Journal of Robotics and Control (JRC)
Volume 5, Issue 6, 2024
ISSN: 2715-5072, DOI: 10.18196/jrc.v5i6.21807

Optimization of Proportional Integral Derivative Controller for Omni Robot Wheel Drive by Using Integrator Wind-up Reduction Based on Arduino Nano

Supriadi [1*], Agusma Wajiansyah [2], Mohammad Zainuddin [3], Arief Bramanto Wicaksono Putra [4]
[1, 2] Computer Enginnering, Politeknik Negeri Samarinda, Samarinda, Indonesia
[3] Electrical Department, Politeknik Negeri Samarinda, Samarinda, Indonesia
[4] The Applied Modern Computing & Robotic Systems Unit, Politeknik Negeri Samarinda, Samarinda, Indonesia
Email: [1] supriadi.polnes@gmail.com, [2] agusma.wajiansyah@gmail.com, [3] zainuddin011062@gmail.com,
[4] ariefbram@gmail.com
*Corresponding Author

***Abstract*—The experimental object used is a three-wheeled omni-robot frame, where the wheel axes have an angle difference of 120 degrees from each other. The Omni wheels have a diameter of 48 mm connected to the DC motor axis through a gearbox, which has a ratio of 80 to 1. Each wheel has been controlled using a proportional plus integral plus derivative (PID) controller embedded in a microcontroller, which is an Arduino nano board. The motor axis is equipped with a two-phase optical encoder that definitively generates four cycles per revolution for wheel speed acquisition as the controller input. The wheel speed control signal is distributed to the wheel through the H bridge as the controller output. The controller constants have been directly tuned to the robot frame's physical omni-wheel speed control system. The controller is tuned to minimize steady-state error, achieve fast settling times, and minimize overshoot. The best constants obtained are 1.5 (proportional), 0.012 (integral), and 10 (derivative). Using a tolerance band of +/- 2.5%, the system achieved a settling time of 1.1 seconds and a steady-state error of 0.3%. The control system is unstable when the actuator is saturated, which produces oscillations. Controller optimization has been successful by using integrator wind-up reduction. The steady-state average error was reduced to 9.95% without oscillation after optimization, compared to 46.37% with oscillations before optimization. The controller has been validated with speed-tracking tests on all velocity vector regions. The robot frame has been tested with basic maneuvers such as rotation, concerning, forward, and sideways.**

Keywords—Physical Hardware; Manual-Tuning; PID; Integrator Wind-Up; Omni Wheel, Omni Robot; CLI; Arduio Nano; Encoder; CPR; PWM.

I. INTRODUCTION

The demand for Robots is increasing in Industry and Society. Omni-wheel drive can be applied to robots used in industries and household appliances. For example, a robot vacuum cleaner is a type of vacuum cleaner that is designed to clean floors autonomously. It uses sensors to navigate around the room and avoid obstacles, Autonomous Mobile Robot (AMR) which is useful for moving goods from one place to another, Automated Ultraviolet C Light Mobile Robot (AUMR) is Ultraviolet light carrier robot for disinfection viruses in the room [1]-[3], as well as various research on omnidirectional robots [4]-[50]. An Omni robot is a robot that can move on a flat surface in all directions. Omni robots can move like they have normal wheels or move sideways using wheels along their circumference. The omnidirectional wheel allows the robot to change from a non-holonomic robot to a holonomic robot. Non-holonomic robots are robots that use normal wheels and only have two of the three degrees of freedom that can be controlled, which moves forward or backward, and rotates. The robot cannot move sideways which makes it slower and less efficient at achieving its given goals. The holonomic omnidirectional wheel can solve this problem because it is very maneuverable. The robot moves with conventional wheels or conventional wheel mobile robots (WMRs), its direction is limited because the robot cannot move sideways without initial maneuvers. Various mechanisms were developed to increase the maneuverability of the WMR. A design that uses three center wheels with steering and independent driving capability, capable of continuously changing its orientation up to 360°, so this design can be called omnidirectional. It is not widely known that the first omnidirectional wheel was patented in 1919 by Grabowiecki. The assembly consists of the main wheels and transverse rollers, as used by most of the RoboCup team. The inventor is considering a vehicle design capable of moving forward and sideways without a steering wheel. Robots built with these wheels usually have three omnidirectional wheels arranged in a △ or Y pattern [51][59].

Research on the omni triangle robot drive has been carried out, which is Integral Proportional (PI) control to get the desired trajectory [60]. Control development research for a three-wheeled omnidirectional moving robot for operation on flat terrain. The heart of the robot base is the omnidirectional wheel. As well as providing normal traction to the rotor axis like regular wheels, they can slide parallel to the rotor axis without much friction. Design and implementation of a robot using three omnidirectional wheels installed with 120° spacing. Omni wheels robots have high efficiency in robot movement but also have their challenges in controlling this robot, especially robots with three omni wheels (3WD). This paper proposes a robot control system using the pole

placement state feedback method to control the inner loop which controls the body reference speed, and the PID method is used to control the outer loop which controls the robot's position in world coordinates. The results obtained using computer simulations show that the proposed method can control the robot's position in world coordinates even though there is an overshoot of 5- 10% on the X and yaw axis, however, the Y axis performance is overshot by more than 100% because of a coupled effect that still cannot be eliminated [61]-[70].

The algorithm for the direction of the robot with Omni wheels can be controlled with an embedded system device in the form of a microcontroller [71][72], and various studies using Arduino-based embedded PID [73]-[80]. In some cases, the robot consists of more than one microcontroller, and the communication between them uses multi-drop communication with a master-slave mode such as Inter-Integrated Circuit (I2C) [81]-[83].

To achieve precise movement of the robot following the specified vector quantity, the speed of the wheels must spin in each Omni wheel shaft and remain consistent at the predetermined reference speed. This necessitates stringent control over the number of turns on each Omni wheel shaft. Multiple techniques and methods exist to ensure the persistence of the number of turns per Omni wheel shaft at the reference speed. This study uses classical PID control, with the controller calculating the error signal to generate the control signal for the PWM signal fed to the Omni wheel. An encoder captures the error signal and then compares it with the reference. These components are part of a closed-loop system implemented using a microcontroller-based embedded system. When implementing PID controllers, a common issue arises when the actuator cannot fully respond to command behavior, leading to the integrator wind-up effect. This can result in a 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 [84]-[95]. This study aims to obtain the best PID control parameters. Then optimize the controller to avoid the occurrence of the integrator wind-up effect and to achieve its best system performance. The tuning process is done experimentally by using the heuristic approach manually which is done in such a way as to obtain the system response as expected. The heuristic approach is a problem-solving method that leverages practical techniques with the aim of achieving a specific goal. Essentially, heuristics involves using rules of thumb, where the solution process relies on intuitive or empirical rules [96]. In this research, a heuristic approach is employed to determine a set of PID controller parameters that yield the most optimal system response performance.

II. METHOD

The experimental block diagram for open-loop and closed-loop control system data acquisition is illustrated in Fig. 1. Connecting a laptop to the Arduino Nano board via USB to serial is essential for tasks, such as data acquisition and tuning processes. The first one is to gathering datas to know how step response of the system on open loop state. Where a DC voltage of 12 volts is given directly to the DC motor as a plant without using a controller. The second is used to observe the system response when using a PID controller and tuning it, and the latter is used to test the controller with several speed targets.

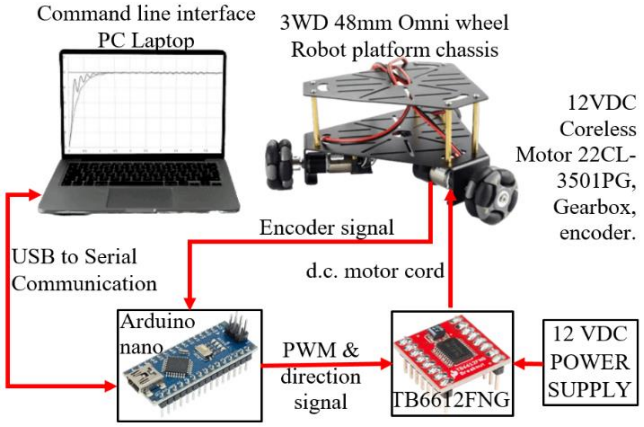

Fig. 1. Experimental block diagram

On the Arduino nano board, a proportional plus integral plus derivative (PID) controller and a command line interface (CLI) have been embedded, which can be observed using a laptop. Using CLI has made it easier to tune the controller constants repeatedly until the desired performance is achieved. The performance of the control system has been observed by considering the smallest overshoot, the fastest setting time, and the smallest steady-state error. The process of tuning controller constants has been done manually on physical systems directly, but not using mathematical models.

Driving an omni wheel with an Arduino board using an H-bridge, so that it can rotate forward or reverse. The H-bridge that has been used is TB6612FNG which is sufficient for the power requirements of the 22CL-3501PG motor. The motor torque is distributed through a gearbox with a ratio of 80:1 connected to the wheel. The number of wheel rotations is determined by pulse width modulation (PWM) with a maximum resolution of 8-bit registers of 255 when the pulse width is 100%. While to get the wheel revolution using the encoder that has been installed on the motor axis, it has 4 cycles per revolution (CPR). So that the encoder output as the wheel speed input and PWM as the control signal output from the PID controller to control the wheel.

