Comparison of Control Methods PD, PI, and PID on Two Wheeled Self Balancing Robot

Bhakti Yudho Suprapto
Electrical Engineering Department
University of Sriwijaya
Palembang, South of Sumatera, Indonesia
bhakti_yudho@yahoo.com

Djulil Amri
Electrical Engineering Department
University of Sriwijaya
Palembang, South of Sumatera,
Indonesia
Amfz_07@yahoo.co.id

Suci Dwijayanti
Electrical Engineering Department
University of Sriwijaya
Palembang, South of Sumatera,
Indonesia
Suci.dwijayanti@gmail.com

***Abstract*—A robot must employ a suitable control method to obtain a good stability. The Two-Wheeled Self Balancing Robot in this paper is designed using a MPU-6050 IMU sensor module and ATmega128 microcontroller as its controller board. This IMU sensor module is employed to measure any change in the robot's tilt angle based on gyroscope and accelerometer readings contained in the module. The tilt angle readings are then utilized as the setpoint on the control methods, namely PD (Proportional Derivative), PI (Proportional Integral), or PID (Proportional Integral Derivative). Based on the conducted testing results, the PID controller is the best control strategy when compared to the PD and PI control. With parameters of Kp = 14, Ki = 0005 and Kd = 0.1, the robot is able to adjust the speed and direction of DC motor rotation to maintain upright positions on flat surfaces.**

***Keywords*— Two-Wheeled Self Balancing Robot; PID; PD; PI; IMU**

I. Introduction

Two-Wheeled Self Balancing Robot is a two-wheeled robot that can balance its positions automatically based on changes and shifts in its balance point. *This type of robots* is an under-actuated systems that can maintain its posture and drive the robot with only two wheels. In order to overcome the limitation in turning velocity due to the centrifugal force effect [5].

To make a *Two-Wheeled Self Balancing Robot*, a really good control method is required so that the robot can automatically maintain its position perpendicular to flat surfaces. Some control methods which can be employed in the robot control system include PD (Proportional Derivative), PI (Proportional Integral), and PID (Proportional Integral Derivative). Application of the control methods should be adapted to the employed system or plant, because each system or plant always deals with different disturbances so it requires adjustment to the employed control method.

Analysis of comparison of PD (Proportional Derivative), PI (Proportional Integral), and PID (Proportional Integral Derivative) control methods is intended to obtain an excellent and suitable control method for use in controlling the speed and direction of DC motor's rotation in *Two-Wheeled Self Balancing Robot*. The problem faced is how to make the robot to keep balanced and move only using two wheels when it is under perturbations (a touch or a boost) in its body.

To be able to stand upright, the robot uses sensors which are usually known as IMU (*Inertial Measurement Units*). These sensors are employed to reduce the centrifugal force influences given to the *Balancing Robot* [1]. Other sensors employed to keep the robot upright balance include gyro and encoder [2]. The robot has the goal to stand upright by controlling the speed and direction of motor rotation. Certainly, a good controller is required.

Most studies employed the PD, PID [2] [3], and LQR control methods [1] [4]. However, this study compares those controllers to obtain one that can control a robot keeping a good balance. The performance of PID based control that this paper proposes is measured on the length of time the robot can be stable when standing. The PID controller will control the motor velocity through pulse width modulation (PWM) based on IMU sensor readings in order to provide stability to the robot.

II. Hardware design

A. Inertial Measurement Units (IMU)

The Inertial Measurement Unit is a device used to measure angular rate, orientation, and gravity. The IMU sensor is divided into two units, namely *accelerometer* and *gyroscope* [6]. The IMU is the main component in the tilt navigation system utilized on aircraft or ships. The IMU has also been widely used in current smartphones.

B. Complementary Filter

A typical application of the complementary filter is to combine measurements of vertical acceleration and barometric vertical velocity to obtain an estimate of vertical velocity [7]. The *Complementary filter* is one type of filter that can be employed to combine measurements or filter the IMU (Inertial Measurement Units) readings, which can set the screen orientation based on tilt and angular rate. The IMU itself consists of two main sensors, namely *accelerometer* and *gyroscope (already mentioned).*

The *accelerometer* can provide measurements of tilt angle accurately when the system is in static mode. When the system is rotating or moving, the *accelerometer* cannot follow the rapid movement due to slow response and noise.

