e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
Volume:03/Issue:06/June-2021 Impact Factor- 5.354 www.irjmets.com

STUDY OF SELF-BALANCING MOVING ROBOT BY USING PID ALGORITHM

Hue - Nguyen Thi*[1], Song - Ngo Duy*[2], Thoi - Le Nam*[3]

*[1,2,3]Faculty Of Technology, Dong Nai Technology University, Dong Nai, Vietnam.

ABSTRACT

This paper presents a two-wheel self-balancing robot, designs and manufactures complete details controlled by stepper motor. This design improves on an earlier model based on a DC motor that was used during the year, helping students learn electronics, program computers, and build this robot. The new design improves performance and reduces the total cost of the device. This robot is researched by us for the purpose of mass transport, helping students learn electronics, computer programming, and building robots.

Keywords: Programming, Modeling, Signal Processing, Self-Balancing.

I. INTRODUCTION

Self-balancing two-wheeled robot is an automatic device that moves on two wheels. It works on the principle of inverted pendulum balance [1-2]. This is an "under-actuated" system, which is multivariable, nonlinear, and unstable. Self-balancing robot is similar to an inverted pendulum. Unlike a normal pendulum that continues to swing when force is applied, this pendulum cannot balance itself, it will fall. So how do we balance it [1-4]. It's almost like when we balance a pen on our index finger, it's a good example of balancing an inverted pendulum [5]. We move our finger in the direction where the bar falls. Similar to the case with self-balancing robots, the robot will fall forward or backward. We balance the robot by controlling the wheel in the direction it falls. What we are trying to do here is keep the center of gravity of the robot perpendicular to the ground [1-4].

To control the motor, we need some information about the state of the robot. We need to know the direction in which the robot is falling, how much the robot is leaning, and the speed at which it falls. All of this information can be inferred from the data obtained from the MPU6050 [5]. We combine all these inputs and generate a signal that controls the motor and keeps the robot balanced. This robot has been developed by companies into a commercial vehicle called Segway with several popular designs [2-4]. However, this system is still being interested by scientists to research and test many different control algorithms from linear to nonlinear as well as intelligent control algorithms. These algorithms mainly focus on controlling the balance, controlling the position and direction of the robot [1], [3-5].

II. DESIGN METHODS

a) Structure of PID controller for self-balancing two-wheeled robot

Three sets of PIDs are used to control the self-balancing two-wheeled robot, including:

PID controller for tilt angle (ψ)

Position control PID unit (θ)

Rotation angle control PID unit (ϕ)

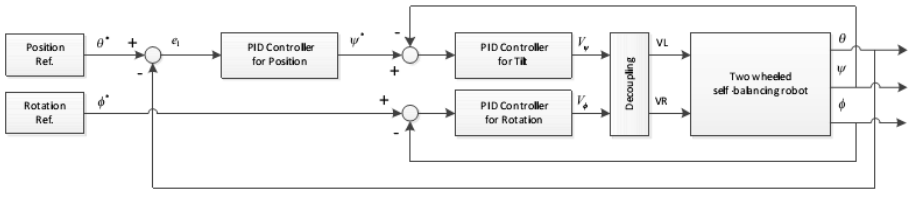

Figure 1: Structure of the PID controller for a self-balancing two-wheeled robot system

Continuous PID controller transfer function:

$$G_{PID}(S) = \frac{U(S)}{E(S)} = K_P + \frac{K_I}{s} + K_D\left(\frac{s}{1+\tau S}\right) \quad [1]$$

Discrete the derivative with respect to time:

$$y(k) \approx \frac{f(k)-f(k-1)}{T_s} \quad [2]$$

Discrete integral over time:

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
Volume:03/Issue:06/June-2021 Impact Factor- 5.354 www.irjmets.com

$\int_0^{k.T_s} y(t)dt = y(k) \approx y(k-1) + \frac{f(k)-f(k-1)}{2}.T_s$ [3]

Discrete Transform (z-Transform)

$X[z] = Z\{x[k]\} = \sum_{k=0}^{\infty} x[k]z^{-k}$ [4]

$z = Ae^{j\theta} = A(\cos\theta + J\sin\theta)$ [5]

$Z\{x[k-n]\} = z^{-n}X[z]$ và $Z\{x[k]\} = X[z]$ [6]