The experimental stages start from the acquisition of open loop system data using a step signal as a reference signal. The second is the controller tuning process. The third is observing the performance of the control system when there is an integral wind-up effect which usually has oscillations. The fourth is optimizing the control system so that it can work when there is an integral wind-up effect. The last stage is using the controller on the three omni wheels on the robot frame.

A. Speed Control of Omni Wheel by PID Controller

The Block diagram of speed control of Omni wheel is shown in Fig. 2.

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

The full PID controller algorithm is shown again in (1). With the correct choice of signs for Kp, Ki, and Kd, a PID controller will generate an actuator command that attempts to drive the error to zero with the proportional gain, remove the steady-state error with the integral gain, and dampen the response with the derivative gain.

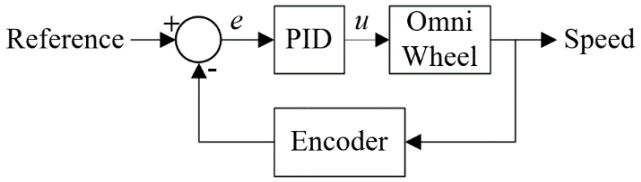

Fig. 2. Block diagram closed loop Omni wheel speed control by using PID controller

B. Tuning a PID Controller

The following are the steps for tuning the PID controller constants,

1) Start by implementing a controller with the algorithm of (1) and choose a small value of Kp. Set $Ki = 0$ $Kd = 0$ for now. A small value of Kp 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 Kp and repeat the test. The speed of the response should increase. If Kp becomes too large, overshoot and oscillation can occur.

4) Increase the value of Kd to reduce any overshoot and oscillation to an acceptable level. If the speed of the response becomes unacceptably slow, try reducing the value of Kd.

5) Continue increasing the value of Kp and adjusting Kd as needed while repeating the test. Watch for the appearance of actuator saturation and reduce Kp if unacceptable saturation occurs.

6) Set Ki 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 Ki and repeating the test. If Ki becomes too large, the overshoot in response to a step input will become excessive. Select a value of Ki 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. If actuator saturation is a problem, continue to PID with intregrator wind-up reduction.

C. PID with Intregrator Wind-up Reduction

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 Fig. 3. Fig. 3 shows the control signal (u), the response signal (y), and the reference (y_{ref}) in the case where the control signal becomes saturated.

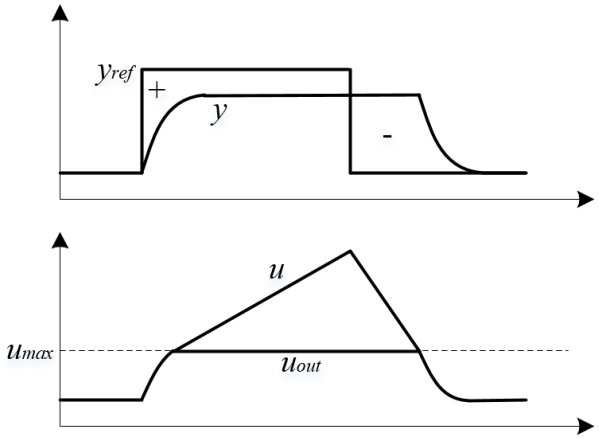

Fig. 3. Illustration of Integral Windup effect

After the first reference 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.

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:

$$u = K_p \left(e + K_i \int_0^t qe \, d\tau + K_d \frac{de}{dt} \right) \quad (2)$$

Where, q is equal to one if the absolute e is less than E, else q is equal to zero, u is control signal, e is error signal, and Kp, Ki, Kd 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. Equation (2) is mathematically describes how the tuning of PID parameters is done by setting the cut-off level E where the amplitude error becomes "large". Fig. 4 is a PID algorithm using a integrator wind-up reducer that works at every specified time step.

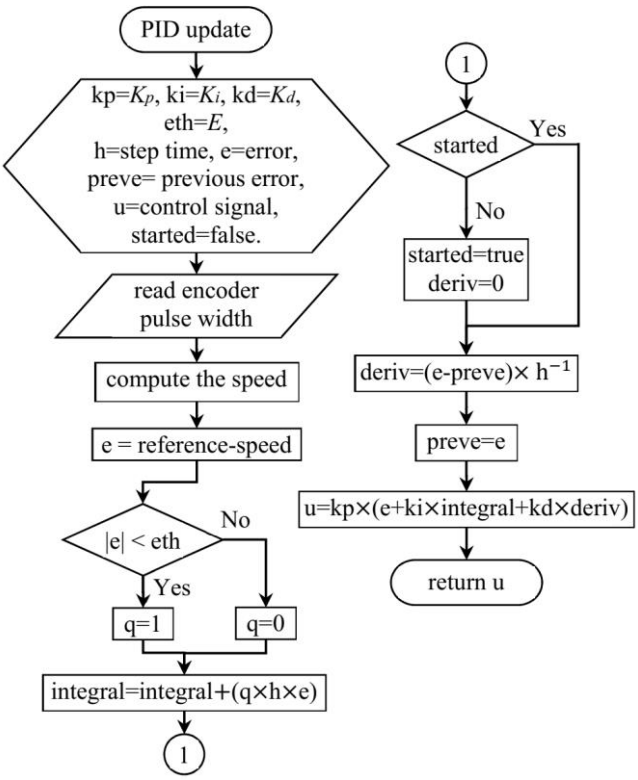

Fig. 4. The algorithm of updating the PID controller parameters with integrator wind-up reduction

D. Revolutions per Minute (rpm) Computing

Equation (3) is revolutions per minute (rpm) calculation. To know the number of speed or rpm on the wheel, which must be known first is CPR and periods (τ) 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 (3). Where n, τ, and CPR are rotation per minutes (rpm), period (μs), and cycle per revolution, respectively. Twice the pulse out in the Fig. 5 is equal to the period (τ).

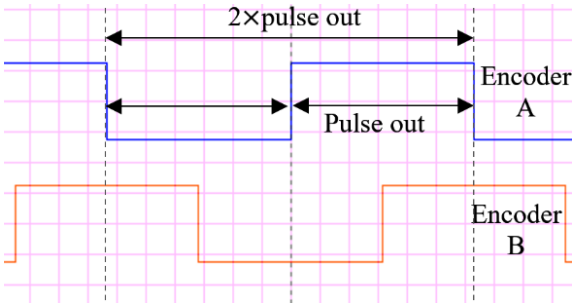

Fig. 5. TTL signal DC motor encoder

$$n = \frac{\left(\frac{60{,}000{,}000}{\tau}\right)}{CPR} \quad (3)$$

III. Results and Discussion

A. Open Loop Step Response

Fig. 6 and Table I are the real open loop step responses DC motor speed with omni wheel. The real open loop step response is important to determine the error threshold (E) that the actuator can generate. From the Table I it can be seen that the actuator has a actuator saturation of around 63.3 rpm if the power supply given to the motor is 12 Volts. This voltage is the same as the control signal generating a maximum PWM equal to 100%.

TABLE I. Real Open Loop Wheel Step Response

Time	Voltage	Speed	Time	Voltage	Speed	Time	Voltage	Speed
(s)	(V)	(rpm)	(s)	(V)	(rpm)	(s)	(V)	(rpm)
0	12.1	0	1.7	12.1	63	3.4	12.1	63.56
0.1	12.1	32.44	1.8	12.1	63.38	3.5	12.1	63.38
0.2	12.1	63	1.9	12.1	63.38	3.6	12.1	63.56
0.3	12.1	63.38	2	12.1	63.56	3.7	12.1	63
0.4	12.1	63.38	2.1	12.1	63.38	3.8	12.1	63.19
0.5	12.1	63	2.2	12.1	63.38	3.9	12.1	63
0.6	12.1	63.19	2.3	12.1	63	4	12.1	63.19
0.7	12.1	63	2.4	12.1	63.19	4.1	12.1	63.38
0.8	12.1	63	2.5	12.1	63.38	4.2	12.1	63.56
0.9	12.1	63.38	2.6	12.1	63.56	4.3	12.1	63.19
1	12.1	63.38	2.7	12.1	63.38	4.4	12.1	62.81
1.1	12.1	63.56	2.8	12.1	63.56	4.5	12.1	63
1.2	12.1	63.38	2.9	12.1	63	4.6	12.1	63.19
1.3	12.1	63.75	3	12.1	63.19	4.7	12.1	63
1.4	12.1	63	3.1	12.1	63	4.8	12.1	63.38
1.5	12.1	63.38	3.2	12.1	63.19	4.9	12.1	63.56
1.6	12.1	63.19	3.3	12.1	63.38	5	12.1	63.38

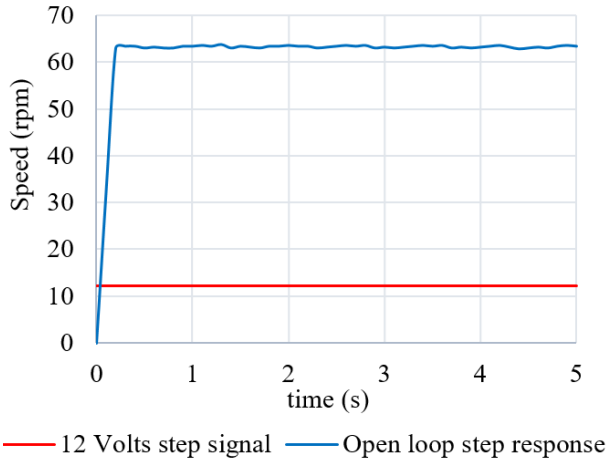

Fig. 6. Real open loop step response of wheel speed

B. *Tuning Process of Closed-loop PID Controller Without Wind-up Reduction*

The determination of the speed reference during the tuning process is selected below the output value of the open loop system response speed when tested using a 12 Volts step signal, which is currently being used at 20 rpm. This is to avoid actuator saturation conditions.