The *gyroscope* can read dynamic angular rate. After computation using the integral data from time to time, the movement angle or tilt angle can be calculated. But the resulting tile angle will be inaccurate in long term due to bias effects found on the *gyroscope*. In other words, the tilt angle measurement using the *gyroscope* can cause shift or deviation in starting point of tilt angle or so-called drift error.

To get ideal and accurate tilt angle readings, a *complementary filter* is employed by processing the data from the *accelerometer* and *gyroscope,* utilizing the gyroscope data in a short period of time because it is very precise and not easily affected by external interference and in long term using the accelerometer data because it has no drift error in measurement.

Here is the simplest form of *complementary filter* :

*angle = a * (angle + gyrData * dt) + (a-1) * (accData)* (1)

time = a.dt / 1-a (2)

The data obtained from the *gyroscope* is added to the actual data of tilt angle in each time increment. After that, the data is combined with the data of the low pass filter from the *accelerometer*.

C. Controller Scheme

PID controller is a closed loop feedback controller for a linear system. PID controller calculates the error between measured values with its desired set point and attempts to correct the calculated error. PID controller stands for proportional, integral and derivative controller [6].

Based on [9], proportional, integral and derivative control characteristics are summarized as follows:

- Proportional control deals with present error. Proportional factor is the product of gain and measured error. Hence, larger proportional gain has faster response time and smaller steady state error but causes overshoots over the desired set point. Setting the proportional gain too high causes a system to oscillate around the set point without settling. For a controller with proportional control action. The relationship between the output of the controller u(t) and the actuating error signal e(t) is [10]

$$u(t) = K_P e(t) \tag{3}$$

Or, in Laplace-transformed quantities, Where K_P is termed the proportional gain

- Integral control deals with accumulation of past errors. When error is too small, proportional factor output becomes negligible, which causes steady state error. Integral factor is the product of gain and summation of past errors. Hence, it corrects even a very small error and eliminates the steady state error. Similar to proportional controller, setting the integral gain too high causes overshoots over the set point. In a controller with integral control action, the value of the controller output u(t) is changed at a rate proportional to the actuating error signal e(t),[10]

$$\frac{du(t)}{dt} = K_i e(t) \tag{4}$$

- Derivative control deals with prediction of future errors. Derivative factor is the product of gain and rate of change of error. Therefore, it is use to reduce the overshoot caused by proportional and integral factor. The downside is that, derivative gain amplifies noise as well, which can cause the system to become unstable if the gain is too high. The equation of derivative control is given by [10]:

$$u(t) = K_P T_d \frac{de(t)}{dt} \tag{5}$$

- The downside of PID control is that when there is a large change in set point, the integral factor will accumulates a large error during response time and eventually overshoot. It will continue to increase over the set point until the accumulated error is decreased by errors in other direction. This situation is called integral windup. The combination of proportional control action, integral control action, and derivative control action is termed Proportional Control Derivative (PID) control action. This combined action has the advantages of each of the three individual control actions. The equation of a controller with this combined action is given by [10]

$$u(t) = K_P e(t) + \frac{K_P}{T_i}\int_0^t e(t)\,dt + K_P T_d \frac{de(t)}{dt} \tag{6}$$

III. Research Method

The process of designing the whole system of Two-Wheeled Self Balancing Robot has several stages, starting with hardware design and software design. The hardware design includes mechanical and electronic system design. The mechanical system design is needed to determine the sizes of required materials and components to reduce risks of installation errors or putting components on the tool to be employed. The electronic system design is required to determine the electronic components to be employed in the *Two-Wheeled Self Balancing Robot* including sensors, controllers, and actuators that will be employed. The whole system design in the *Two-Wheeled Self Balancing Robot* can be seen in Figure 1.

