Real-Time PID Control of Wireless Two-Wheeled Balancing Lego EV3 Robot

M.R Hao, N.M Abdul Ghani, M.S Abdul Wahid
Faculty of Electrical & Electronic Engineering, Universiti Malaysia Pahang 26600 Pekan Pahang
normaniha@ump.edu.my

***Abstract*—Research on balancing two-wheeled robot has gained interest among researchers due to its highly nonlinear characteristic. The objective of this project is to control the stability of a two-wheeled EV3 Lego robot and maintain in the upright position while performing linear motion as well steering right/left and moving up/down a ramp. In this project, a two-wheeled balancing Lego EV3 robot is modelled in a state space and controlled by a PID controller. The robot is controlled in real time via Matlab/Simulink interface using Graphical User Interface (GUI) and the robot performance can be observed wirelessly at the same time using Wi-Fi connection between the robot and MATLAB. The two-wheeled EV3 Lego robot is able to stay in the upright position while performing steering and going up/down the ramp. The analysis of the system plant has been made in terms of overshoot, settling time of tilt angle stabilization using simulation approach and successfully controlled using real two-wheeled EV3 hardware.**

***Index Terms*—Real-time PID controller; Two-wheeled Lego EV3 Robot; Upright Stability.**

I. INTRODUCTION

In the past few years, the two-wheeled inverted pendulum has gained much attention from researchers, educators as well as robotic lab around the world. The two-wheeled inverted pendulum is a self-balancing vehicle with two wheels attached on both sides of chassis where it is obviously non-holonomic or unstable. This system has been widely used as a challenging benchmark to test the performance of a wide range of control strategies.

It is very expensive and costly to build new experimental devices to verify the effectiveness of a control theory. Thus, to overcome this problem, the Lego EV3 was used because of its high degree of freedom at low cost. PID controller is one of the linear and conventional type controller where it related to the mathematical calculation of error. This controller is very synonym with the tuning method, as the suitable parameters must be found to improve the robustness and performance of the system. In this project, a two-wheeled EV3 Lego robot was assembled and the mathematical modeling of its dynamic model were derived by using the Euler Lagrange's or Lagrangian 's method.

Limited works were succeeded in balancing the two-wheeled EV3 Lego Robot by using the PID controller. There are a few linear controllers that have been used by previous researchers as their control strategy for two-wheeled inverted pendulums (TWIP). Most of these researchers aimed to find and analyse the designated controller according to its ability to stabilise the overall performance of controller on the TWIP. A.M Almeshal et. al. used PD-PID controllers for their simulation and found that this controller able to stabilise the TWIP successfully, but further studies have to make to include the intelligent control scheme [1]. Jung et. al. also used the same strategy and the result is the TWIP was able to stabilize, even with the external disturbance, with the addition of Kalman's filter [2]. S.W.Nawawi et. al. have reported that pole placement also was able to deliver the desired outcome on TWIP [3]. Next, Airton et. al. presented their simulation of TWIP as an assistive robotic walker using LQR control approach and found that this controller is not suitable. They suggested a better controller still needed even though LQR was able to stabilise the TWIP [4].

T.Chen et. al. used the PID to propose its effectiveness over two-wheeled vehicle by implementing motion control system to stabilize the robot [5]. The tuning parameters of the controller were automatically done by neural-network. Based on their finding, the robot was able to stabilise, overcame the disturbance and followed the desired command motion. Two linear controllers also can be combined to control a two-wheeled robot. L.Sun et. al. reported that the combination of these two controllers were able to overcome the effect of the constraint to linearizing the system [6]. The hardware validation has shown that the robot was able to stabilise at a shorter time and the validity of the controller design were also verified. From the paper by A.N. Kasruddin et. al., they presented a comparative assessment between fuzzy logic controller and PID controller. The fuzzy logic controller is better in terms of performance of the system, but both of the controllers are actually capable to stabilise the two-wheeled robot [7].

