International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-5C, May 2019

Tuning PID Controller Parameters to avoid the Effect of Integrator Wind-up by using The Experimental-based Heuristic Approach

Supriadi Supriadi, Agusma Wajiansyah, Achmad Fanany Onnilita Gaffar, Arief Bramanto Wicaksono Putra

Abstract---Dc motor is widely used as actuator in industry and robotics. In this experiment a dc motor is used as an Omni wheel actuator on Omni-directional 3WD robot. In order for the movement of the robot to follow the specified vector quantity, then the speed of the wheels connected in each motor shaft must persist at each set of points that have been determined. For these needs then, every dc motor must be controlled the number of revolutions. One of the problems with implementing a PID controller is the possibility of a wind-up integrator effect that causes the system to be unable to follow the command behavior. The purpose of this study was to obtain PID controller parameters in such a way as to avoid the effects of wind-up integrators and achieve the best system response performance using an experimental based heuristic approach. The results of this study show that the wheel spin is able to follow various reference speeds with settling time and steady state error at about 1.1s and 0.648%.

Keywords---PID, speed control, observer-based, cordless DC motor

I. INTRODUCTION

The use of DC Motor (DCM) as an actuator in various industrial applications is well-liked. This is because the DCM is more easily controlled than the AC motor. A common application using DCM as an actuator is related to speed and position control. In both applications, DCM will be regulated by controlling the voltage on its armature. A common technique used for this purpose is to feed a Pulse-Width-Modulation (PWM) signal. The speed of a dc motor may change by feeding it the PWM signal generated by the microcontroller [1]. Controlling speed of DCM can be done with closed-loop control through microcontroller [2], where one of controller commonly used is conventional PID controller [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16]. In this study, DCM is used as an Omni steering wheel of Omni-Directional 3WD robot. In order for the movement of the robot to follow the specified vector quantity, then the speed of the wheels spin in each motor shaft must persist at each set-point that has been determined. For that purpose, the number of turns on each DCM shaft must be controlled. There are many techniques and methods to keep the number of turns per DCM shaft persisted in the set-point. In this study is using the classical Proportional-Integral-Derivative (PID) control. The PID controller will calculate the error signal to generate the control signal used to set the PWM signal fed to DCM. An encoder is used to acquire this error signal. The output of the encoder will be compared with the reference. All of these components are connected in closed-loop systems. PID controller, PWM signal generation, and speed acquisition process are applied by using a microcontroller-based embedded system. One of the problems with implementing PID controllers is when the actuator cannot fully respond to command behaviour. In this condition, the integrator part of the PID controller will produce a large error signal during this period. This phenomenon is referred to as the integrator wind-up effect. This condition will result in a very significant overshoot on the system response. If this happens in a long enough duration then the oscillation possibilities in the system response will be very large. The aim of this study is to obtain the best PID control parameters especially to avoid the occurrence of wind-up integrator effect and able to achieve its best system performance. As a reference is the speed response generated from the system. The tuning process is done experimentally by using the heuristic approach which is done in such a way as to obtain the system response as expected.

Manuscript published on 30 May 2019.
* Correspondence Author (s)
Supriadi Supriadi, Department of Information Technology,State Polytechnic of Samarinda, East Kalimantan, Samarinda (E-Mail: supriadi.polnes@gmail.com)
Agusma Wajiansyah, Department of Information Technology,State Polytechnic of Samarinda, East Kalimantan, Samarinda (E-Mail: agusma.wajiansyah@gmail.com)
Achmad Fanany Onnilita Gaffar, Department of Information Technology,State Polytechnic of Samarinda, East Kalimantan, Samarinda (E-Mail: onnygaffar212@gmail.com)
Arief Bramanto Wicaksono Putra, Department of Information Technology,State Polytechnic of Samarinda, East Kalimantan, Samarinda (E-Mail: ariefbram@gmail.com)

II. METHODS

2.1 Speed control of cordless DCM by using PID controller

The Block diagram of speed control of cordless DCM is shown in Figure 1. One of the problems in implementing the PID controller is how to avoid the integrator wind-up effect. When the actuator is saturated there is an integral windup phenomenon on the PID control. The integrator wind-up is the result of the integration of large error signals during these periods when the actuator is unable to fully respond to its command behavior.

The problem is the integration of accumulated error can achieve a very large value during these periods, which produces a very significant overshoot in the system response. This phenomenon is shown in Figure 2.

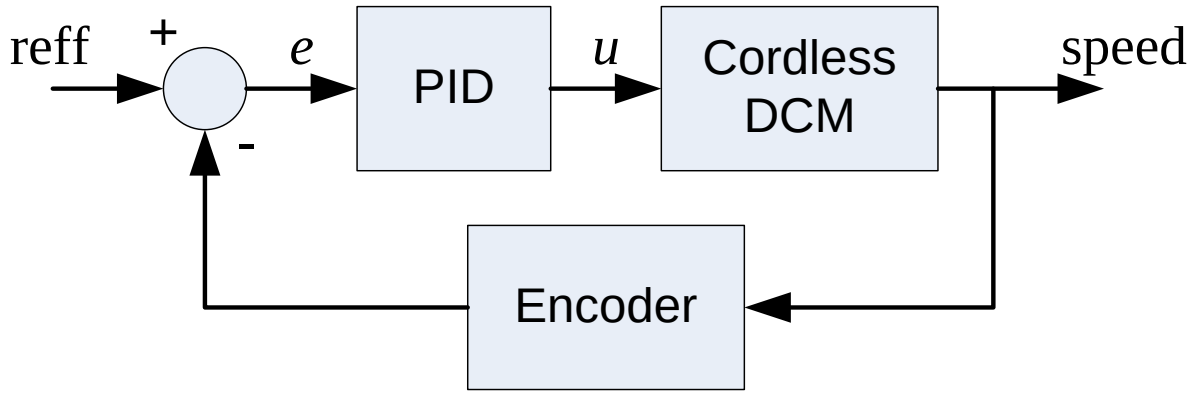

Figure 1: Block diagram of speed control of cordless DCM by using PID controller.