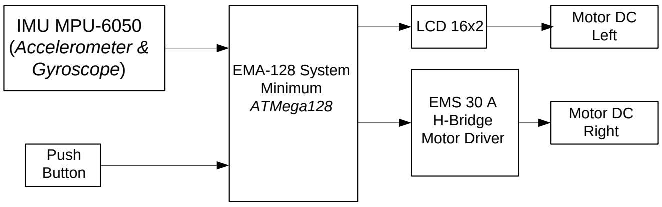

IMU MPU-6050
(Accelerometer & Gyroscope)
Push Button
EMA-128 System Minimum ATMega128
LCD 16x2
EMS 30 A H-Bridge Motor Driver
Motor DC Left
Motor DC Right

Fig. 1. Diagram of Design System of Self Balancing Two-Wheeled Robot

A. Mechanical Design

The mechanical system design on *Two-Wheeled Self Balancing Robot* covers the whole body of the robot design and layout of electronic components. Most mechanical making utilizes wood because this material is readily available, lightweight and low cost.

The robot's whole body consists of three main parts, namely lower base, connecting pole, and upper base. The size of lower and upper base is 300 mm x 100 mm and the height of connecting pole between lower and upper base is 400 mm. The size of wheel diameter is about 10 mm.

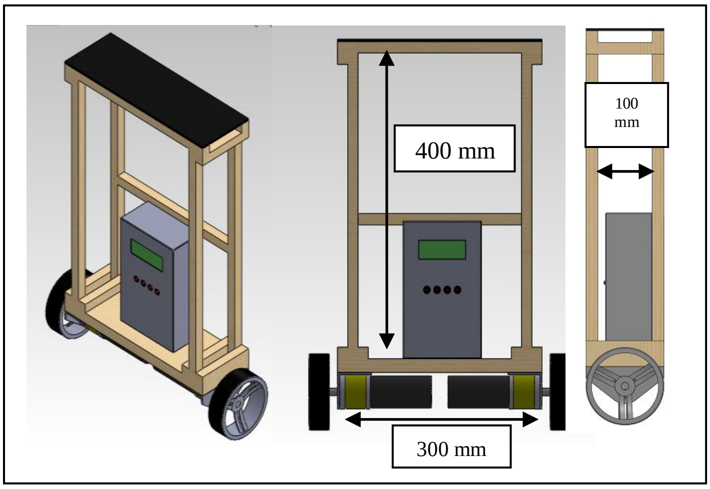


Fig. 2. Whole Body Mechanical Design of Two-Wheeled Self-Balancing Robot

B. Design of MPU-6050 IMU Sensor Module

MPU-6050 IMU sensor module is an integrated motion detector which consists of *3-axis gyroscopes*, *3-axis accelerometer*, and *digital motion processor* [8]. By using the I^2 C data path, MPU-6050 is also designed to interface with some non-inertial sensors such as pressure sensors. MPU-6050 has 16 bit ADC (Analog to Digital Converter) for processing digital output of *gyroscope* and *accelerometer sensors*.

For precise result detection between fast and slow movements, MPU-6050 has selection feature on measurements scale of *gyroscope* and *accelerometer*. The scale measurement on *gyroscopes* has a range of ± 250 ^{o} / sec, ± 500 ° /sec, ± 1000 ° /sec, and ± 2000 ^{o} /sec, while the measurement scale on *accelerometer* has a range between 2 g, 4 g, 8 g, and 16 g [8].

C. Control Design

The design of control method on *Two-Wheeled Self Balancing Robot* is something that must be done before programming the robot. To obtain a good response from the robot, selection of a suitable control method is needed. In this case the analyzed control methods are PD (Proportional Derivative), PI (Proportional Integral), and PID (Proportional Integral Derivative).

By using the control method of PD, PI, or PID, the setpoint employed in the robot's control method is the inertial angle value. The actual inertial angle value at the time of robot movement is obtained from calculation of *complementary filter*. The *complementary filter* combines the readings obtained from *accelerometer* and *gyroscope* as well, by performing *a low pass filter* on the *accelerometer* to reduce noise and *a high pass filter* on calculation results of *gyroscope integration* so that ideal inertial angle readings will be obtained. A block diagram of control methods employed in the *Two-Wheeled Self Balancing Robot* can be seen in Figure 3.

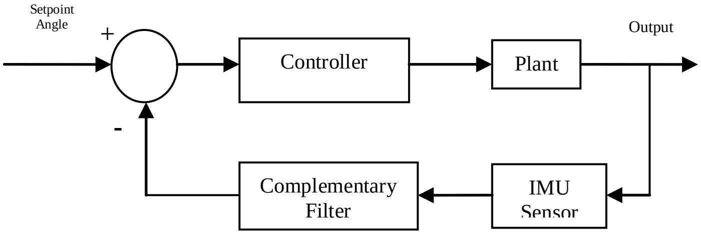


Fig. 3. Block Diagram of Control Method of Two-Wheeled Self-Balancing Robot

The block diagram of *complementary filter* can be seen in Figure 4 below.