Fig. 7 to Fig. 11 illustrate the tuning process until the best speed control performance is obtained. Fig. 7 shows the system closed-loop response without the controller ($Kp = 1$, $Ki = Kd = 0$). 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%. To perform the tuning parameter of PID controller used, the first step is to set the Kp value until the oscillation of response signal. This can be seen in Fig. 8. With $Kp = 2$, the e_{ss} decreases to 70.9% but the oscillation increases. Since the oscillation increases, the Kp value is reduced to 1 as the Kd value increases until the oscillation of the system response decreases. To reduce the e_{ss} value then set the value of Ki = 0.005. This setting yields e_{ss} = 3.49% as shown in Fig. 9. To minimize settling-time (ts), the value of Kp is increased to 1.5 as shown in Fig. 10. By increasing the Kp value above 1.5 then there will be overshoot on the system response so that the best value of Kp is 1.5.

By using the tolerance band of +/- 2.5% as shown in Fig. 11 then the observation obtained the settling time of 1.1 seconds and ess = 0.3% by Kp =1.5, Ki = 0.012, Kd = 10 and the reference signal speed of 20 rpm.

To test the performance of the system then given the reference signal speed varies are 5 rpm, 10 rpm, 20 rpm, 30 rpm, 40 rpm, 50 rpm, and 60 rpm as shown in Fig. 12. The result of the observation shows that the signal response system has been able to follow the reference signal given by settling time (ts) and e_{ss} at about 1.1 seconds and 0.3% respectively. Fig. 13 is the result of testing that has been tried on the control system before optimization with the integrator wind-up reduction. When it has been tested with the speed reference around the actuator saturation, which is 63 rpm and 65 rpm. It can be seen that, when the speed reference is above the actuator saturation, the greater the oscillation.

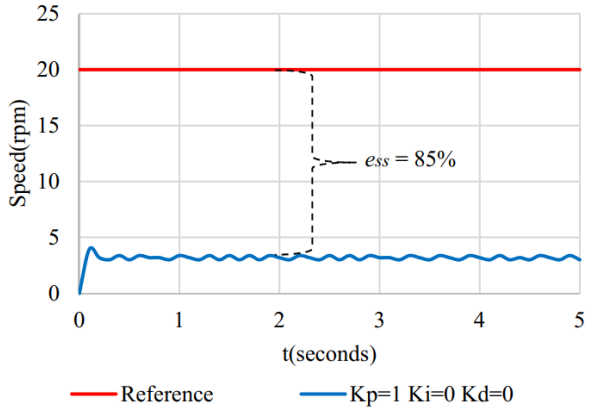

Fig. 7. Wheel closed-loop system response without controller

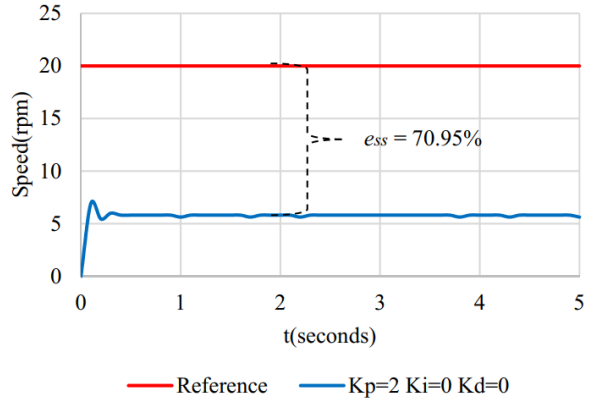

Fig. 8. Closed-loop system response with P controller (Kp = 2)

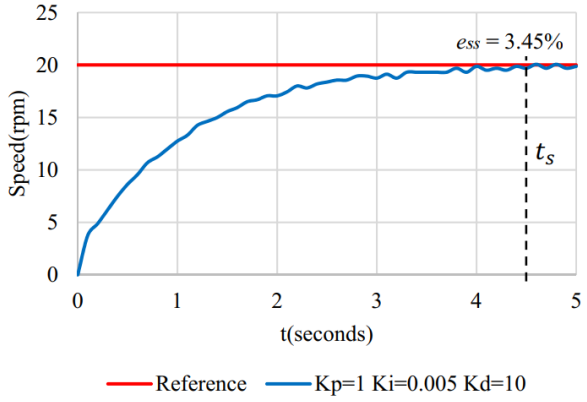

Fig. 9. Closed-loop system response with PID controller (Kp = 1, Ki = 0.005, Kd = 10)

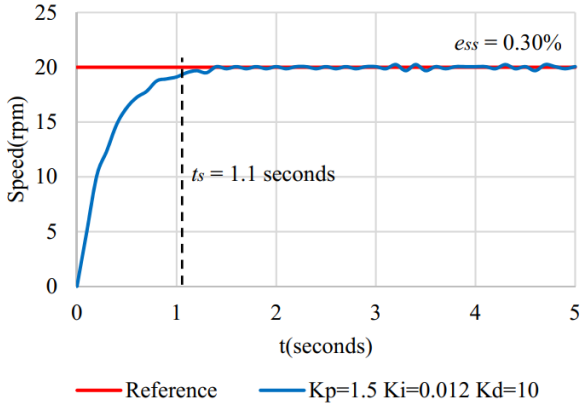

Fig. 10. Closed-loop system response with PID controller (Kp = 1.5, Ki = 0.012, Kd = 10)

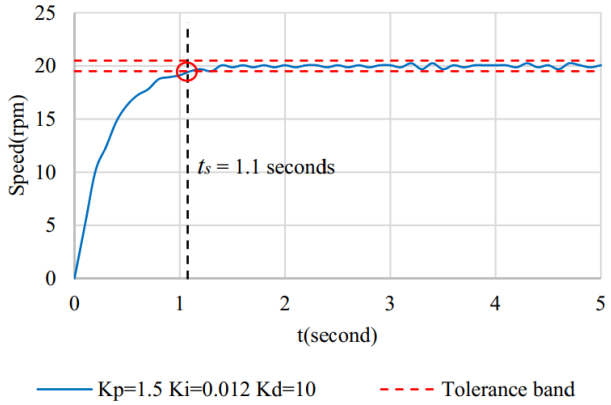

Fig. 11. Performance of Omni-wheel speed control system response at 20 rpm reference signal

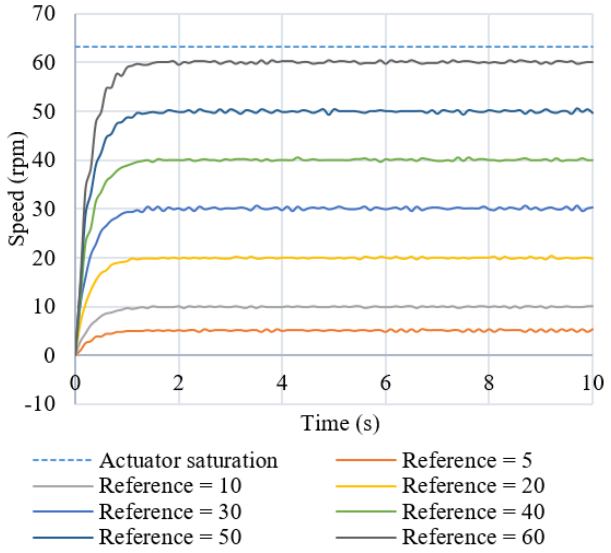

Fig. 12. Response performance of wheel speed control system with various reference signals under actuator saturation

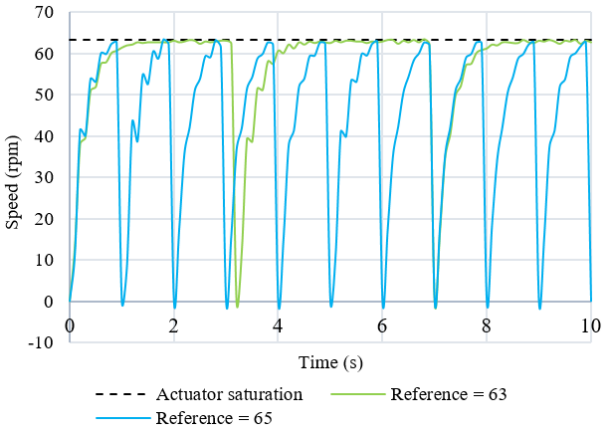

Fig. 13. Response performance of the wheel speed control system with various reference signals around actuator saturation

Fig. 14 is the result that has been tested with a reference speed of 70 rpm, which is greater than the actuator saturation, the oscillation is greater. Increasing control signal caused by increasing integral error beyond the actuator's capabilities, that is called as integral wind-up as shown in Fig. 15. Fig. 16 is the control signal *u* when the speed reference value is equal to 70 rpm, which exceeds the saturation of the control signal.

In the experiment, the saturation of the control signal was 255, this value refers to the resolution capacity of the pulse width modulation (PWM) register, which is only 8 bits. Fig. 17 shows the integral control signal that must be eliminated when it exceeds 14212.5, because this value is the actuator is reaching its saturation.