From the research papers reviewed, it can be concluded that the PID controller is the most efficient yet simplest type of controller when compares to other linear controllers. It is well known that the linear controller is widely used in process control industry due to its efficiency in terms of cost and implementation. It also can be proven that Lego Mindstorms can be used as validation device for the control theory and can be used as an experimental prototype before the real application.

II. LEGO EV3 MINDSTORM SYSTEM

Various non-linear controllers have been done by researchers in order to stabilize the inverted pendulum system [8-9]. These controllers have been introduced to many platforms such as Arduino and LEGO Mindstorms kit. LEGO Mindstorms kit has 2 different model which are NXT and EV3. EV3 is the latest model and few types of research have been done to this model compared with the NXT which has been done many research on it. In this project, EV3 will be used as a platform to analyse the performance of fuzzy logic control in stabilizing the EV3 LEGO robot.

Few research papers have been using the Lego robot as a validation approach for the control theory. T. Kanada et. al. was using H_2 control for their control verification through simulation, experimental and result comparison [8]. They want to prove that Lego Mindstorms has potential for verification of control theory. Based on the result, it is shown that the Lego Mindstorms can validate the control theory since the feature and the differences between simulations are identical in the experimental results. Sheelu T.M et. al. in their paper aimed to validate the robustness of H_2 compare to the LQR controller in terms of performance [9]. By designing and comparing the two controllers, they found that H_2 are more effective than the LQR controller.

Figure 1: Two-wheeled balancing EV3 LEGO robot.

III. PID CONTROLLER

PID controller will be used to maintain the stability of the TW EV3 Lego robot to achieve zero-degree upright position. This controller is well known and popular in the industrial process control due to its simplicity. It has become the basis for many advances control algorithm and strategies. PID implementation in analogue circuits has made it possible to control systems such as domestic heaters to chemical process plants. As microprocessor and microcontrollers became popular in control engineering, PID controller has become a popular embedded software implementation.

PID stands for proportional, integral and derivative and the equation is shown in Equation (1). The PID controller is designed to eliminate the need for continuous operator attention. Cruise control in a car and a plant's thermostat are common examples of how controllers are used to automatically adjust some variable to hold the measurement.

$$u_{PID} = K_p + \frac{K_I}{s} + \frac{K_D}{s} \quad (1)$$

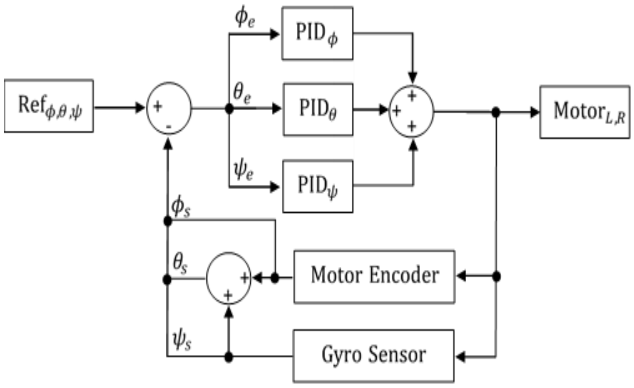

Figure 2: The PID controller diagram.

The set-point or reference point is the desired measurement set by the user. The error is defined as the difference between set-point and measurement. The variable being adjusted is called the manipulated variable which usually is equal to the output of the controller. The output of PID controllers will change in response to a change in measurement or set-point.

IV. RESULTS

A. Simulation Analysis

The selection of tuning method is very crucial in the PID controller because each process system has its own criteria. This tuning method is needed in order to find the suitable value of each gain of the K_P, K_I and K_D. For this study, three methods have been proposed and the evaluation were made between these methods.

From Figure 3, it can be seen that the system takes less than 2 seconds to settle and continuously stable. Referring to the control input signal (U), the voltage that is supplied to the motors are started from -20V to 20V. This is due to the saturation block used in the Simulink block diagram function that limits the input voltage. The voltage also decreased across time because as the plant is getting stable, the motor has less movement thus less voltage is needed.