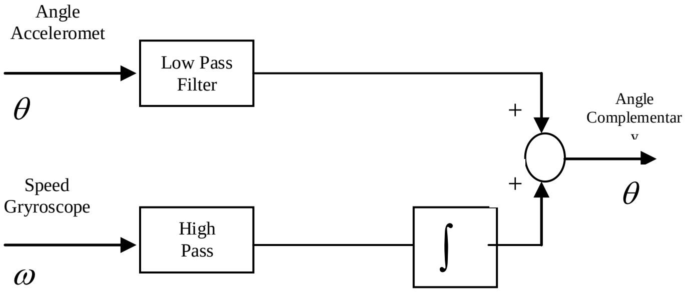


Fig. 4. Block Diagram of Complementary Filters

D. Motion System Design

The motion system on Two-Wheeled Self-Balancing Robot depends on inertial angle which is read by IMU sensor. The IMU itself consists of *accelerometer* and *gyroscope* which each has drawbacks. The *accelerometer* has noise and *gyroscope* always experiences drift errors (shift in starting point) in long term so that both sensors cannot be used separately. To get good angle reading, both sensors are combined by using a filter so-called a *complementary filter*.

Based on inertial angle readings obtained from the *complementary filter*, the microcontroller is programmed to make decisions in setting the direction and speed of DC motor on robot. The robot will move forward if it tends to tilt forward and conversely the robot will move backward if it tends to tilt backward. The robot will also move faster if the inertial angle is read greater than the setpoint or if it experiences faster error change.

IV. Result

A. Testing of Tilt Angle Readings against the Complementary Filters

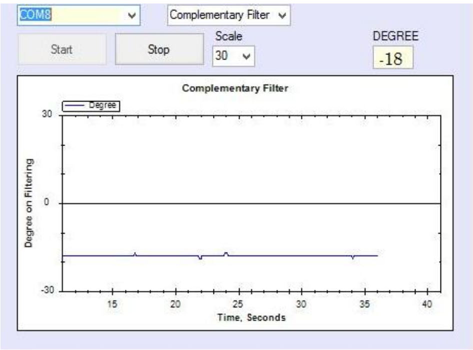

COM8 · Complementary Filter · Scale 30 · DEGREE -18

Fig. 5. Testing of Tilt Angle Readings of Complementary Filter with value of a = 0.99

The time required to update the value of tilt angle of *complementary filter* with a = 0.99 is 0.99 s. From the overall testing on data of *complementary filter*, it appears that the tilt angle of *complementary filter* will be even better (do not have noise) if the value of filter coefficient *a* is closer to 1 (one). But the bigger the value of filter coefficient *a* employed in the *complementary filter* operation is, the longer the update time of *complementary filter's* output value will be.

B. Testing of PD control methods against robot stability in Maintaining the Upright Standing Balance Position

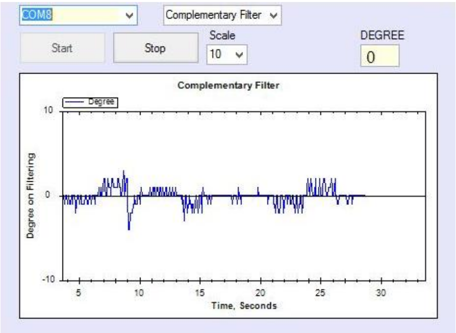

COM8 · Complementary Filter · Scale 10 · DEGREE 0

Fig. 6. Testing of PD control with Kp = 14 and Kd = 0.4

In testing the PD control with Kp = 14 and Kd = 0.4, the robots response is considerably better. The robot can maintain its upright position for 29 seconds without any interruption.