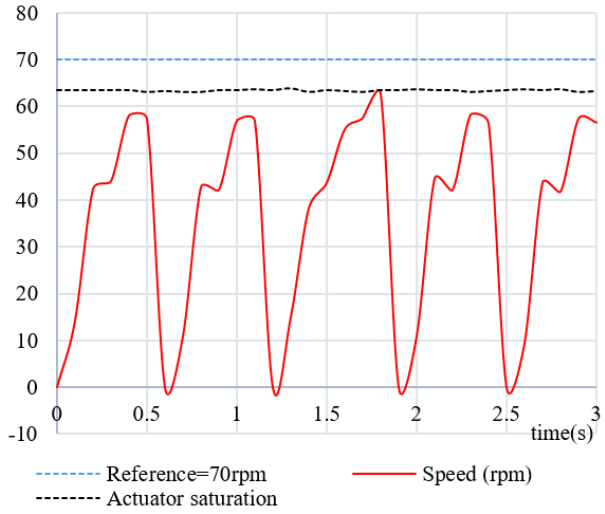

Fig. 14. Response performance of the wheel speed control system with reference signals above actuator saturation

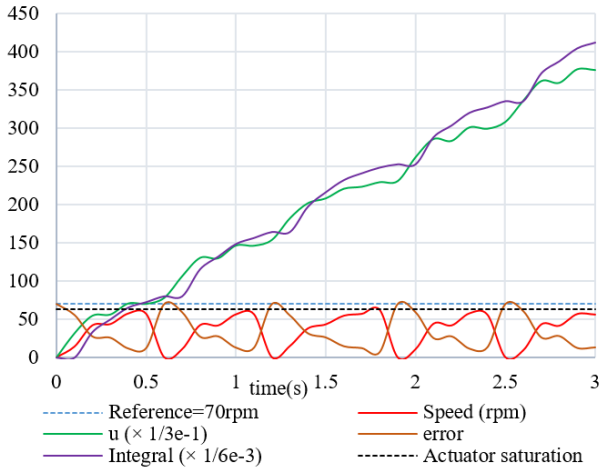

Fig. 15. Response performance of the wheel speed control system when integrator wind-up occurs

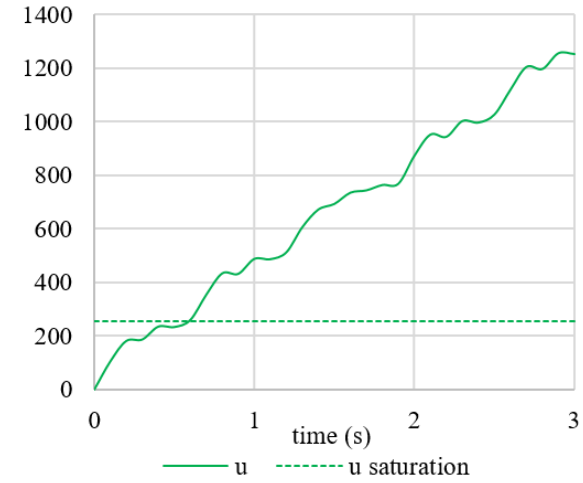

Fig. 16. Control signal *u* when it exceeds the control signal saturation

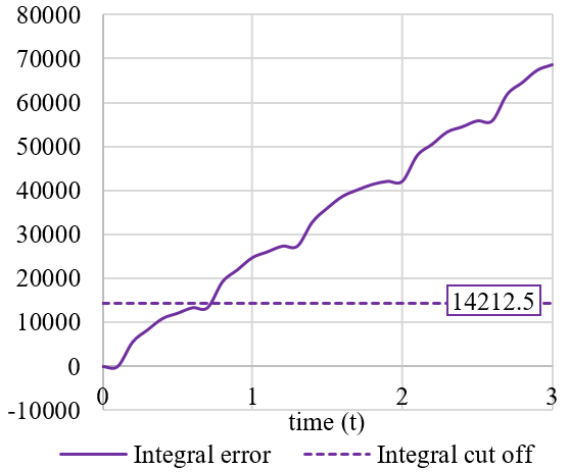

Fig. 17. The integral control signal must be cut off when it exceeds 14212.5

C. PID Controller With Wind-up Reduction

As discussed in the method, the technique of turning off integration when the amplitude of the error signal is above a threshold can reduce the effects of integrator windup to a tolerable level. Which in the open-loop system response experiment has been known that the actuator is saturated at a speed of 63.3 rpm or equivalent to the control signal value of 255. In the experiment that has been done, the value of 63.3 is used as the *E* value which is equivalent to 255 in the control signal. The threshold value of 255 is used as the decision to disable the integral value so that it does not increase.

Fig. 18 is controller response at 70 rpm reference after threshold of control signal and error Integral are implemented, which indicates that integrator windu-up has been successfully removed. Reference equal to 70, control signal equal to 255 (76.5×3.33E+00), integral equal to 14212.5 (85.275×1.67E+02), and the omni wheel speed is stuck at around 63.3 rpm.

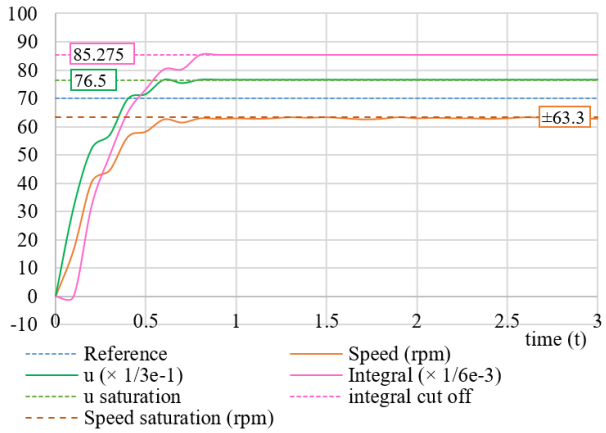

Fig. 18. Response performance of control system with a 70 rpm reference after integrator wind-up reduction is implemented

When the reference is 70 rpm in the steady state region, it has been shown that the average error value is 46.37% with oscillation before optimization and 9.95% without oscillation after optimization, as shown in Fig. 19.

Fig. 20 shows the tracking of speeds of 70, 40, 70 rpm. The experiment shows that there is a delay effect if the integrator reduction is not implemented. The tracking shows that there is a delay of 1.2 seconds at the high to low speed transition. So that the greater the speed reference value, the greater the delay.

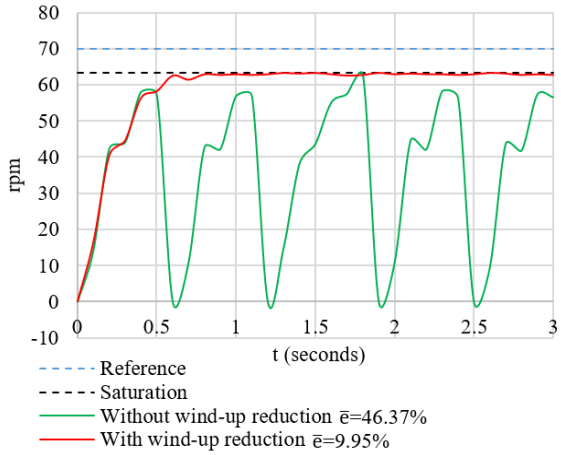

Fig. 19. With and without wind-up reduction at 70 rpm reference

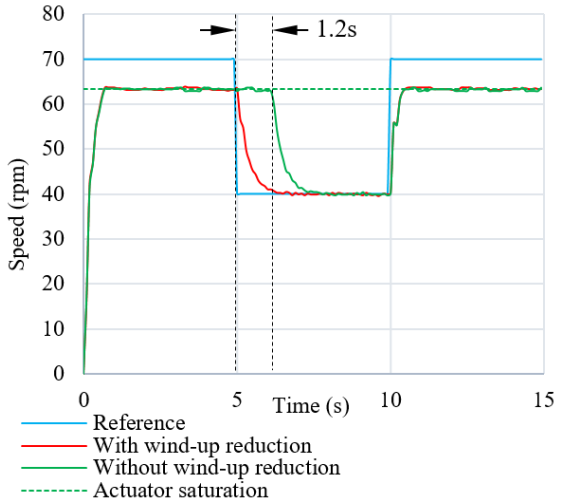

Fig. 20. Speed tracking 70, 40, 70 (rpm) with and without wind-up reduction

Fig. 21 shows the speed tracking in the positive region of 10, 20, 40, 70, 30, 60 rpm. Conversely, Fig. 22 shows the speed tracking in the negative region of -10, -20, -40, -70, -30, -60 rpm. Both experiments involved the saturated actuator region, which is the reference equal to 70 rpm. The results have shown the performance of the control system is able to reject the integrator wind-up effect.