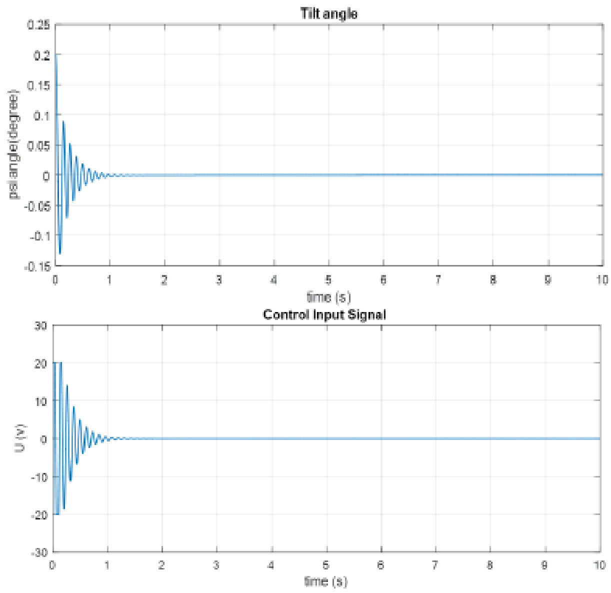

Figure 3: The tilt angle and control input signal (U) of the system using PID tuner tuning method.

There are some standard procedures that have to be taken into account to make the PID controller efficient. Firstly, the strong understanding of the function of each parameter of PID is important. Because in order to get the desired output, each of the gain plays an important role in the output. Secondly, the table of trial and error have to be constructed so that the same value is not entered repeatedly. This can reduce the time to find the value of the gain.

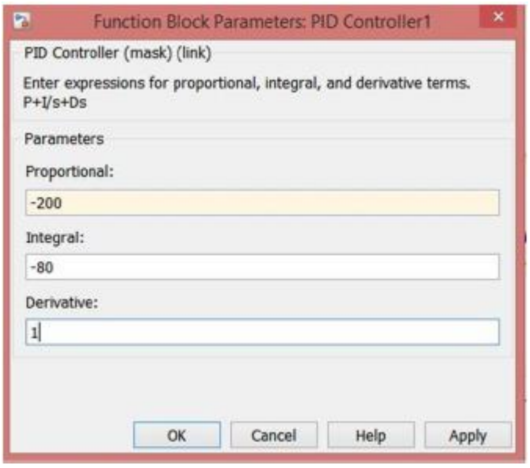

Figure 4: The PID gain parameter box

Figure 5 shows that the system is unable to achieve the desired result. Even though it converges and reaching zero but there is some ripple at the zero. The control input signal also shows similar pattern. This is due to the movement of the robot. It might balance but we can predict the robot will be shaking.

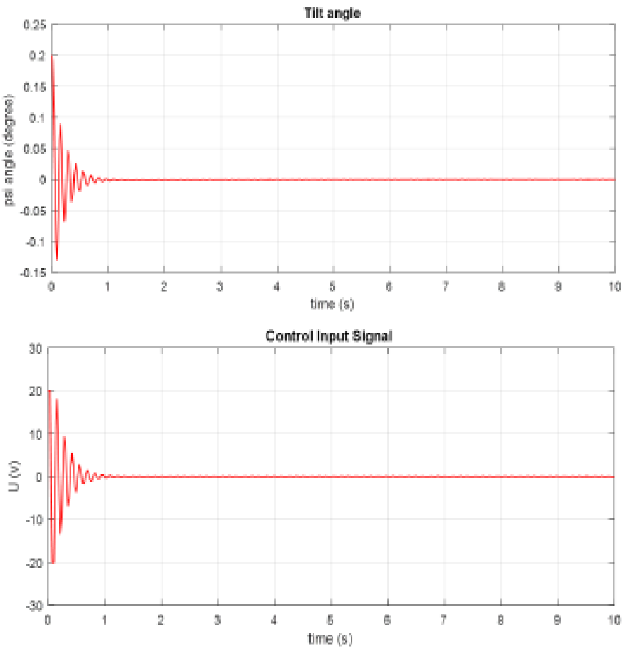

Figure 5: The tilt angle and control input signal (U) of the system using heuristic tuning method