C. Testing of PI control method against Robot stability in Maintaining the Upright Standing Balance Position

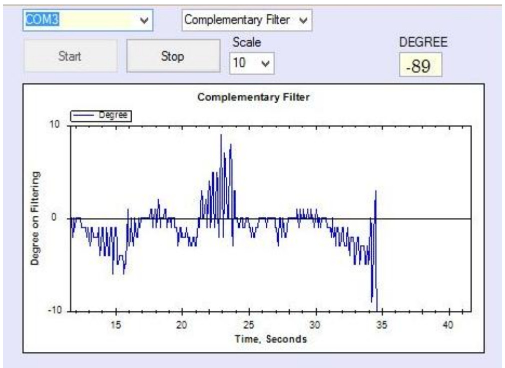

COM3 · Complementary Filter · Scale 10 · DEGREE -89

Fig. 7. Testing of PI control with Kp = 15 and Ki = 0.005

The robot response with Kp = 15 and Ki = 0.005 is slightly better when compared to the robot response in the previous testing. The robot can stand for about 34 seconds, but still experiences high oscillations.

D. Testing of PID controller against Robot Stability in Maintaining the Upright Standing Balance Position

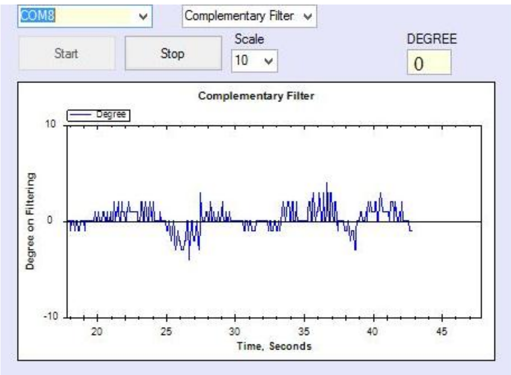
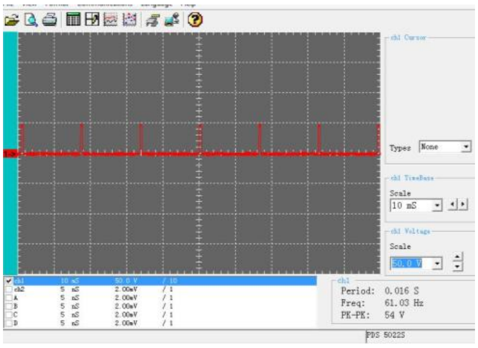

COM8 · Complementary Filter · Scale 10 · DEGREE 0

Fig. 8. Testing of PID control with Kp = 14, Ki = 0005 and Kd = 0.1

Response obtained by using the value of Kp = 14, Ki = 0005 and Kd = 0.1 is the best response when compared with the previous testing. The robot is able to maintain the upright standing position for about 43 seconds without receiving interference. The robots still oscillates with the angle range between -3^{o} and +3^{o}.

E. Testing of PWM Transmission to Motor Driver Based on readings of IMU Sensor

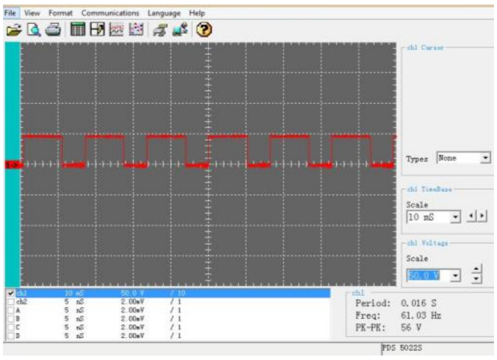


Types: None
Scale: 10 mS
Scale: 50.0 V

```
☑ ch1   10 mS   50.0 V   / 10
□ ch2   5  nS   2.00mV   / 1
□ A     5  nS   2.00mV   / 1
□ B     5  nS   2.00mV   / 1
□ C     5  nS   2.00mV   / 1
□ D     5  nS   2.00mV   / 1
```

```
ch1
Period:  0.016 S
Freq:    61.03 Hz
PK-PK:   54 V
```