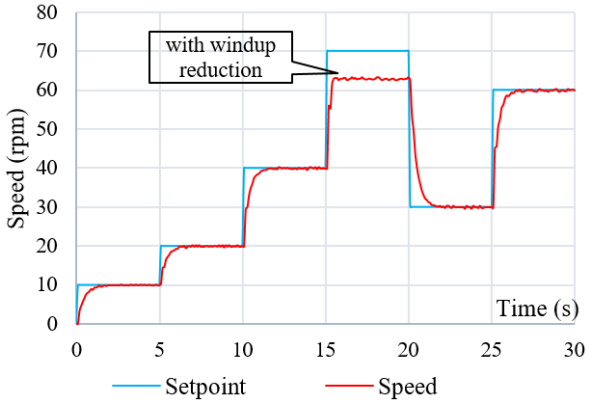

Fig. 21. Speed tracking 10, 20, 40, 70, 30, 60 (rpm)

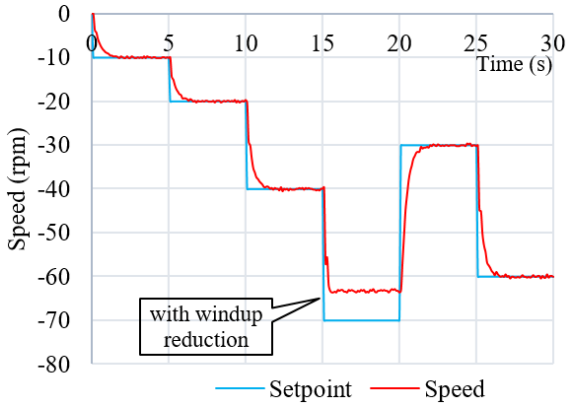

Fig. 22. Speed tracking 10, -20, -40, -70, -30, -60 (rpm)

Fig. 23 shows the speed tracking in all regions 30, 0, -70, 0, 60, 20 rpm. This experiment has been done to find out the response of the control system when transitioning to 0 rpm from the positive axis and the negative axis. This experiment also involves the saturated actuator region, the reference -70 rpm shows the performance of the control system is able to reject the windup effect.

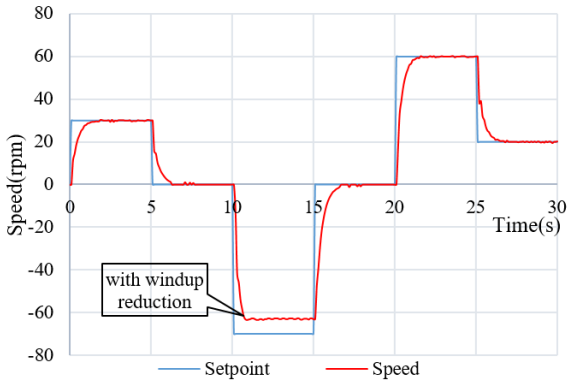

Fig. 23. Speed tracking 30, 0, -70, 0, 60, 20 (rpm)

Fig. 24 shows the speed tracking in the area that does not involve tracking the speed of 0 rpm, such as 30, -40, 50, -60, 50, 40 rpm. This experiment has been done to determine the response of the control system when transitioning from positive to negative and vice versa directly. This illustrates the performance of the control system when the wheel is able to move forward and reverse.

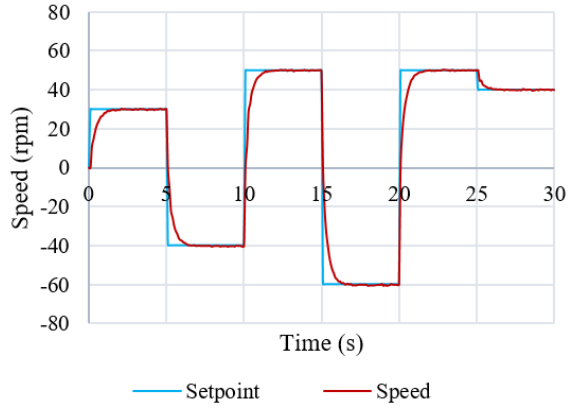

Fig. 24. Speed tracking 30, -40, 50, -60, 50, 40 (rpm)

D. *Performance of PID Controller on 3-Wheeled Omni Robot Frame*

All experiments that have been done, when the PID controller constant tuning process only uses the rear omni wheel. In the basic movement experiment of the robot frame, each rear, right, and left omni wheel uses its own PID controller. The controller constant on each omni wheel uses the same constant as the rear omni wheel PID controller constant.

Fig. 25 shows three Arduino nano boards that have PIDs embedded in them and an Arduino mega board. Each Arduino nano board functions to control the omni wheel. Each Arduino nano board as slave 1, slave 2, slave 3 is connected to an I2C bus so that it can be commanded by the Arduino mega board as the master.

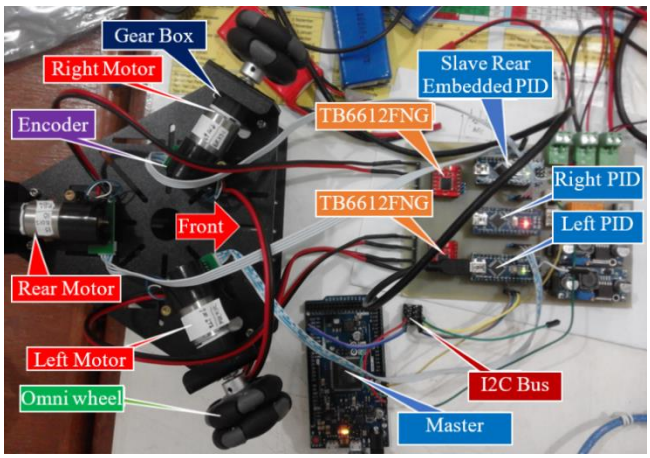

Fig. 25. Experimental object

Fig. 26, Fig. 27, and Fig. 28 respectively, are the responses of the rear, right, and left systems at a reference of 50 rpm. Applying the same PID controller constants to all of them, it is seen that only the right omni wheel has a slight oscillation in the transient region. The e_{ss} values for the rear and right wheels show the same value of 0.24%, while the left wheel has a value of 0.12%.

Fig. 29 shows a top view of the frame to see the layout of the rear, right and left wheels. The red arrow in Fig. 29 is the default direction of the forward wheel rotation. Fig. 30 shows the results of data acquisition of the speed of each wheel, which has been tested only by trying basic frame maneuvers, including rotation, concerning, forward, and sideways movements. The experiment used a reference or set point of 50 rpm to all wheels, and then all the basic movements are performed for 10 seconds.

This experiment has been implemented a PID controller for Omni-wheel speed control which integrator wind-up effect certainly occur, when the reference speed is above 63.3 rpm. Before the integrator wind-up reduction is applied, an integrator wind-up effect occurs which can cause the system to be unable to control the speed when the actuator is saturated. After knowing the cause of the integrator wind-up, the right way to overcome it can be determined.

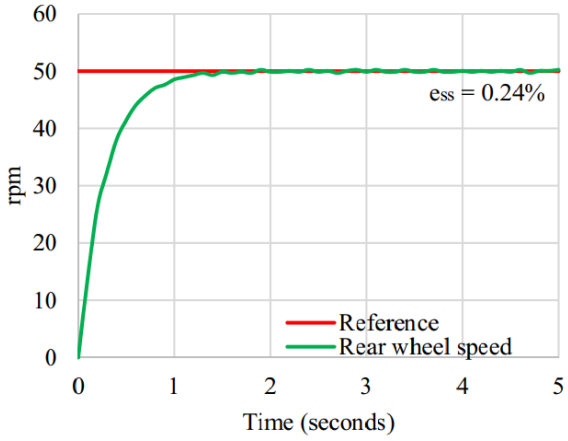

Fig. 26. Rear wheel response at 50 rpm reference

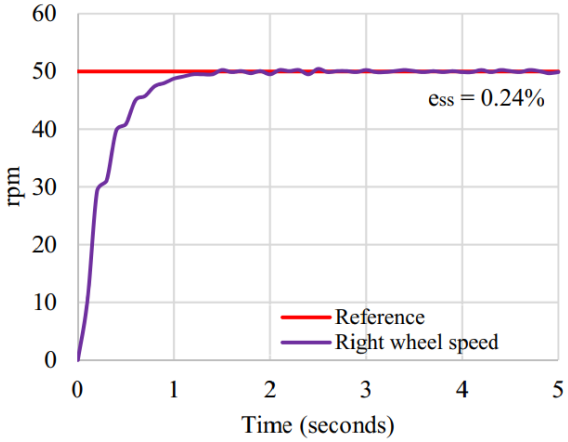

Fig. 27. Right wheel response at 50 rpm reference

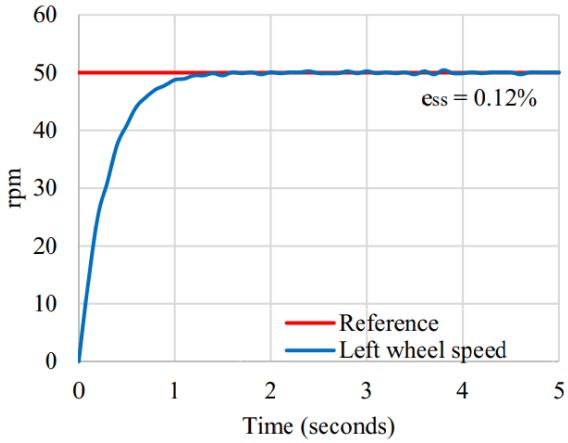

Fig. 28. Left wheel response at 50 rpm reference

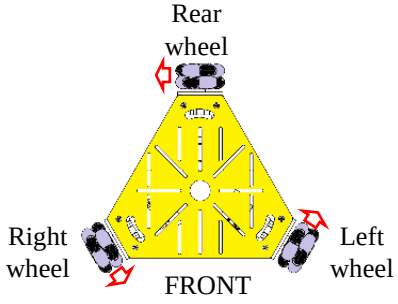

Fig. 29. Top view of the robot platform with 3 omni wheel drive

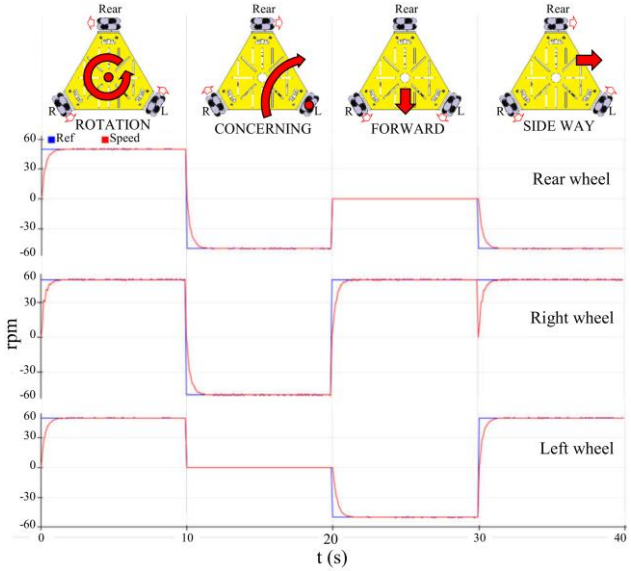

Fig. 30. Basic frame movements of omni wheels at 50 rpm

IV. CONCLUSION

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 1.1 seconds and error steady state 0.3% by Kp = 1.5, Ki = 0.012, and Kd = 10 at 20 rpm reference. 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. Speed tracking has been attempted in positive, zero, and negative, including saturated regions, where the results successfully followed the reference or set point.