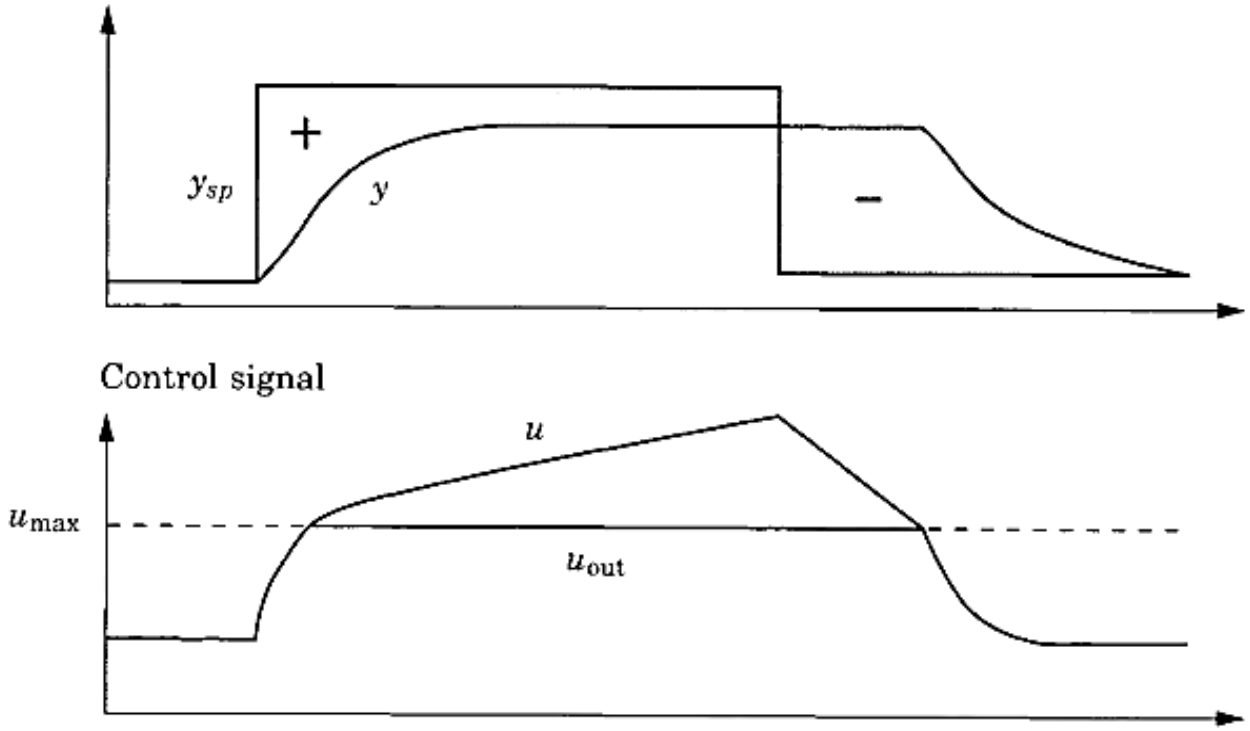

Figure 2: Illustration of Integral Windup effect

Figure 2 shows the control signal (u), the response signal (y), and the set point (y_{sp}) in the case where the control signal becomes saturated. After the first set point change, the control signal (u), rises to the upper limit of u_{max}. This control signal is not large enough to eliminate control errors. Therefore, the integral of control error and the integral part of the control signal increases. Due to the desired control signal increases, then there is a difference between the desired control signal and the actual signal control signal. The control signal starts to decrease because the sign of the control error becomes negative. But since the desired control signal is above the u_{max} limit, the actual control signal will not stop for a while and the response will be delayed. In can be said that the occurrence of integrator wind-up phenomenon will cause the system response cannot follow command behavior. Oscillations may also occur during this situation which may cause the actuator to bump from one end of the range of movement to another. Obviously, the integrator wind-up should be reduced to an acceptable level in a good controller design. In certain situations, where possible the effects of integrator wind-up should be avoided. Illustration of the PID controller with and without integrator wind-up effect avoidance is shown in Figure 3.