Fig. 9. PWM Testing when error value = 0

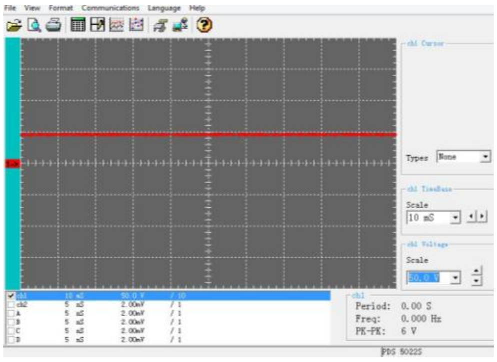


Types: None
Scale: 10 mS
Scale: 50.0 V

```
☑ ch1   10 mS   50.0 V   / 10
□ ch2   5  nS   2.00mV   / 1
□ A     5  nS   2.00mV   / 1
□ B     5  nS   2.00mV   / 1
□ C     5  nS   2.00mV   / 1
□ D     5  nS   2.00mV   / 1
```

```
ch1
Period:  0.016 S
Freq:    61.03 Hz
PK-PK:   56 V
```

Fig. 10. PWM Testing when error value = 5



Types: None
Scale: 10 mS
Scale: 50.0 V

```
☑ ch1   10 mS   50.0 V   / 10
□ ch2   5  nS   2.00mV   / 1
□ A     5  nS   2.00mV   / 1
□ B     5  nS   2.00mV   / 1
□ C     5  nS   2.00mV   / 1
□ D     5  nS   2.00mV   / 1
```

```
ch1
Period:  0.00 S
Freq:    0.000 Hz
PK-PK:   6 V
```

Fig. 11. PWM Testing when error value = 10

Based on the results obtained on the tests conducted on the PWM (Pulse Width Modulation) signal, it was found that the number of pulses sent by the microcontroller is directly proportional to the readable error value. Based the oscilloscope readings, the PWM frequency transmitted by the microcontroller is 61.03 Hz.

V. Conclusion

Based on the overall design and testing of two-wheeled self-balancing robot, it can be concluded that the PID controller is the best controller to be applied for the two-wheeled self-balancing robot. The optimal response of the robot is obtained with the gain parameters of Kp = 14, Ki = 0.005, and Kd = 0.1. With *proportional*, *integral* and *derivative* constant parameter values which are obtained through *trial* and *error*, the robot can balance the upright position only by using two wheels for 43 seconds.

References

[1] M. Alarfaj and G. Kantor, “Centrifugal Force Compensation of a Two-Wheeled,” *Balancing Robot.*,” 11th International Conference on Control, Automation, Robotics and Vision, Singapore Dec. 7-10, 2010, pp. 2333-2338.

[2] S. J. Lee and S. Jung, “Experimental Studies of an Object Handling Task by Force Control between Two Balancing Robots,” 11th International Conference on Control, Automation and Systems, Gyeonggi-do, Korea, Oct. 26-29, 2011, pp. 197-201.

[3] J. Ryu, F. Ruggiero, and K. M. Lynch, “Control of Nonprehensile Rolling Manipulation: Balancing a disk on a disk,” IEEE International Conference on Control, Automation and Systems.

[4] K. Peng, X. Ruan, and G. Zuo,” Dynamic Model and Balancing Control for Two-Wheeled Self-Balancing Mobile Robot on the Slopes,” Proceedings of the 10th World Congress on Intelligent Control and Automation July 6-8, 2012, Beijing, China, pp. 3681-3685.

[5] S. Kim, J. Seo, and S. Kwon,” Development of a Two-Wheeled Mobile Tilting & Balancing (MTB) Robot,” 11th International Conference on Control, Automation and Systems, Oct. 26-29, 2011 in KINTEX, Gyeonggi-do, Korea, pp. 1-6.

[6] C. Yong and C. F. Kwong,” Wireless Controlled Two Wheel Balancing Robot,” International Journal of Network and Mobile Technologies, vol 2, Issues 2, ISSN : 2229-9114 Electronic Version, May 2011, pp. 88-109.

[7] W. T. Higgins, Jr.,”*A Comparison of Complementary and Kalman Filter*,” IEEE Thans. Aerospace and Electronic Systems,Vol. AES-li, no. 3, May 1975, pp. 321-325.

[8] MPU-6000 and MPU-6050 Product Specification Revision 3.3, <http://www.invensense.com/mems/gyro/documents/PS-MPU-6000A.pdf>

[9] M. A. Johnson, and M. H. Moradi, “*PID Control: New Identification and Design Methods,”* Springer, 2005.

[10] K. Ogata,”*Modern Control Engineering 3rd Edition,”*Prentice-Hall, 1997.