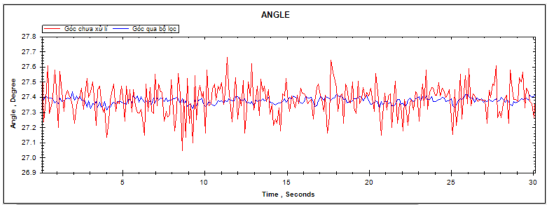


Fig 9.Robot doesn't oscillate

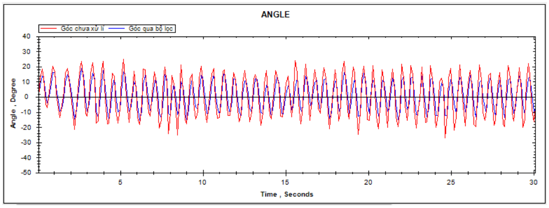


Fig 10. Robot oscillates

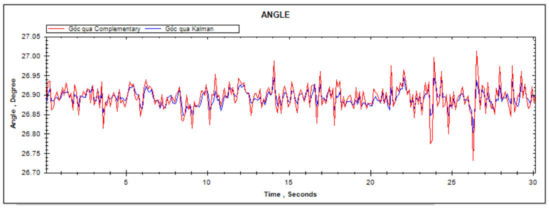


Fig 11. Complementary filter and Kalman filter

4.2 *PID cascade*

The Fig. 12 and Fig. 13 show the response of robot when we use PID speed and PID Angle.

In the first experiment, it is verified how the robot behaves given zero target velocity. It is seen that velocity and tilt of the robot slightly oscillate in control methods. The robot cannot be absolutely immobile since its state is never statically stable.

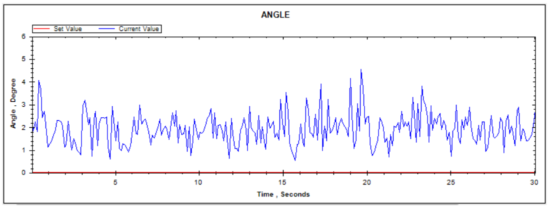


Fig 12.The angle of robot when it does not move.

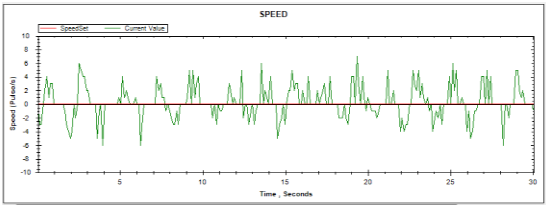


Fig 13.The speed of robot does not move

When we use two PID controllers at the same time the robot will become stable at the balance point. On one hand, the robot still remains stable after being pushed. On the other hands, if only use PID Angle is used, the robot cannot stay stable almost immediately after being push.

4.3 *Robot moving on plane*

Fig. 14 and Fig. 15 show the robot's response when it was moving forward, backward.

Besides that, it also shows the position of the robot when it moves. The robot still keeps balance while it was moving.

Because of three PID controller loops, the robot will move straight to forward and backward. This is the different compare with when we are not use PID Angle rotate. The robot will turn left or right when it moves for a long distance.

The robot meets installation parameters well. Travelled distance moves forward steadily. The robot's rotation on two wheels always remain stable (delta Pulse towards 0); on the other hand, the robot's angle is stable too.

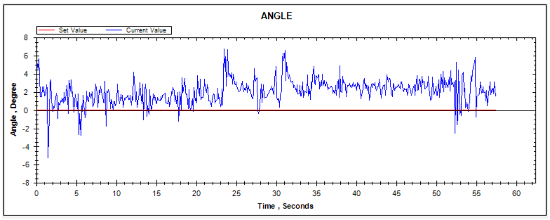


Fig 14. The angle of robot moves with ± 20 (cm/s)

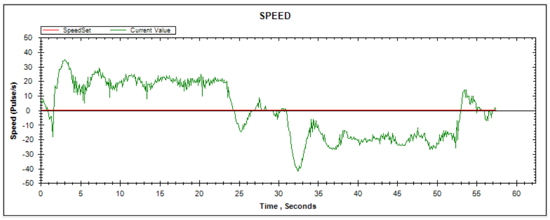


Fig 15. The speed of robot moves with ± 20 (cm/s)

The robot is able to stabilize itself and managed to reject disturbances such as gentle pushes.

Fig. 16 and Fig. 17 show the robot's response when it was pushed by external factors.

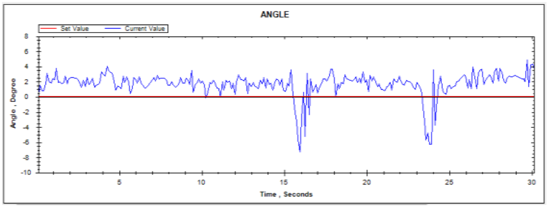


Fig 16. The angle of robot when being pushed

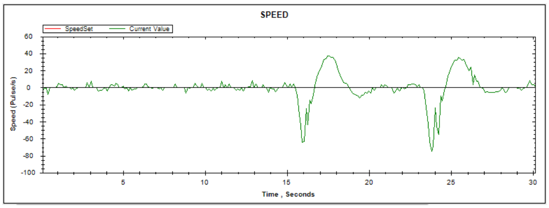


Fig 17. The speed of robot when being pushed

Fig. 18 and Fig. 19 show the robot's response when it was rotated.