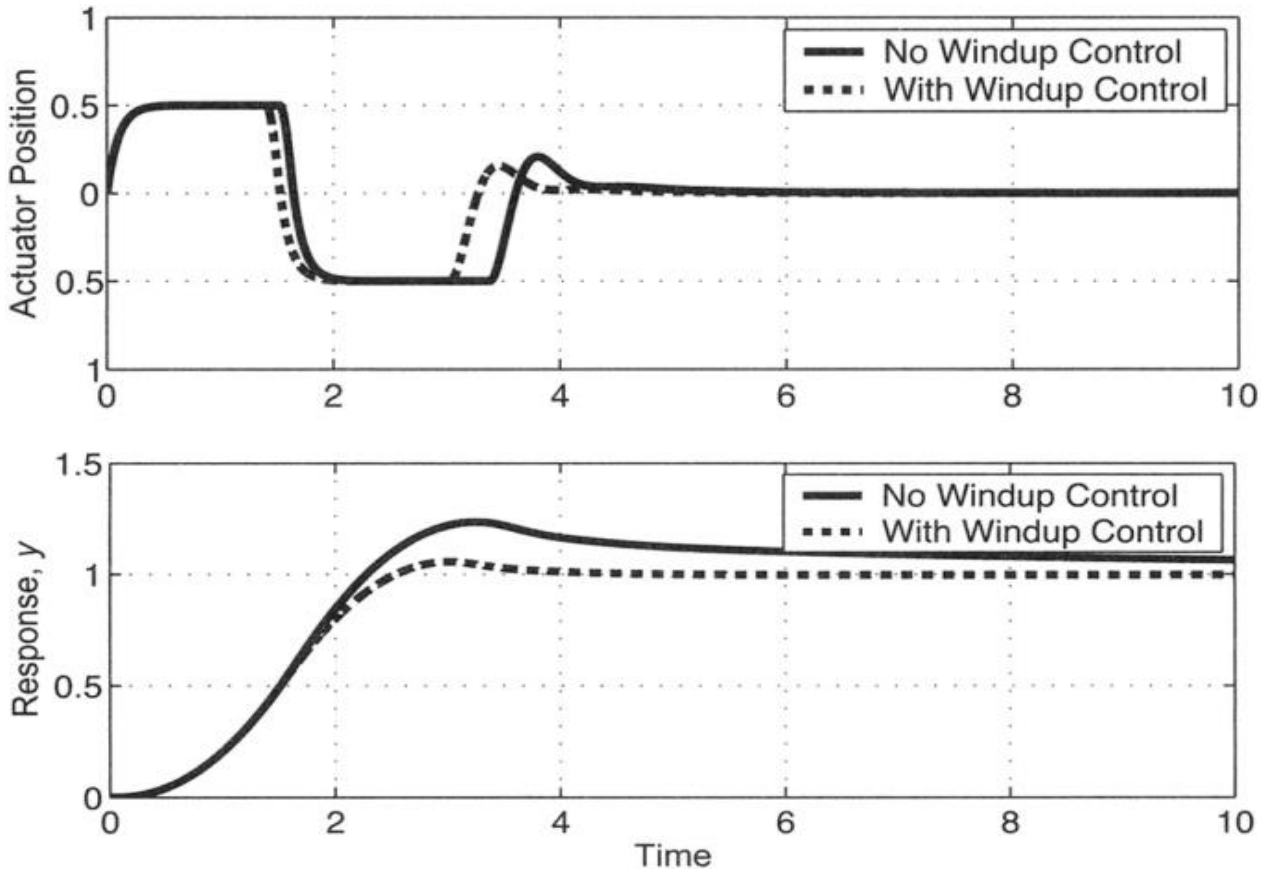

Figure 3: PID controller with and without integrator wind-up effect avoidance

One way to reduce the effect of integrator windup is to turn the integration off when the amplitude (absolute value) of the error signal is above a cut-off level. This result can be achieved by setting the integrator input to zero during periods of large error. In addition to improving controller response in the presence of actuator saturation, this technique can also reduce the overshoot in situations where the system response is linear. PID controller response for a linear plant has a significant overshoot because of the integral term. The overshoot in that example would be reduced substantially by the addition of integrator windup reduction to the controller. The PID controller algorithm with integrator windup avoidance included is expressed by:

$$u = K_p\left(e + K_i\int_0^t qe\,d\tau + K_d\frac{de}{dt}\right) \quad (1)$$

Where, $q = 1\ if\ |e| < E, else\ q = 0$, u is control signal, e is error signal, and K_p, K_i, K_d are proportional constant, integral constant, derivative constant, respectively. Variable E is cut-off level. The parameter q freezes the value of the integrated error during periods of large error. Eq. (1) is mathematically describes how the tuning of PID parameters is done by setting the cut-off level E where the amplitude error becomes "large".

2.2 Experimental-based Heuristic Approach

The Heuristic Approach is a problem-solving approach that uses practical methods that are significant enough to achieve a goal. In other words, heuristics is a rule of thumb where the solution process depends on intuitive or empirical rules [17]. In this study, a heuristic approach is used to derive a combination of PID controller parameters in such a way as to obtain the best system response performance that avoids the occurrence of wind-up integrator effects. Firstly, it is assumed that the integrator wind-up effect is certainly to occur. The cut-off level must be high enough for normal steady-state errors (positive control signal) to be eliminated or avoided. It should also be small enough for negative control signal so that the effects of the integrator windup are reduced to an acceptable level. Selecting a cut-off rate may require some repetitive testing similar to the adjustment made to select the controller's gain. In this case, the determination of cut-off level E is based on observation of the experimental results. The Experimental-based heuristic approach stages used is shown in Figure 4. The updating of the PID controller parameters is performed by C programming which the algorithm is shown in Figure 5.