B. P Controller

The system is able to reach and converge to zero but there is a small oscillation at the zero and continuously oscillates indefinitely. It shows that the robot is able to stand at the upright position but the aim to achieve zero degree is unable to satisfy.

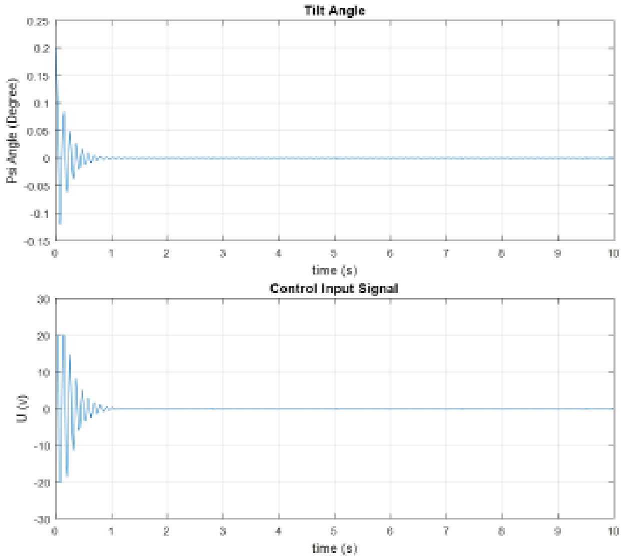

Figure 6: The tilt angle and control input signal (U) of P controller

C. PI Controller

Both tilt angle and control input signal as shown in Figure 7 are able to converge but it takes a longer time to stabilize and reach zero degree or minimum voltage. This can prove that for this system, the PI controller is not suitable. A better controller should be implemented for a better result.

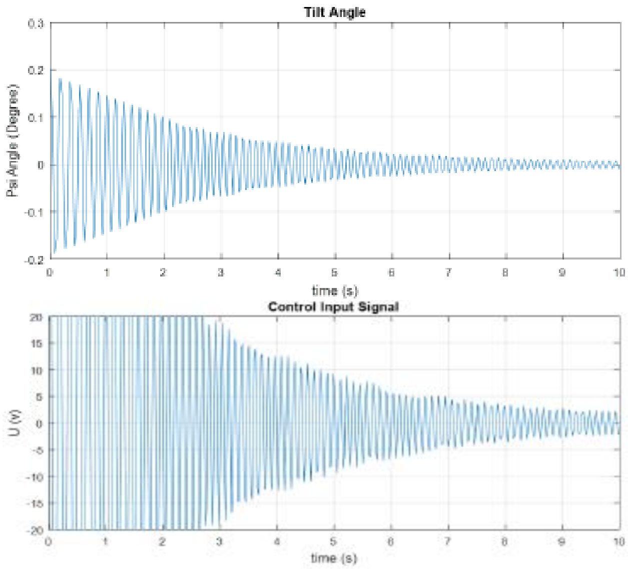

Figure 7: The tilt angle and control input signal (U) of the system using PID trial and error tuning method.

As seen in Figure 8, obviously the PI controller is not able to stabilize the system the P controller can stabilize the system by converging and achieved close to zero but is not able to maintain the stability because the oscillation exists at the zero. PID is still the best controller compares to the rest in terms of stability and performance.