Therefore

$y(k) = \frac{f(k)-f(k-1)}{T_S} \rightarrow y[z] = \frac{F[z]-z^{-1}F[z]}{T_S} \Longrightarrow \frac{y[z]}{F[z]} = \frac{z-1}{zT_S}$ [7]

and

$y(k) = y(k-1) + \frac{f(k)-f(k-1)}{2}.T_s \rightarrow Y[z](1-z^{-1}) = \frac{T_s}{2}F[z](1+z^{-1}) \Rightarrow \frac{y[z]}{F[z]} = \frac{T_s}{2}\frac{z+1}{z-1}$ [8]

$\frac{U[z]}{E[z]} = K_p + K_i\frac{T_s}{2}\frac{z+1}{z-1} + K_d\frac{z-1}{zT_s}$ [9]

$\Leftrightarrow \frac{U[z]}{E[z]} = \frac{K_p(z^2-z)+K_i\frac{T_s}{2}(z^2+z)+\frac{K_d}{T_S}(z^2-2z+1)}{z^2-z}$

$\Leftrightarrow \frac{U[z]}{E[z]} = \frac{\left(K_p+K_i\frac{T_s}{2}+\frac{K_d}{T_S}\right)z^2+\left(-K_p+K_i\frac{T_s}{2}+\frac{K_d}{T_S}\right)Z+\frac{K_d}{T_S}}{z^2-z}$

$\Leftrightarrow \frac{U[z]}{E[z]} = \frac{\left(K_p+K_i\frac{T_s}{2}+\frac{K_d}{T_S}\right)+\left(-K_p+K_i\frac{T_s}{2}+\frac{K_d}{T_S}\right)z^{-1}+\frac{k_d}{T_S}z^{-2}}{1-z^{-1}}$

$\Leftrightarrow U[z] = z^{-1}U[z] + aE[z] + bz^{-1}E[z] + cz^{-2}E[z]$ [10]

$\Leftrightarrow u[k] = u[k-1] + ae[k] + be[k-1] + ce[k-2]$ [11]

Inside: $a = K_p + K_i\frac{T_s}{2} + \frac{K_d}{T_s}$; $b = -K_p + K_i\frac{T_s}{2} + \frac{K_d}{T_s}$; $c = \frac{K_d}{T_s}$ [12]

b) PID controller with fixed parameter

PID controller with fixed parameters KP, KI, KD as above, the system only

good work in the condition that the coefficients KP, KI, KD have been optimally adjusted and during the working process, the parameters in the model remain constant. Structure of a discrete PID controller with a fixed factor used to control the angle tilt, position and rotation are the same and are implemented as follows in Matlab Simulink.

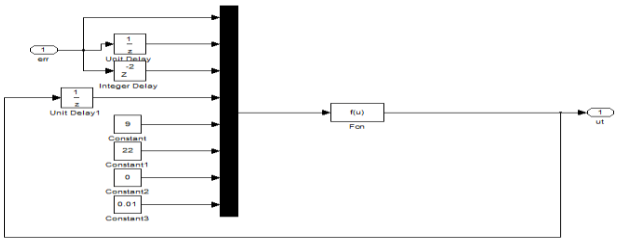

Figure 2: Internal structure of a discrete PID controller with fixed parameters

Self-balancing two-wheeled robot control system using discrete PID with The actual fixed parameters are as follows:

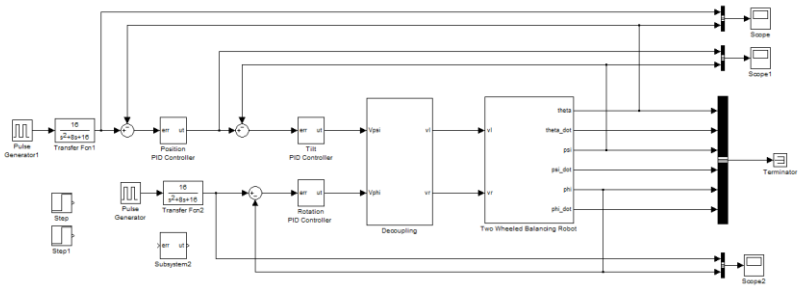

Figure 3. Self-balancing two-wheeled robot using 3 fixed PID controllers

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
Volume:03/Issue:06/June-2021 Impact Factor- 5.354 www.irjmets.com