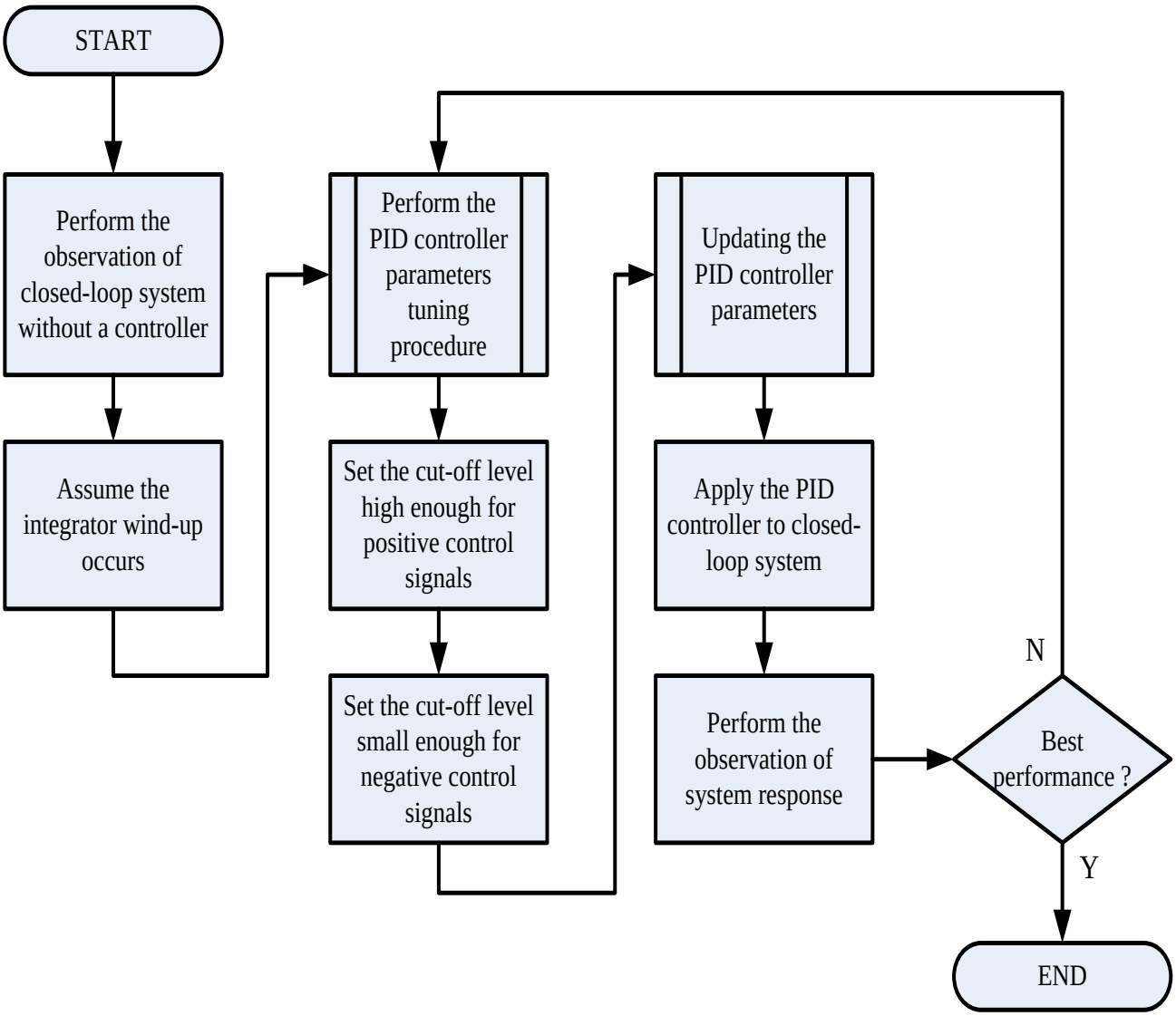

Figure 3: The Experimental-based heuristic approach stages used

The PID controller tuning procedures are as follows:

1. Start by implementing a controller with the algorithm of Eq. (1) and choose a small value of K_p. Set $K_i = K_d = 0$ for now. A small value of K_p will minimize the possibility of excessive overshoot and oscillation.
2. Select an appropriate input signal such as a step input. Perform a test run by driving the controller and plant with that input. The result should be a sluggish response that slowly drives the error in the output toward zero. The response can be unstable. If it is unstable or highly oscillatory, go to step 4.
3. Increase the value of K_p and repeat the test. The speed of the response should increase. If K_p becomes too large, overshoot and oscillation can occur.
4. Increase the value of K_d to reduce any overshoot and oscillation to an acceptable level. If the speed of the response becomes unacceptably slow, try reducing the value of K_d.
5. Continue increasing the value of K_p and adjusting K_d as needed while repeating the test. Watch for the appearance of actuator saturation and reduce K_p if unacceptable saturation occurs.
6. Set K_i to a small value and repeat the test. Observe the time it takes to reduce the steady-state error to an acceptable level.
7. Continue increasing K_i and repeating the test. If K_i becomes too large, the overshoot in response to a step input will become excessive. Select a value of K_i that gives acceptable overshoot while reducing the steady-state error at a sufficient rate. Watch for actuator saturation to occur, which will increase the overshoot in the response.

The algorithm of updating the PID controller parameters is shown in Figure 5.

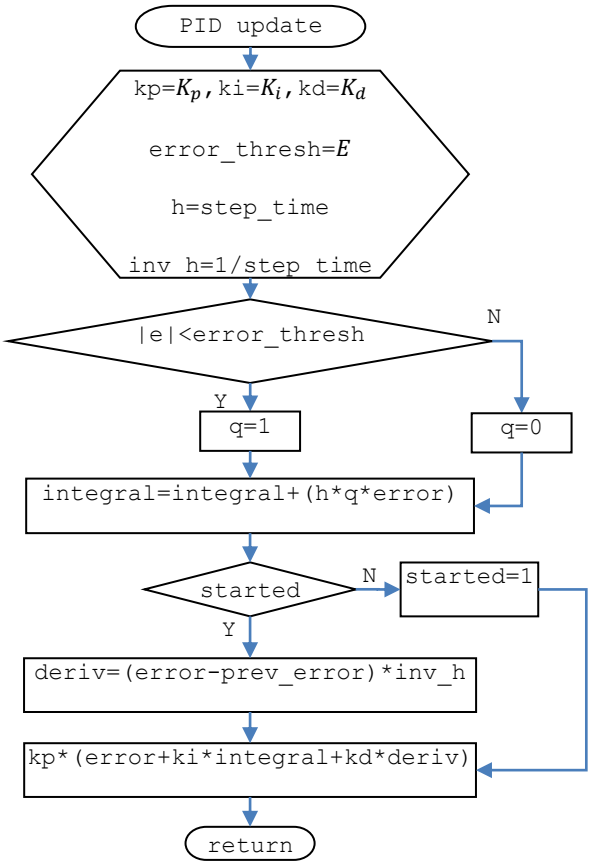

Figure 4: The algorithm of updating the PID controller parameters

To know the number of speed or rotation-per-minute (rpm) on the wheel, which must be known first is cycle-per-revolution (CPR) and periods generated by the encoder. CPR is obtained from the multiplication of gearbox ratio with encoder resolution. If CPR and period are known, then the number of rpm can be calculated by using the following formula:

$$n = \left(\frac{60000000}{\tau}\right)/\text{CPR (rpm)} \qquad (2)$$

Where n, τ, and CPR are rotation per minutes (rpm), period (μs), and cycle-per-revolution, respectively.