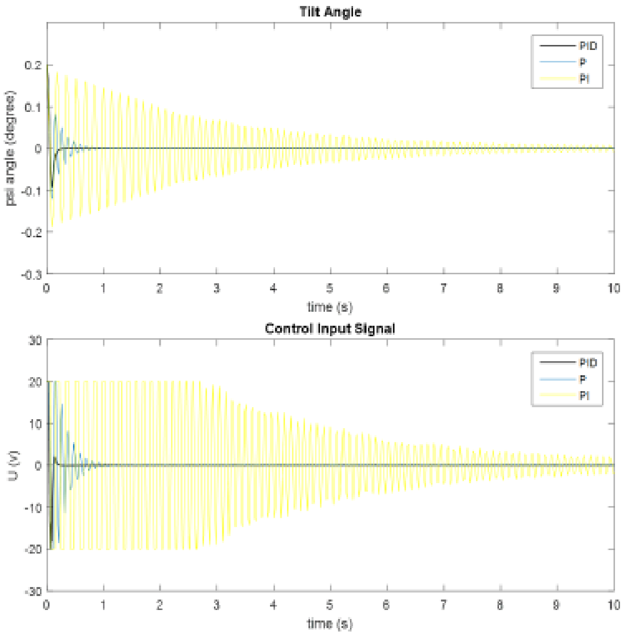

Figure 8: The tilt angle and control input signal (U) of P, PI and PID controller.

D. *PID Gains Analysis*

The PID gain analysis has been conducted to see the effect of each parameter and to prove whether the results are similar to the standard table of the PID gain parameter or not. This analysis is made by increasing 50% of each gain and the rest remains the same with the original value. The actual result stated here were taken from the data obtained from Ziegler-Nichols method previously. In this analysis, the observation only been made to the tilt angle and not the control input signal.

E. *Increase K_p*

Figure 9, 10 and 11 show that by increasing the gain of the proportional, the overshoot of the system can be reduced as well as the steady state error.

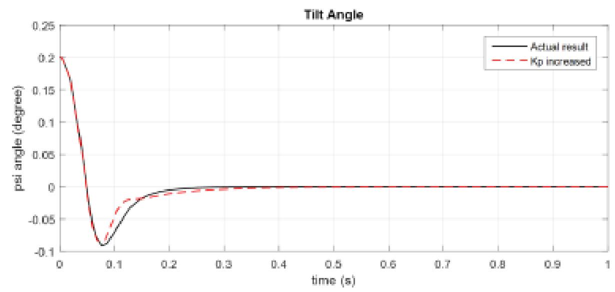

Figure 9: Tilt angle after increase the gain P

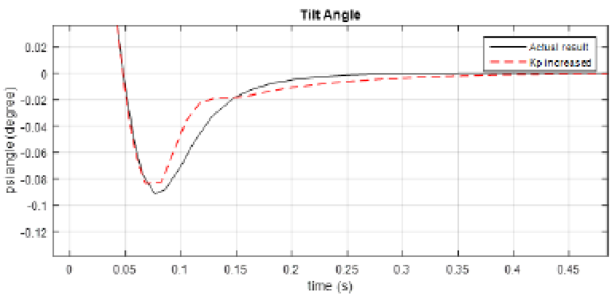

Figure 10: Overshoot

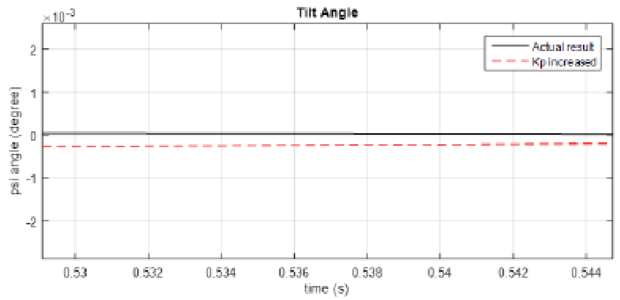

Figure 11: Steady-state error.

F. *Increased K_I*

Figure 12, 13 and 14 show that by increasing the gain of the proportional, the overshoot of the system increase as well as the steady state error.