In particular, the self-balancing 2-wheel robot model in this case will be changed parameters such as the robot body mass (M) and the coefficient of friction between the wheel and the moving surface (fw) through the file " Two_wheelRobot.m" It can be seen that the system position output response is significantly affected, the system position output cannot continue to respond to the original set signal when the fw factor changes. As for the tilt angle control PID units, it still ensures that the system output follows the set signal well. Especially when there is a change in the robot's mass (M), the position output of the system is greatly overstated. However, the ability to control the tilt angle of the PID unit still maintains and sticks well to the set signal. Thus, in case there is a change in the coefficient of friction fw and the volume of Robot M in the model, the position control fixed PID cannot guarantee the control quality. However, the tilt angle and rotation angle respond quite well.

The robot uses two sensors: an accelerometer and a gyroscope to measure tilt angle and angular velocity. However, the problem is that it is necessary to combine the information from the two sensors to accurately determine the true tilt angle of the robot system, eliminating the influence of measurement noise and process noise. To solve this problem, the Kalman filter algorithm is used, with the aim of estimating the tilt angle value of the robot system from the two types of sensors above and eliminating the influence of noise. The Kalman filter is investigated with the model of three state variables as follows:

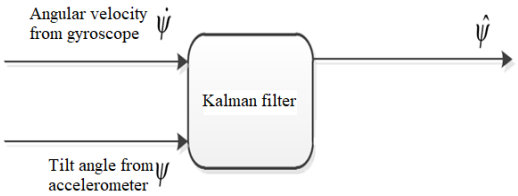

Figure 4. Kalman filter model with 3 state variables

With this model, the filter uses 2 input variables, which are the angular velocity from the gyroscope sensor and the tilt angle from the accelerometer; 1 output variable is the estimated tilt angle

With $P_{Khoitao} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ [13]

Angle q_bias is the angle and velocity used in the filter calculation, respectively.

R stands for the covariance noise value. In this case, it is the expected 1x1 matrix having a value of 0.08 rads ≈ 4.5 degrees from the accelerometer:

R_angle=0.08 [14]

Q is a 2x2 matrix representing the covariance noise. In this case, it indicates the confidence level of the accelerometer in relation to the gyro sensor:

$$Q = \begin{bmatrix} q\_angle & 0 \\ 0 & q\_gyro \end{bmatrix} = \begin{bmatrix} 0.001 & 0 \\ 0 & 0.003 \end{bmatrix} \quad [15]$$

III. DETAILED STRUCTURE

DC motor driver circuit

The L298N DC motor driver circuit is capable of controlling 2 DC motors, maximum current 2A each, protection diode integrated circuit and 7805 power IC to supply 5VDC power to other modules (only use this 5V if power supply <12VDC).

L298N DC motor driver circuit is easy to use, low cost, easy to install, is the optimal choice in the price range.

Figure 5: L298N DC motor driver circuit

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
Volume:03/Issue:06/June-2021 Impact Factor- 5.354 www.irjmets.com

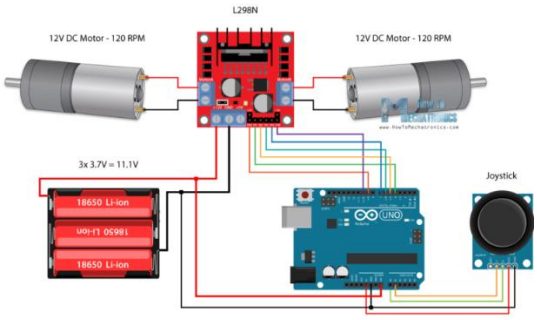

Figure 6: L298N DC motor driver circuit diagram

IV. RESULTS

Mechanical design

The frame uses 4mm clear mica and color mica, which is fixed by the slotted mechanism. Encorder drive mechanism presses the wheel and arranges the circuit in the vertical direction.