III. RESULT AND DISCUSSION

The Omni-wheel speed control prototype is shown in Figure 6. The Omni Wheel 48mm connected with gearbox for DCM cordless 12 VDC Type: 22CL-3501PG. The driver acts as a PWM pulse output signal controller. The controller on this prototype uses the Arduino Uno board. The cord encoder acts as a feet-back signal carrier.

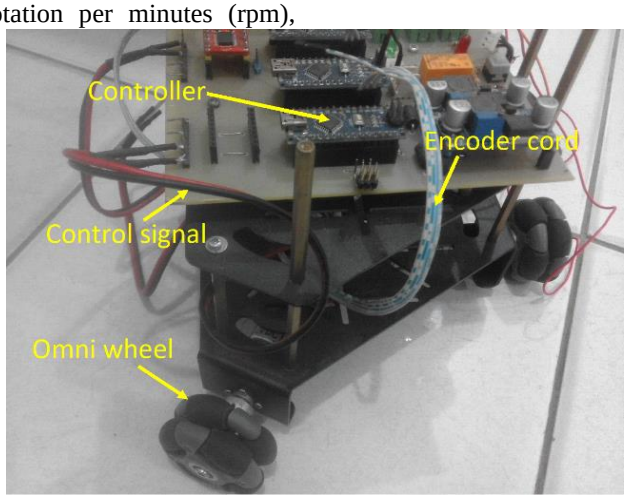

Figure 5: The algorithm of updating the PID controller parameters

Figures 7 - 13 illustrate the tuning process until the best speed control performance is obtained. Figure 7 shows the system closed-loop response without the controller (K_p = 1, K_i = K_d = 0). It can be seen that the response signal (blue color) is still far from the reference signal (red color) provided. Without the controller, the error steady state (e_{ss}) is 85%.