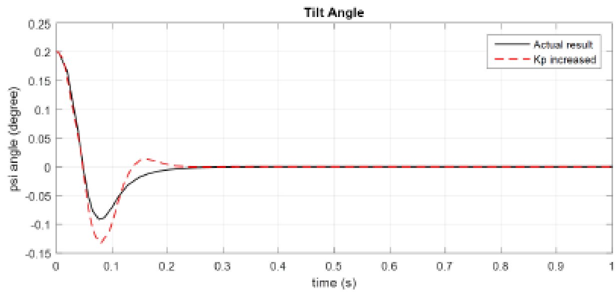

Figure 12: Tilt angle for increasing the Integral gain, K_I.

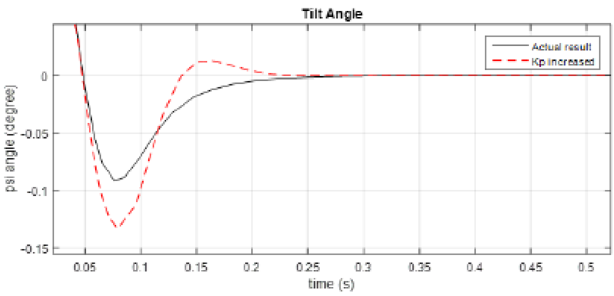

Figure 13: Overshoot for increasing the Integral gain, K_I.

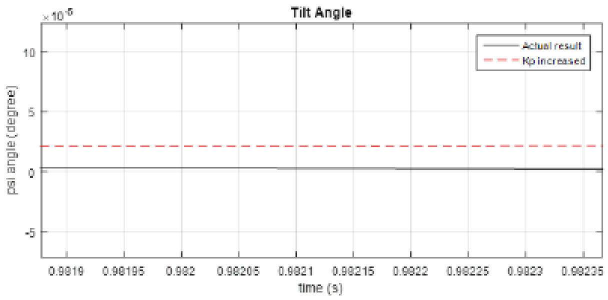

Figure 14: Steady-state error for increasing the Integral gain, K_I.

G. *Increase K_D*

Figure 15, 16 and 17 show that by increasing the gain of the proportional, the overshoot of the system can be reduced but the steady state error increased.

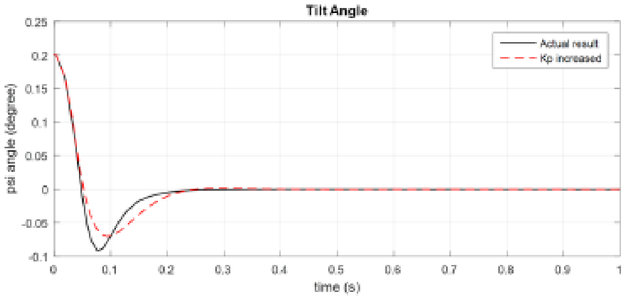

Figure 15: Tilt angle for increasing the Derivative gain, K_D.

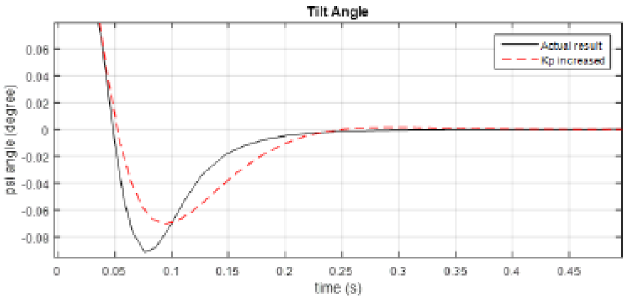

Figure 16: Overshoot for increasing the Derivative gain, K_D.