Applying the same PID controller constants to all of them with 50 rpm reference, it is seen that only the right omni wheel has a slight oscillation in the transient region. The *ess* values for the rear and right wheels show the same value of 0.24%, while the left wheel has a value of 0.12%. Also tested the controller on the frame with basic maneuvers such as rotation, concerning, forward, and sideways movements.

When the reference is 70 rpm in the steady state region, it has been shown that the average error value is 46.37% with oscillation before optimization and 9.95% without oscillation after optimization.

Future works are how to determine the efficiency of the design and implementation of the robot drive. By embedding three PID controls in a microcontroller, it can minimize the dimensions of the embedding system, also improve the movement of the robot frame by using the odometry method.

ACKNOWLEDGMENT

The authors are grateful to the Applied Modern Computing & Robotic Systems Unit Politeknik Negeri Samarinda, East Kalimantan, Indonesia.

REFERENCES

[1] M. A. A. Mutalib and N. Z. Azlan, "Prototype development of mecanum wheels mobile robot: A review," *Applied Research and Smart Technology (ARSTech)*, vol. 1, no. 2, pp. 71–82, 2020, doi: 10.23917/arstech.v1i2.39.

[2] J. Holland, L. M. Kingston, C. Mccarthy, E. Armstrong, P. O'dwyer, F. Merz, and M. Mcconnell, "Service Robots in the Healthcare Sector," *Robotics*, vol. 10, no. 1, p. 47, 2021, doi: 10.3390/robotics10010047.

[3] D. H. T. Kim, T. N. Manh, C. N. Manh, N. D.c Nguyen, D. P. Tien, M. T. Van, and M. P. Xuan, "Adaptive Control for Uncertain Model of Omni-directional Mobile Robot Based on Radial Basis Function Neural Network," *SpringerLink, International Journal of Control, Automation and Systems*, vol. 19, pp. 1715–1727, 2021, doi: 10.1007/s12555-019-1004-6.

[4] T. T. Tun, L. Huang, R. E. Mohan, and S. G. H. Matthew, "Four-wheel steering and driving mechanism for a reconfigurable floor cleaning robot," *Automation in Construction*, vol. 106, 2019, doi: 10.1016/j.autcon.2019.03.017.

[5] T. Terakawa, M. Komori, Y. Yamaguchi, and Y. Nishida, "Active omni wheel possessing seamless periphery and omnidirectional vehicle using it," *Precision Engineering*, vol. 56, pp. 466-475, 2019, doi: 10.1016/j.precisioneng.2019.02.003.

[6] M. Seder *et al*., "Open Platform Based Mobile Robot Control for Automation in Manufacturing," *IFAC-PapersOnLine*, vol. 52, no. 22, pp. 95-100, 2019, doi: 10.1016/j.ifacol.2019.11.055.

[7] D. Nemec, V. Šimák, A. Janota, M. Hruboš, and E. Bubeníková, "Precise localization of the mobile wheeled robot using sensor fusion of odometry, visual artificial landmarks and inertial sensors," *Robotics and Autonomous Systems*, vol. 112, pp. 168-177, 2019, doi: 10.1016/j.robot.2018.11.019.

[8] G. Bayar and S. Ozturk, "Investigation of The Effects of Contact Forces Acting on Rollers Of a Mecanum Wheeled Robot," *Mechatronics*, vol. 72, 2020, doi: 10.1016/j.mechatronics.2020.102467.

[9] S. Mellah, G. Graton, E.-M. E. Adel, M. Ouladsine, and A. Planchais, "Trajectory reconfiguration for time delay reduction in the case of unexpected obstacles: application to 4-mecanum wheeled mobile robots (4-MWMR) for industrial purposes," *IFAC-PapersOnLine*, vol. 53, pp. 15653-15658, 2020, doi: 10.1016/j.ifacol.2020.12.2546.

[10] H. Taheri and C. X. Zhao, "Omnidirectional mobile robots, mechanisms and navigation approaches," *Mechanism and Machine Theory*, vol. 153, 2020, doi: 10.1016/j.mechmachtheory.2020.103958.

[11] A. S. Staal *et al*., "Towards a Collaborative Omnidirectional Mobile Robot in a Smart Cyber-Physical Environment," *Procedia Manufacturing*, vol. 51, pp. 193-200, 2020, doi: 10.1016/j.promfg.2020.10.028.

[12] M. W. Mehrez, K. Worthmann, J. P. V. Cenerini, M. Osman, W. W. Melek, and S. Jeon, "Model Predictive Control without terminal constraints or costs for holonomic mobile robots," *Robotics and Autonomous Systems*, vol. 127, 2020, doi: 10.1016/j.robot.2020.103468.

[13] H. Wu, W. Xu, B. Yao, Y. Hu, and H. Feng, "Interacting Multiple Model-Based Adaptive Trajectory Prediction for Anticipative Human Following of Mobile Industrial Robot," *Procedia Computer Science*, vol. 176, pp. 3692-3701, 2020, doi: 10.1016/j.procs.2020.09.330.

[14] I. M.-Caireta, E. Celaya, and L. Ros, "Model Predictive Control for a Mecanum-wheeled Robot Navigating among Obstacles," *IFAC-PapersOnLine*, vol. 54, pp. 119-125, 2021, doi: 10.1016/j.ifacol.2021.08.533.

[15] Z. Sun, S. Hu, D. He, W. Zhu, H. Xie, and J. Zheng, "Trajectory-tracking control of Mecanum-wheeled omnidirectional mobile robots using adaptive integral terminal sliding mode," *Computers & Electrical Engineering*, vol. 96, 2021, doi: 10.1016/j.compeleceng.2021.107500.

[16] Z. Sun, H. Xie, J. Zheng, Z.g Man, and D. He, "Path-following control of Mecanum-wheels omnidirectional mobile robots using nonsingular terminal sliding mode," *Mechanical Systems and Signal Processing*, vol. 147, 2021, doi: 10.1016/j.ymssp.2020.107128.

[17] G. Peng, Z. Lu, Z. Tan, D. He, and X. Li, "A novel algorithm based on nonlinear optimization for parameters calibration of wheeled robot mobile chasses," *Applied Mathematical Modelling*, vol. 95, pp. 396-408, 2021, doi: 10.1016/j.apm.2021.02.012.

[18] H. Eschmann, H. Ebel, and P. Eberhard, "Data-Based Model of an Omnidirectional Mobile Robot Using Gaussian Processes," *IFAC-PapersOnLine*, vol. 54, pp. 13-18, 2021, doi: 10.1016/j.ifacol.2021.08.327.

[19] D. B.-Vásquez, M. M.-Herrera, and J. S. B.-Valencia, "Open source and open hardware mobile robot for developing applications in education and research," *HardwareX*, vol. 10, 2021, doi: 10.1016/j.ohx.2021.e00217.

[20] B. S. Pallares O., T. A. Rozo M., E. C. Camacho, J. G. Guarnizo, and J. M. Calderon, "Design and Construction of a Cost-Oriented Mobile Robot for Domestic Assistance," *IFAC-PapersOnLine*, vol. 54, pp. 293-298, 2021, doi: 10.1016/j.ifacol.2021.10.462.

[21] V. L. Popov, N. G. Shakev, A. V. Topalov, and S. A. Ahmed, "Detection and Following of Moving Target by an Indoor Mobile Robot using Multi-sensor Information," *IFAC-PapersOnLine*, vol. 54, pp. 357-362, 2021, doi: 10.1016/j.ifacol.2021.10.473.

[22] S. Long, T. Terakawa, M. Komori, Y. Nishida, T. Ougino, and Y. Hattori, "Effect of double-row active omni wheel on stability of single-track vehicle in roll direction," *Mechanism and Machine Theory*, vol. 163, 2021, doi: 10.1016/j.mechmachtheory.2021.104374.

[23] M. U. Shafiq, A. Imran, S. Maznoor, and A. H. Majeed, "Real-time navigation of mecanum wheel-based mobile robot in a dynamic environment," *Heliyon*, vol. 10, no. 5, 2024, doi: 0.1016/j.heliyon.2024.e26829.