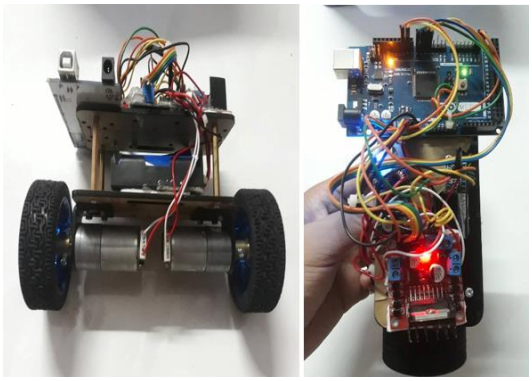

Figure 7. Realistic Robot Model

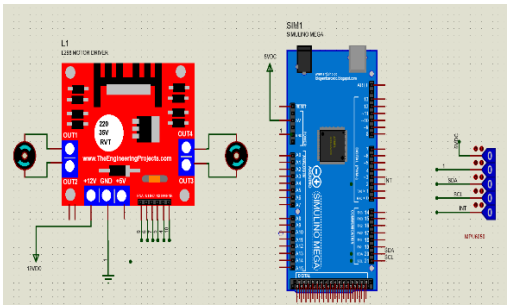

Figure 8. Simulated connection diagram on Proteus 8 Professional

Based on the foundation of the PID control algorithm, giving results by balancing and controlling the control results according to the initial position on the robot model has been built according to the following diagram:

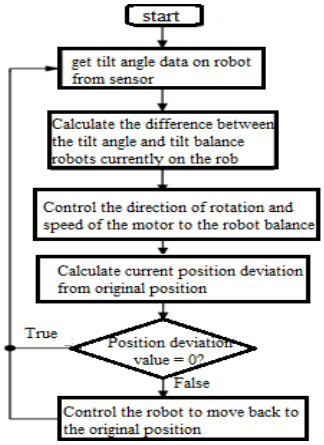

Figure 9. Robot model building flowchart

First as all, we will complete the circuit and structure of the robot. The robot is assembled on three frames spaced 25mm apart using nylon spacers. The bottom layer contains two motors. The middle layer has a controller, an IMU and a 5V boost regulator module. The top layer contains the battery, a switch, and an ultrasonic proximity sensor. Before we start testing, we need a clear diagram of where each part is to be placed. Based on the assembly diagram of the parts and welding, connect the three frames together using nylon spacers. You may have noticed that I used two separate voltage regulation modules to drive the motor and controller even though both require 5V power. Because when using a single 5V boost regulator to power the controller as well as the motors, the program will get an error. This is due to the noise generated by the motor circuit acting on the controller and the IMU. This was effectively eliminated by decoupling the voltage regulator for the controller and motor and adding a 10uF capacitor at the power terminals to the motor.

V. CONCLUSION

The research results of the basic inverted pendulum model, such as the car-pendulum model, the rotating inverted pendulum ...can be applied and inherited to other similar models but have more practical application, such as rocket model, self-balancing two-wheeled vehicle model, etc., thus overcoming the inherent disadvantages of classic two- or three-wheeled robots. Classic two- or three-wheeled robots, which are composed of a drive wheel and a free wheel (or whatever else) to support the robot's weight. If a lot of weight is placed on the rudder, the robot will be unstable and easy to fall, and if put on many tail wheels, the two main wheels will lose their ability to grip. Many robot designs can move well on flat terrain but cannot move up and down on convex or inclined terrain. When moving up a hill, the robot's weight is on the rear of the vehicle, causing it to lose its ability to grip and slip.

Two-wheel self-balancing robot, fully detailed design and manufacture controlled by stepper motor. This design improves on an earlier model based on a DC motor that was used during the year. This robot is researched by us for the purpose of mass transport, helping students learn electronics, computer programming, and building robots.

VI. REFERENCES

[1] Hau-Shiue, J. and Kai-Yew, L. (2013). Design and control of a two-wheel self-balancing robot using the arduino microcontroller board. 10th IEEE International Conference on Control and Automation (ICCA), 634-- 639.

[2] T. Takei, Ryoko Imamura; Sh. Yuta. Baggage Transportation and Navigation by a Wheeled Inverted Pendulum Mobile Robot. IEEE Transactions on Industrial Electronics, Volume 56, Issue 10. Page(s): 3985 – 3994. 21 July 2009.

[3] Manabe, S. (1995). A low-cost inverted pendulum system for control system education. In Advances in Control Education 1994 (pp. 21-24).

[4] Lilienkamp, K.A. and Lundberg, K. (2004). Low-cost magnetic levitation project kits for teaching feedback system design. Proceedings of the American Control Conference.

[5] Nash, Jr., J. F. (1950). Equilibrium points in n-person games. Proceedings of the National Academy of Sciences 36, 48–49.