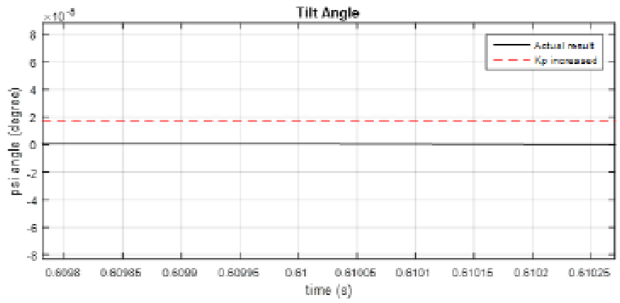

Figure 17: Steady-state error for increasing the Derivative gain, K_D.

The result obtained are slightly different from the reference table, may be due to the different nature of the system or some mathematical reason. The result obtained are summarized and tabulated in Table 1.

Table 1
Summarization of Result Obtained.

Parameter	Rise Time	Overshoot	Settling Time	Steady State Error
K_p	None	Decrease	Increase	Decrease
K_I	None	Increase	None	Increase
K_D	None	Decrease	Small Change	Increase

From the above statement, it can be proved that the result obtained is still accepted since different system might have different nature of control mechanism. Then, the controller is implemented in a real-time hardware and the responses are observed.

H. *Hardware Implementation*

The hardware control block diagram is shown in Figure 18. It consists of two parts which are the Controller and the EV3. The PID controller algorithm is contained in the controller block where the desire input value is used as a reference and the output is produced to compensate with the feedback state value of the robot. Meanwhile, the EV3 block contains the corresponding I/O port of gyro-sensor, left motor and right motor of the EV3 robot. On the other hand, Figure 19 is the input setting of the desired value of speed and turn angle. The user can control the robot movements by varying the controller knob and the robot will move according to the input setting. For example, if a user sets the turn angle to 90°, then the robot will turn to 90° direction.

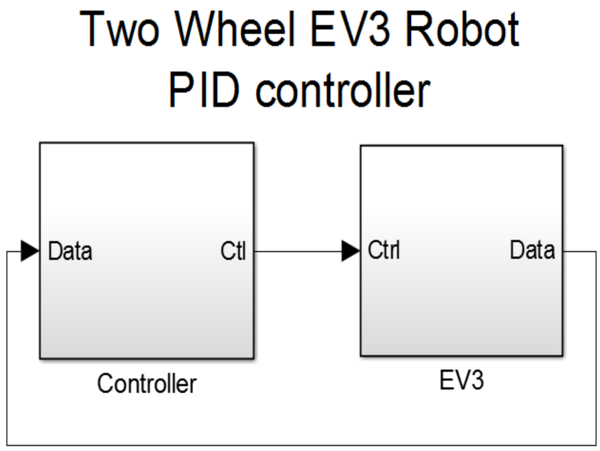

Figure 18: Hardware control block diagram.

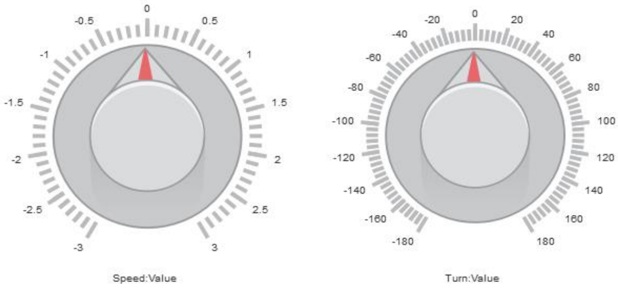

Figure 19: Hardware control knob for setting desire input value.

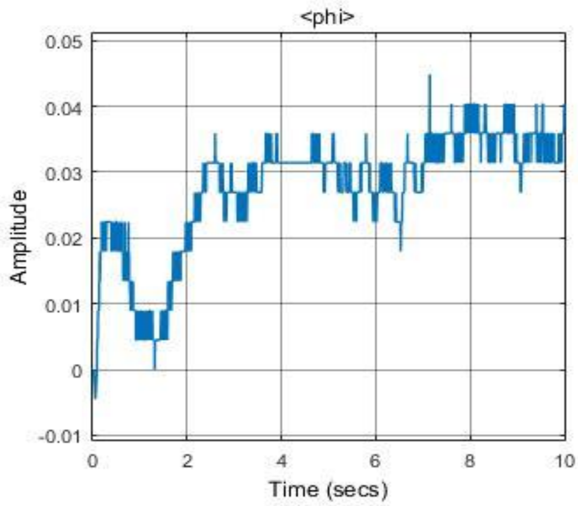

Figure 20: Hardware real-time tilt angle response.