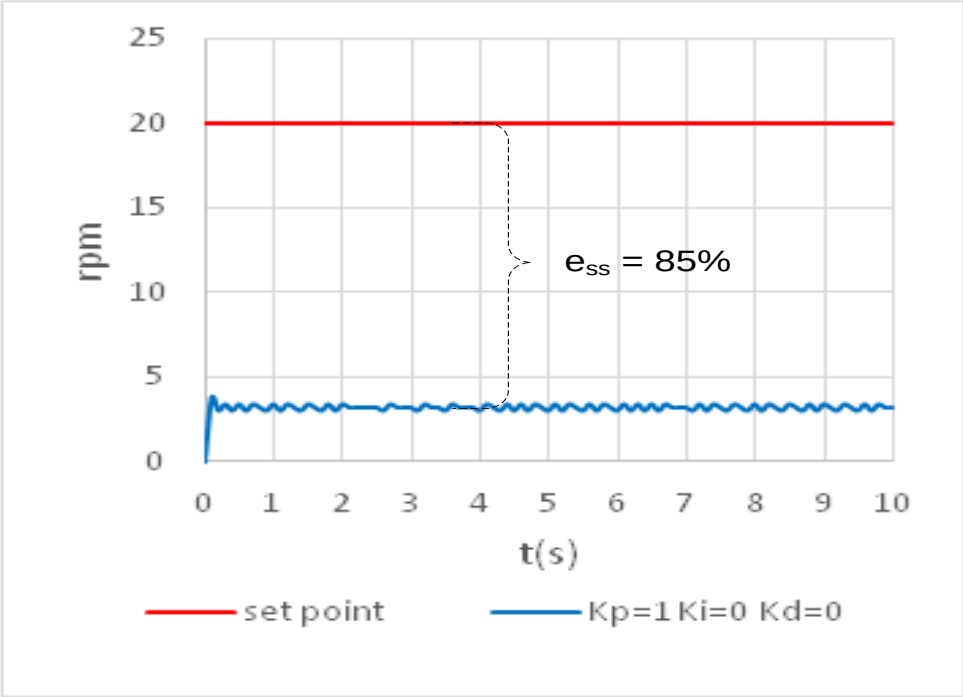

Figure 7. Closed-loop system response without controller

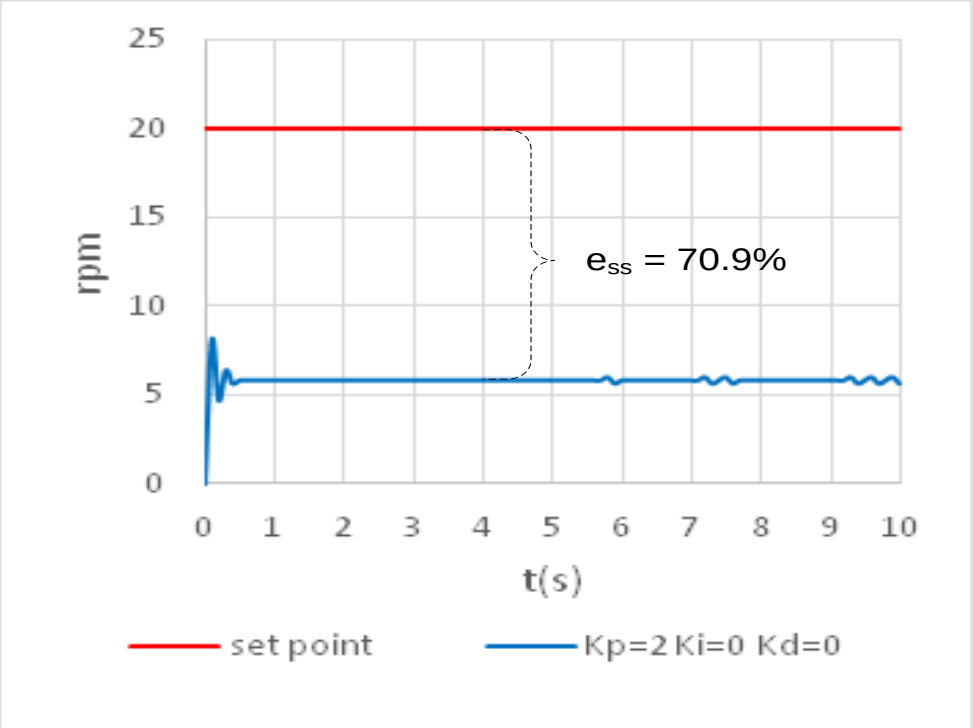

Figure 8. Closed-loop system response with P controller (Kp=2)

To perform the tuning

parameter of PID controller used, the first step is to set the K_p value until the oscillation of response signal. This can be seen in Figure 8. With K_p = 2, the e_{ss} decreases to 70.9% but the oscillation increases. Since the oscillation increases, the K_p value is reduced to 1 as the K_d value increases until the oscillation of the system response decreases. These are shown in Figure 9 and 10.

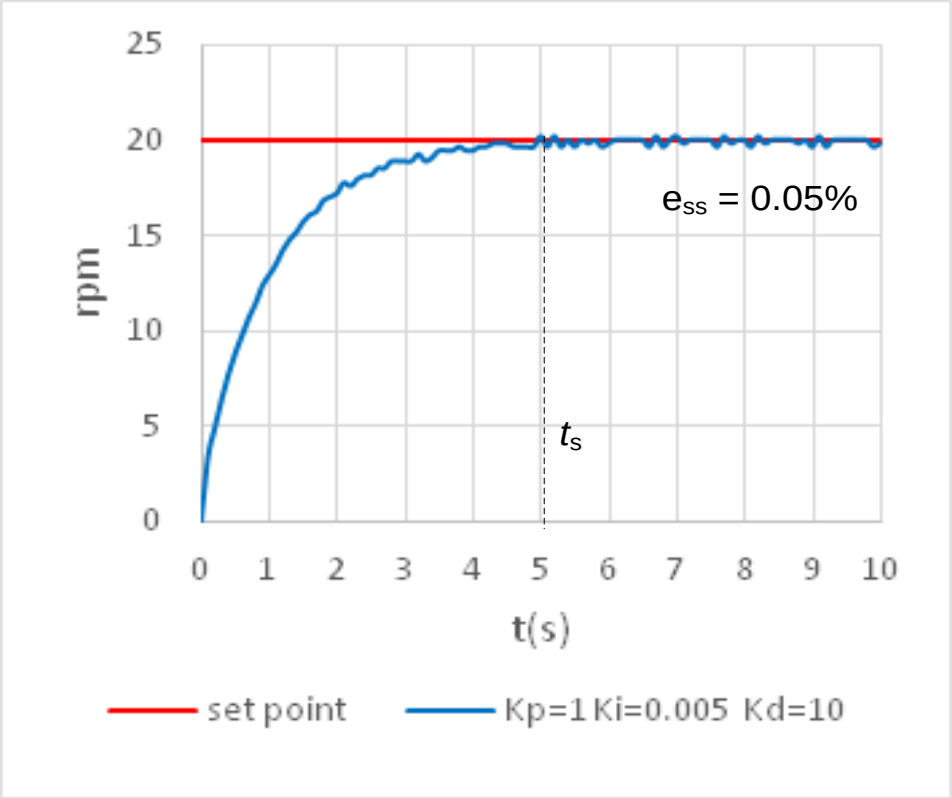

Figure 11. Closed-loop system response with PID controller (Kp =1, Ki =0.005, Kd =10)