[24] H. Xing, A. Torabi, L. Ding, H. Gao, W. Li, and M. Tavakoli, "Enhancing kinematic accuracy of redundant wheeled mobile manipulators via adaptive motion planning," *Mechatronics*, vol. 79, 2021, doi: 10.1016/j.mechatronics.2021.102639.

[25] H. Xing, A. Torabi, L. Ding, H. Gao, Z. Deng, V. K. Mushahwar, and M. Tavakoli, "An admittance-controlled wheeled mobile manipulator for mobility assistance: Human–robot interaction estimation and redundancy resolution for enhanced force exertion ability," *Mechatronics*, vol. 74, 2021, doi: 10.1016/j.mechatronics.2021.102497.

[26] L. Qi, T. Zhang, K. Xu, H. Pan, Z. Zhang, and Y. Yuan, "A novel terrain adaptive omni-directional unmanned ground vehicle for underground space emergency: Design, modeling and tests," *Sustainable Cities and Society*, vol. 65, 2021, doi: 10.1016/j.scs.2020.102621.

[27] A. S. Belyaev, O. A. Brylev, and E. A. Ivanov, "Slip Detection and Compensation System for Mobile Robot in Heterogeneous Environment," *IFAC-PapersOnLine*, vol. 54, pp. 339-344, 2021, doi: 10.1016/j.ifacol.2021.10.470.

[28] A. Grabowski, J. Jankowski, and M. Wodzyński, "Teleoperated mobile robot with two arms: the influence of a human-machine interface, VR training and operator age," *International Journal of Human-Computer Studies*, vol. 156, 2021, doi: 10.1016/j.ijhcs.2021.102707.

[29] P. S. Yadav, V. Agrawal, J. C. Mohanta, and M. D. F. Ahmed, "A robust sliding mode control of mecanum wheel-chair for trajectory tracking," *Materials Today: Proceedings*, vol. 56, pp. 623-630, 2022, doi: 10.1016/j.matpr.2021.12.398.

[30] H. Xiao, D. Yu, and C. L. P. Chen, "Self-triggered-organized Mecanum-wheeled robots consensus system using model predictive based protocol," *Information Sciences*, vol. 590, pp. 45-59, 2022, doi: 10.1016/j.ins.2021.12.108.

[31] Z. Miao, F. Zhou, X. Yuan, Y. Xia, and K. Chen, "Multi-heterogeneous sensor data fusion method via convolutional neural network for fault diagnosis of wheeled mobile robot," *Applied Soft Computing*, vol. 129, 2022, doi: 10.1016/j.asoc.2022.109554.

[32] M. K. Nutalapati, L. Arora, A. Bose, K. Rajawat, R. M. Hegde, "A generalized framework for autonomous calibration of wheeled mobile robots," *Robotics and Autonomous Systems*, vol. 158, 2022, doi: 10.1016/j.robot.2022.104262.

[33] C. Wang, J. Ji, Z. Miao, and J. Zhou, "Udwadia-Kalaba approach based distributed consensus control for multi-mobile robot systems with communication delays," *Journal of the Franklin Institute*, vol. 359, pp. 7283-7306, 2022, doi: 10.1016/j.jfranklin.2022.07.046.

[34] O. Oladunjoye *et al*., "Omnidirectional All-Terrain Screw-Driven Robot Design, Modeling, and Application in Humanitarian Demining," *IFAC-PapersOnLine*, vol. 55, no. 27, pp. 7-12, 2022, doi: 10.1016/j.ifacol.2022.10.480.

[35] X. Liu *et al*., "MPC-based high-speed trajectory tracking for 4WIS robot," *ISA Transactions*, vol. 123, pp. 413-424, 2022, doi: 10.1016/j.isatra.2021.05.018.

[36] Z. Sun, S. Tang, Y.g Zhou, J. Yu, and C. Li, "A GNN for repetitive motion generation of four-wheel omnidirectional mobile manipulator with nonconvex bound constraints," *Information Sciences*, vol. 607, pp. 537-552, 2022, doi: 10.1016/j.ins.2022.06.002.

[37] T. Ding, Y. Zhang, G. Ma, Z. Cao, X. Zhao, and B. Tao, "Trajectory tracking of redundantly actuated mobile robot by MPC velocity control under steering strategy constraint," *Mechatronics*, vol. 84, 2022, doi: 10.1016/j.mechatronics.2022.102779.

[38] L. Jiang, S. Wang, Y. Xie, S. Q. Xie, S. Zheng, and J. Meng, "Fractional robust finite time control of four-wheel-steering mobile robots subject to serious time-varying perturbations," *Mechanism and Machine Theory*, vol. 169, 2022, doi: 10.1016/j.mechmachtheory.2021.104634.

[39] Z. Slanina, "Comprehensive study of parking houses for smart cities," *IFAC-PapersOnLine*, vol. 55, no. 4, pp. 1-12, 2022, doi: 10.1016/j.ifacol.2022.06.001.

[40] X. Wu and Y. Huang, "Adaptive fractional-order non-singular terminal sliding mode control based on fuzzy wavelet neural networks for omnidirectional mobile robot manipulator," *ISA Transactions*, vol. 121, pp. 258-267, 2022, doi: 10.1016/j.isatra.2021.03.035.

[41] M. Q. Zaman and H.-M. Wu, "Hand Gesture-based Teleoperation Control of a Mecanum-wheeled Mobile Robot," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 1484-1489, 2023, doi: 10.1016/j.ifacol.2023.10.1841.

[42] G. Bayar and G. Hambarci, "Improving measurement accuracy of indoor positioning system of a Mecanum wheeled mobile robot using Monte Carlo - Latin hypercube sampling based machine learning algorithm," *Journal of the Franklin Institute*, vol. 360, no. 17, pp. 13994-14021, 2023, doi: 10.1016/j.jfranklin.2022.07.037.

[43] D. N. Zakharov *et al*., "Quality Improvements of Omnidirectional Platforms," *IFAC-PapersOnLine*, vol. 56, pp. 2140-2145, 2023, doi: 10.1016/j.ifacol.2023.10.1118.

[44] A. A. S. Gunawan, B. Clemons, I. F. Halim, K. Anderson, and M. P. Adianti, "Development of e-butler: Introduction of robot system in hospitality with mobile application," *Procedia Computer Science*, vol. 216, pp. 67-76, 2023, doi: 10.1016/j.procs.2022.12.112.

[45] Z. Sun, S. Hu, H. Xie, H. Li, J. Zheng, and B. Chen, "Fuzzy adaptive recursive terminal sliding mode control for an agricultural omnidirectional mobile robot," *Computers and Electrical Engineering*, vol. 105, 2023, doi: 10.1016/j.compeleceng.2022.108529.

[46] J. Cenerini, M. W. Mehrez, J.-w. Han, S. Jeon, and W. Melek, "Model Predictive Path Following Control without terminal constraints for holonomic mobile robots," *Control Engineering Practice*, vol. 132, 2023, doi: 10.1016/j.conengprac.2022.105406.

[47] F. Mateusz and J. Bałchanowski, "A Mobile Robot with Omnidirectional Tracks—Design and Experimental Research," *Applied Sciences*, vol. 11, no. 24, 2021, doi: 10.3390/app112411778.

[48] P. Sesmero, Carlos, L. R. Buonocore, and M. D. Castro, "Omnidirectional Robotic Platform for Surveillance of Particle Accelerator Environments with Limited Space Areas," *Applied Sciences*, vol. 11, no. 14, 2021, doi: 10.3390/app11146631.

[49] P.-J. Chen, S.-Y. Yang, Y.-P. Chen, M. Muslikhin, and M.-S. Wang, "Slip Estimation and Compensation Control of Omnidirectional Wheeled Automated Guided Vehicle," *Electronics*, vol. 10, no. 7, 2021, doi: 10.3390/electronics10070840.

[50] M. R. Azizi, A. Rastegarpanah, and R. Stolkin, "Motion Planning and Control of an Omnidirectional Mobile Robot in Dynamic Environments," *Robotics*, vol. 10, no. 1, p. 48, 2021, doi: 10.3390/robotics10010048

[51] C. Wang, X. Liu, X. Yang, F. Hu, A. Jiang, and C. Yang, "Trajectory tracking of an omni-directional wheeled mobile robot using a model predictive control strategy," *Applied Sciences*, vol. 8, no. 2, p. 231, 2018.

[52] D. B. Setiawan *et al*., "Ball Direction Prediction for Wheeled Soccer Robot Goalkeeper Using Trigonometry Technique," *Applied Technology and Computing Science Journal*, vol. 2, no. 1, pp. 39-51, 2019, doi: 10.33086/atcsj.v2i1.1204.

[53] M. B. Emara, A. W. Youssef, M. Mashaly, J. Kiefer, L. A. Shihata, and E. Azab, "Digital Twinning for Closed-Loop Control of a Three-Wheeled Omnidirectional Mobile Robot," *Procedia CIRP*, vol. 107, pp. 1245-1250, 2022, doi: 10.1016/j.procir.2022.05.139.

[54] M. Eyuboglu and G. Atali, "A novel collaborative path planning algorithm for 3-wheel omnidirectional Autonomous Mobile Robot," *Robotics and Autonomous Systems*, vol. 169, 2023, doi: 10.1016/j.robot.2023.104527.

[55] M. D. Correia, A. Gustavo, and S. Conceição, "Modeling of a Three Wheeled Omnidirectional Robot Including Friction Models," *IFAC Proceedings Volumes*, vol. 45, no. 22, pp. 7-12, 2012, doi: 10.3182/20120905-3-HR-2030.00002.