Figure 20 shows the real-time hardware response of tilt angle for 10 seconds run time. The result shows the robot is able to balance itself within $\pm 0.04°$ which is very close to 0^0 origin point in the upright position. Therefore, we concluded that, the controller has potential to implement this application. The robot is able to maintain upright while performing all tasks: on a flat surface, uneven surface, ramp (60 degrees), turning left and right, moving forward and backwards.

V. Conclusion

PID controller is able to stabilize the robot with less than 0.4 seconds. This proves the PID controller's robustness and controllability. The PID controller is also proven to be a much better controller than other controllers from the PID family. Each gain parameter of P, I and D gives a significant impact on the performance of the system. The controller is also tested in real hardware of two-wheeled EV3 LEGO robot and showed stable performance in term of the tilt angle. The system is able to stay in the upright position while steering to left, right and going up/down the ramp.

Acknowledgment

This work is supported by Universiti Malaysia Pahang under the internal grant, RDU 150361.

References

[1] A. M. Almeshal, K. M. Goher, and M. O. Tokhi, "Dynamic modelling and stabilization of a new configuration of two-wheeled machines," *Rob. Auton. Syst.*, vol. 61, no. 5, pp. 443–472, 2013.

[2] J.-S. Ha and J. Lee, "Position control of mobile two-wheeled inverted pendulum robot by sliding mode control," *2012 12th Int. Conf. Control. Autom. Syst.*, pp. 715–719, 2012.

[3] S. W. Nawawi, M. N. Ahmad, and J. H. S. Osman, "Development of a Two-Wheeled Inverted Pendulum Mobile Robot," no. December, pp. 1–5, 2007.

[4] A. R. da Silva and F. Sup, "Design and control of a two-wheeled robotic walker for balance enhancement.," *IEEE Int. Conf. Rehabil. Robot.*, vol. 2013, p. 6650448, 2013.

[5] T.-J. Ren, T.-C. Chen, and C.-J. Chen, "Motion control for a two-wheeled vehicle using a self-tuning PID controller," *Control Eng. Pract.*, vol. 16, no. 3, pp. 365–375, Mar. 2008.

[6] L. Sun and J. Gan, "Researching of two-wheeled self-balancing robot base on LQR combined with PID," in *Proceedings - 2010 2nd International Workshop on Intelligent Systems and Applications*, ISA 2010, 2010.

[7] A. N. Kasruddin Nasir, M. A. Ahmad, R. Ghazali, and N. S. Pakheri, "Performance comparison between Fuzzy Logic Controller (FLC) and PID controller for a highly nonlinear two-wheels balancing robot," in *Proceedings - 1st International Conference on Informatics and Computational Intelligence*, ICI 2011, 2011.

[8] T. Kanada, Y. Watanabe, and G. Chen, "Robust H2 control for two-wheeled inverted pendulum using LEGO Mindstorms," *Aust. Control Conf. (AUCC)*, 2011, no. November, pp. 136–141, 2011.

[9] H.-T. Yau, C.-C. Wang, N.-S. Pai, and M.-J. Jang, "Robust Control Method Applied in Self-Balancing Two-Wheeled Robot," *2009 Second Int. Symp. Knowl. Acquis. Model.*, pp. 268–271, 2009.

[10] Control Tutorials for MATLAB and Simulink -," Control Tutorials for MATLAB and Simulink. [Online]. Available at: http://ctms.engin.umich.edu/ctms/in-dex.php?example=introduction. [Accessed:1-May-2016.