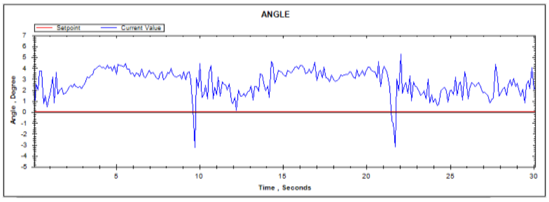


Fig 18. The angle of robot when being rotated

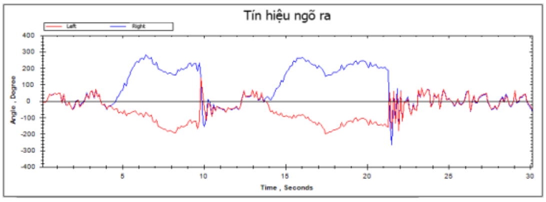


Fig 19. The speed of robot when being rotated.

4.4 *Robot moving on inclined plane*

The Fig. 20 and Fig. 21 shows that angle of robot is greater than angle of slope when moving up slope. The robot tilted back to reduce speed when moving down slope and when robot switched from plane to inclined plane or from inclined plane to plane, it has time of transition.

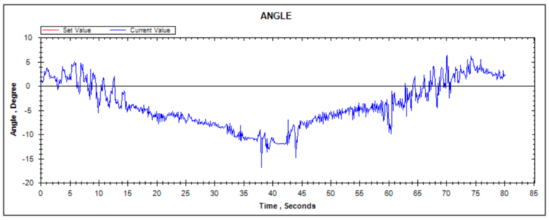


Fig 20. The angle of robot moves with ± 5 (cm/s)

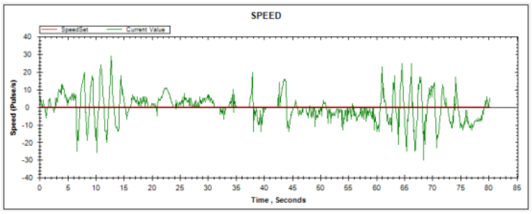


Fig 21. The speed of robot moves with ± 5 (cm/s)

5 CONCLUSION AND FUTURE WORK

This project has successfully presented that two – wheeled self – balancing robot can move on plane, inclined plane and preset orbit. The robot can itself balance when it was changed gravity by force from outside. It also can be controlled from smartphone or laptop.

However, there are still some shortcomings. Although the robot can move forward and backward on inclined plane, it cannot turn left or turn right on inclined plane. Although robot can move follow some preset orbit such as square, triangular orbit with edge which can change by people or orbit of number eight…The orbit can only observe directly by eye, cannot draw it on computer.

This is a very challenging problem that should be studied more in the future research. It is also necessary to control exactly the position of the robot so that it can be controlled to move to narrow places where people cannot move.

Furthermore, the proposed straightening algorithm is built on a small platform with affordable and quality material, so that group students can afford to have a project to apply their knowledge.

REFERENCES

[1] Nguyen Thi Phuong Ha and Huynh Thai Hoang, *Automatic Control Theory*, Vietnam National University, 2005.

[2] Huynh Thai Hoang, *Slide Advanced Control Theory*, University of Technology, 2011.

[3] Duong Duy Chien, *Two-wheeled self-balancing robot moving on plane*, Electrical & Electronic Engineering, University of Technology, 2016.

[4] Mai Tuan Dat, *Two – wheeled self – balancing robot moving on plane*, Mechatronics Engineering, University of Technology, 2002.

[5] Nguyen Gia Minh Thao, *Two–wheeled self–balancing robot*, Electrical & Electronic Engineering, University of Technology, 2010.

Duc To-NGUYEN was born in Quang Ngai Province, Vietnam in 1994. He received the degree in Control Engineering and Automation from Ho Chi Minh City University of Technology, VNU-HCM in 2017. From 2012 to 2017, he was a student at Ho Chi Minh City University of Technology, VNU-HCM. His research interests include Advanced Control Theory, Intelligent Control, Control Systems.

Điều khiển rô-bốt hai bánh tự cân bằng di chuyển trên mặt phẳng nghiêng

Nguyễn Đức Tô

Tóm tắt**—** Nghiên cứu này không những mô tả thiết kế và thực hiện robot hai bánh tự cân, mà còn cho thấy cách mô phỏng trên Simulink Matlab sử dụng các luật điều kiển PID cascade và so sánh với mô hình thực tế. Bộ lọc Kalman được sử dụng để lọc nhiễu cho góc nghiêng của robot. Thuật toán điều khiển PID cascade được sử dụng để kết hợp cân bằng và chuyển động. Sự di chuyển của robot được điều khiển bằng cách sử dụng một bộ điều khiển sử dụng encoder để đo khoảng cách di chuyển của nó. Bên cạnh đó, robot có thể di chuyển về phía trước, lùi, quay và đạt được vị trí góc mong muốn bằng cách tính góc nghiêng của robot. Thí nghiệm cho thấy robot có khả năng di chuyển trên mặt phẳng nghiêng với 25^0. Cuối cùng nhưng không kém phần quan trọng, nghiên cứu này cũng cho thấy cách điều khiển robot bằng smartphone và laptop.

Từ khóa**—** Algorithm, PID cascade, C#.