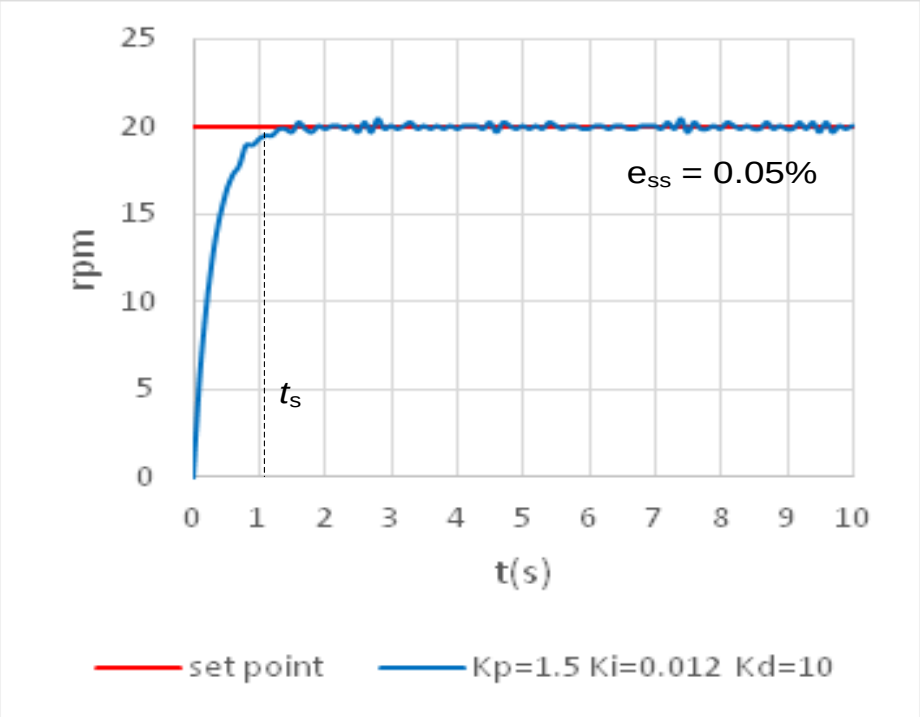

Figure 12. Closed-loop system response with PID controller (Kp =1.5, Ki =0.012, Kd =10)

By using the tolerance band of +/- 2.5% as shown in Figure 13 then the observation obtained the settling time of 1.1s and e_{ss} = 0.648% by K_p = 1.5, K_i = 0.012, K_d = 10 and the reference signal speed of 20 rpm.

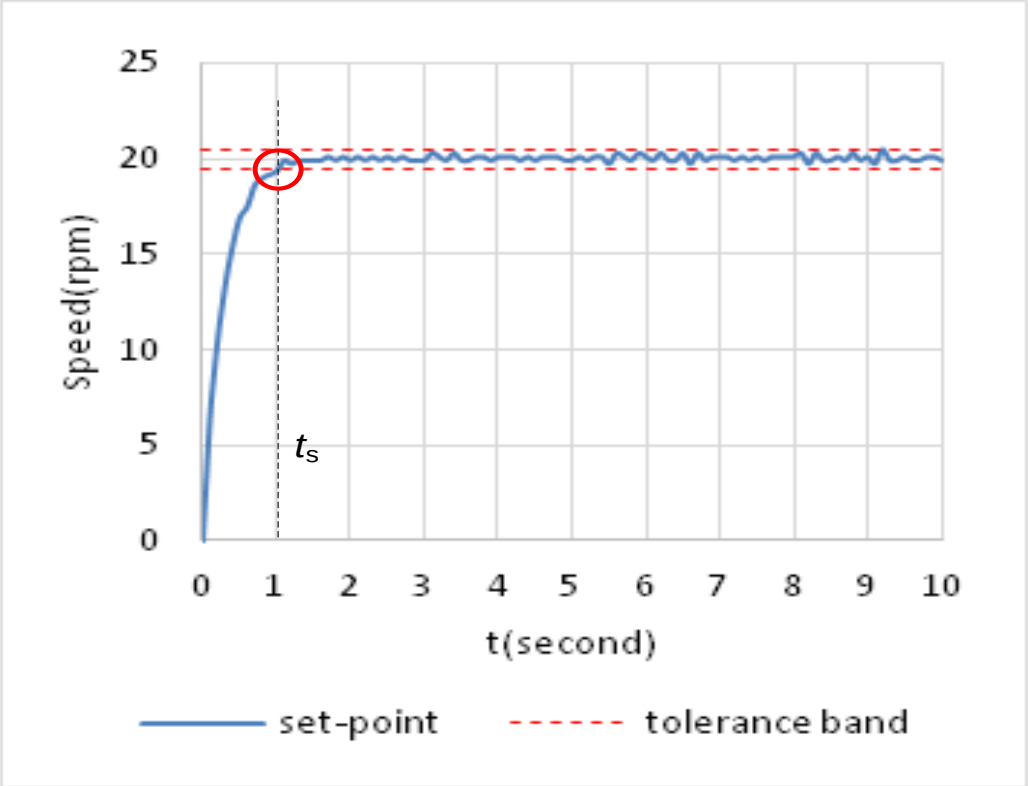

Figure 13. Performance of Omni-wheel speed control system response at 20 rpm reference signal

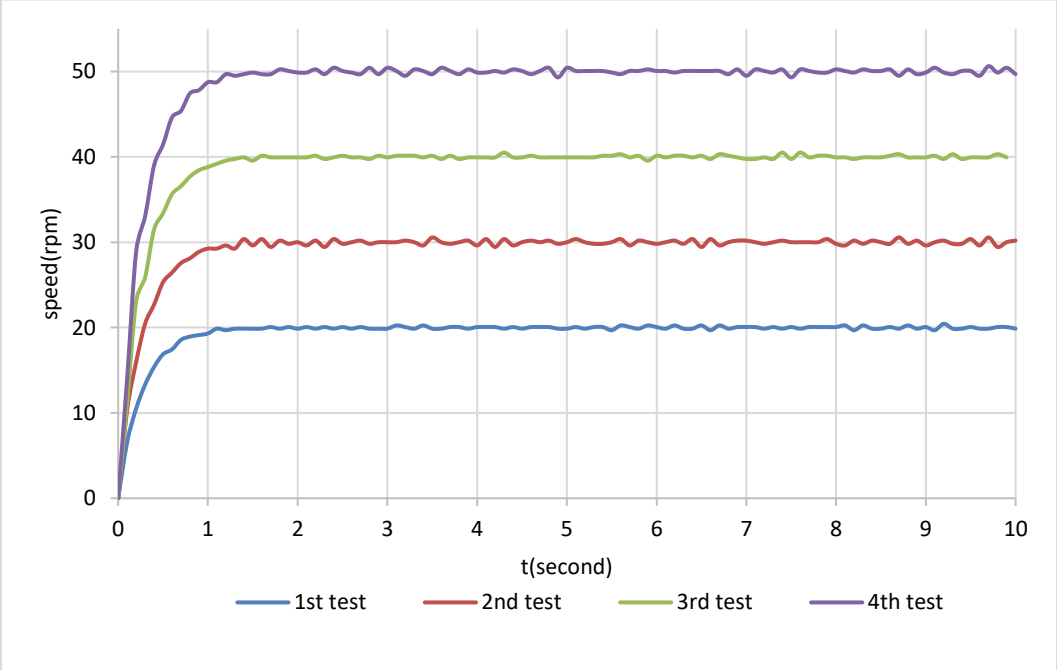

Figure 14. Performance of Omni-wheel speed control system response by various reference signals

To test the performance of the system then given the reference signal speed varies are 20rpm, 30rpm, 40rpm, and 50rpm as shown in Figure 14. The result of the observation shows that the signal response system has been able to follow the reference signal given by settling time (t_s) and e_{ss} at about 1.1s and 0.648%. This proves that the effects of integrator wind-up have been avoided.

CONCLUSION

This study has been implemented a PID controller for Omni-wheel speed control which assumed a wind-up integrator effect certainly occur. The experimental based heuristic approach has been used to derive a combination of PID controller parameters that yield the best system performance (error steady state close to zero, settling time as small as possible). Using the tolerance band of +/- 2.5%, the result of observation on the system response is obtained the settling time 1.1s and error steady state 0.648% by K_p = 1.5, K_i = 0.012, and K_d = 10. Performance test of the system has also been done that is by providing several of the speed reference signals with the performance results that are not different from the previous observations. The observation result of the system response signal also shows that the PID controller has been able to avoid the occurrence of the integrator wind-up effect.

Future works are how to analyze the integrator wind-up effect more deeply if the PID controller uses various combinations of its parameters and how the heuristic approach is used to avoid them.

ACKNOWLEDGMENTS

The authors would like to express their heartfelt thanks to The Modern Computing Research Center, Department of Information Technology, State Polytechnic of Samarinda for providing all their support.

REFERENCES

1. G. Nautiyal, H. Singh, M. Pandey, R. Rana, and S. Rawat, "Modified Microcontroller based Speed Control of DC motor using PWM," International Journal of Energy Technology and Management, vol. 1, pp. 29-36, 2017.
2. S. S. Thakare and S. Kompelli, "Design and implementation of dc motor speed control based on pic microcontroller," International Journal Of Engineering And Computer Science, vol. 3, pp. 8075-8079, 2014.
3. S. Azadi and M. Nouri, "Utilizing Azadi controller to stabilize the speed of a DC motor," presented at the International Conference on Advanced Mechatronic Systems (ICAMechS), Tokyo, Japan, 2012.
4. L. B. Palma, F. V. Coito, B. G. Ferreira, and P. S. Gil, "PSO based on-line optimization for DC motor speed control," pp. 301-306, 2015.
5. Z. Q. Zheng, Y. H. Zhang, and J. S. Zhang, "Application of Kalman Filter in DC Motor Speed Control System," Applied Mechanics and Materials, vol. 150, pp. 129-132, 2012.
6. R. Silva-Ortigoza, V. M. Hernandez-Guzman, M. Antonio-Cruz, and D. Munoz-Carrillo, "DC/DC Buck Power Converter as a Smooth Starter for a DC Motor Based on a Hierarchical Control," IEEE Transactions on Power Electronics, vol. 30, pp. 1076-1084, 2015.
7. T. R. Dil Kumar and S. J. Mija, "Dynamic SMC control scheme with adaptively tuned PID controller for speed control of DC motor," pp. 187-191, 2015.
8. V. Srikanth, G. Prasad, B. C. Chakrapani, S. Jaswanth, V. R. Shankar, and P. Sridhar, "Microcontroller Based Speed Control of a DC Motor Using PWM Technique," International Journal of Advanced Trends in Computer Science and Engineering, vol. 5, pp. 124-127, 2016.
9. H. L. Chan and K. T. Woo, "Closed Loop Speed Control of Miniature Brushless DC Motors," Journal of Automation and Control Engineering, pp. 329-335, 2015.
10. M. S. S. Umadi and D. Patil, "DC Motor Seed Control using Microcontroller," International Journal of Engineering and Techniques, vol. 2, pp. 70-74, 2016.
11. A. Divakar, J. Joseph, J. T. George, N. N. Prabhu, and A. Nabi, "A Study on DC Motor Operations and Speed Control Using Microcontroller," International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, vol. 5, pp. 2460-2464, 2016.
12. P. J. Patel, H. J. Patel, A. D. Unadkat, C. U. Patel, and A. P. S. Bhanderi, "PWM Based Speed Control for a DC Motor," International Journal of Science, Engineering and Technology Research (IJSETR), vol. 6, pp. 665-667, 2017.
13. S. K. Arvind, T. A. Arun, T. S. Madhukar, and J. Deka, "Speed Control of DC Motor using PIC 16F877A Microcontroller," Multidisciplinary Journal of Research in Engineering and Technology, vol. 1, pp. 223-234, 2014.
14. E. Guerrero, J. Linares, E. Guzman, H. Sira, G. Guerrero, and A. Martinez, "DC Motor Speed Control through Parallel DC/DC Buck Converters," IEEE Latin America Transactions, vol. 15, pp. 819-826, 2017.
15. R. K. Munje, P. P. Shinde, and S. S. Kale, "Performance comparison of PI/PID controllers for DC motor," pp. 1-3, 2014.
16. J. Ledin, Embedded Control Systems in C/C++ using MATLAB. San Fransisco: byCMP Books, 2004.
17. L. Magnani, Heuristic Reasoning vol. 16: Springer International Publishing Switzerland, 2015..