[56] M. Gavani, D. Tanpure, and P. Falake, "Path Planning of Three Wheeled Omni-Directional Robot Using Bezier Curve Tracing Technique and PID control Algorithm," *2019 IEEE Pune Section International Conference (PuneCon)*, pp. 1-6, 2019, doi: 10.1109/PuneCon46936.2019.9105899.

[57] N. Z. Zailan, M. A. Ayob, A. S. Sadun, H. M. Poad, R. Sawarno, and N. Rohaziat, "Obstacle Avoidance of a 3WD Omni-Wheel Mobile Robot in Webots Environment," *2021 IEEE 11th International Conference on System Engineering and Technology (ICSET)*, pp. 143-146, 2021, doi: 10.1109/ICSET53708.2021.9612576.

[58] S. Huang, C. Li, Z. Cai, G. Zhu, L. Yao, and Z. Fan, "Synchronized 2D SLAM and 3D Mapping Based on Three Wheels Omni-directional Mobile Robot," *2019 IEEE 9th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, pp. 1177-1181, 2019, doi: 10.1109/CYBER46603.2019.9066733.

[59] A. Krishnan and P. Sudarshan, "Self-Localization and Waypoints following of Holonomic Three Wheeled Omni-Directional Mobile Robot," *IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*, pp. 253-258, 2021, doi: 10.1109/DISCOVER52564.2021.9663644.

[60] M. A. Kawtharani, V. Fakhari, and M. R. Haghjoo, "Tracking Control of an Omni-Directional Mobile Robot," in *Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, pp. 1-8, 2020. doi: 10.1109/HORA49412.2020.9152835.

[61] J. Meng, A. Liu, Y. Yang, Z. Wu, and Q. Xu, "Two-Wheeled Robot Platform Based on PID Control," *2018 5th International Conference on Information Science and Control Engineering (ICISCE)*, pp. 1011-1014, 2018, doi: 10.1109/ICISCE.2018.00208.

[62] J. Yu, M. Liang, W. Peng, T. Wu, C. Rong, and D. Zhang, "Speed Control Based on an Improved PID Controller with BP Neural Network for Two Wheel Differential AGV System," *2023 IEEE 6th Student Conference on Electric Machines and Systems (SCEMS)*, pp. 1-5, 2023, doi: 10.1109/SCEMS60579.2023.10379320.

[63] A. El fatimi, A. Addaim, and Z. Guennoun, "Real-time Software In the Loop Simulation for PID Control of the Nanosatellite Reaction Wheel," *2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*, pp. 1-6, 2022, doi: 10.1109/IRASET52964.2022.9737929.

[64] T. A. Mai, T. S. Dang, D. N. Anisimov, and E. Fedorova, "Fuzzy-PID Controller for Two Wheels Balancing Robot Based on STM32 Microcontroller," *2019 International Conference on Engineering Technologies and Computer Science (EnT)*, pp. 20-24, 2019, doi: 10.1109/EnT.2019.00009.

[65] T. Nikita and K. T. Prajwal, "PID Controller Based Two Wheeled Self Balancing Robot," *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 1-4, 2021, doi: 10.1109/ICOEI51242.2021.9453091.

[66] M. Saad, A. Amhedb, and M. A. Sharqawi, "Real Time DC Motor Position Control Using PID Controller in LabVIEW," *Journal of Robotics and Control (JRC)*, vol. 2, pp. 342-347, 2021, doi: 10.18196/jrc.25104.

[67] V. Rajs, N. L. Rašević, M. Z. Bodić, M. M. Zuković, and K. B. Babković, "PID Controller Design for Motor Speed Regulation with Linear and Non-Linear Load," *IFAC-PapersOnLine*, vol. 55, no. 4, pp. 225-229, 2022, doi: 10.1016/j.ifacol.2022.06.037.

[68] R. Kristiyono and Wiyono, "Autotuning Fuzzy PID Controller for Speed Control of BLDC Motor," *Journal of Robotics and Control (JRC)*, vol. 2, pp. 400-407, 2021, doi: 10.18196/jrc.2511440.

[69] S. Istiqphara, A. U. Darajat, Fahmizal, and M. F. Ferdous, " Movement Control of Three Omni-Wheels Robot using Pole Placement State

Feedback and PID Control," *Journal of Fuzzy Systems and Control*, vol. 1, no. 2, pp. 44-48, 2023, doi: 10.59247/jfsc.v1i2.36.

[70] M. Soliman, A. T. Azar, M. Abdallah, and H. H. Ammar, "Path Planning Control for 3-Omni Fighting Robot Using PID and Fuzzy Logic Controller," *Handbook of Experimental Pharmacology*, pp. 442–452, doi: 10.1007/978-3-030-14118-9_45.

[71] M. Hijikata, R. Miyagusuku, and K. Ozaki, "Wheel Arrangement of Four Omni Wheel Mobile Robot for Compactness," *Journals Applied Sciences*, vol. 12, no. 12, pp. 5798, 2022, doi: 10.3390/app12125798.

[72] R. T. Yunardi, D. Arifianto, F. Bachtiar, and J. I. Prananingrum, "Holonomic Implementation of Three Wheels Omnidirectional Mobile Robot using DC Motors," *Journal of Robotics and Control (JRC)*, vol. 2, no. 2, 2021, doi: 10.18196/jrc.2254

[73] A. Hayatal F, S. Syahrorini, A. Ahfas, and Z. Nur F, "Line Tracer Robot Navigation System Using Arduino Uno Microntroller With PID Control," *Academia Open*, vol. 8, no. 2, 2023, doi: 10.21070/acopen.8.2023.7275.

[74] D. U. Suwarno, "Simulation on the effects of the Arduino PID controller parameters using the WOKWI online simulator," *International Conference on Information Science and Technology Innovation (ICoSTEC)*, vol. 1, no. 1, pp. 1-5, 2022, doi: 10.35842/icostec.v1i1.1.

[75] D. Permana, M. W. Sari, and R. H. Hardyanto, "System Water Purifier With Arduino Based PID Control," *Applied Science And Technology Reaserch Journal*, vol. 1, no. 2, pp. 20-25, 2023, doi: 10.31316/astro.v1i2.4642.

[76] S. Nakamori, "Arduino-based PID Control of Humidity in Closed Space by Pulse Width Modulation of AC Voltage," *Wseas Transactions On Circuits And Systems*, vol. 21, pp. 49-56, doi: 10.37394/23201.2022.21.6.

[77] K. Sozanski, "Low Cost PID Controller for Student Digital Control Laboratory Based on Arduino or STM32 Modules," *Electronics*, vol. 12, no. 15, 2023, doi: 10.3390/electronics12153235.

[78] A. Kherkhar, Y. Chiba, A. Tlemçani, and H. Mamur, "Thermal investigation of a thermoelectric cooler based on Arduino and PID control approach," *Case Studies in Thermal Engineering*, vol. 36, no. 1, 2022, doi: 10.1016/j.csite.2022.102249.

[79] R. Rikwan and A. Ma'arif, "DC Motor Rotary Speed Control with Arduino UNO Based PID Control," *Control Systems and Optimization Letters*, vol. 1, vol. 1, pp. 17-31, 2023, doi: 10.59247/csol.v1i1.6.

[80] R. Shrivastava, "Digital PID Controller based Speed Control of DC Motor," *Interantional Journal of Scientific Research in Engineering And Management*, vol. 8, no. 5, pp. 1-5, 2024, doi: 10.55041/IJSREM33352.

[81] N. Prasad N and Dr. Kiran V, "I2C Master Scan Chain Insertion and Functional Coverage," *International Journal of Research and Review*, vol. 9, no. 11, pp. 54-59, 2022, doi: 10.52403/ijrr.20221108.

[82] P. F. Díaz, M. I. U. Fassler, X. Lopez, and B. Casignia, "Design and Calibration of an Arduino-Based I2C Hydraulic Flow Sensor," in *XV Multidisciplinary International Congress on Science and Technology*, pp. 181-194, 2022, doi: 10.1007/978-3-031-08280-1_13.

[83] E. J. Maevskaya, S. I. Senik, M. V. Negretskul, and D. V. Polishchuk, "Laboratory Stand Smart House On the Basis Arduino Nano Platforms Using the Sequential I2C Data Exchange Protocol," *Electrical and Computer Systems*, vol. 34, vol. 110, pp. 105-109, 2021, doi: 10.15276/eltecs.34.110.2021.11.

[84] U. Lyu and Z. Lin, "PID Control of Planar Nonlinear Uncertain Systems in the Presence of Actuator Saturation," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 1, pp. 90-98, 2022, doi: 10.1109/JAS.2021.1004281.

[85] A. A. Argaloka, H. Aptadarya, F. M. Arentaka, and F. Y. Suratman, "A Method of Anti-Windup PID Controller for a BLDC-Drive System," *Journal of Measurements Electronics Communications and Systems*, vol. 10, no. 2, p. 58, 2023, doi: 10.25124/jmecs.v10i2.7209.

[86] I. Al-Wesabi, F. Zhijian, H. M. H.n Farh, and W. Zhiguo, "Dynamic global power extraction of partially shaded PV system using a hybrid MPSO-PID with anti-windup strategy," *Engineering Applications of Artificial Intelligence*, vol. 126, p. 106965, 2023, doi: 10.1016/j.engappai.2023.106965.

[87] Z. Y. Hitit, İ. Koçer, G. Kuş, and N. Z. Arslan, "Optimal PID Control with Anti-windup in Neutralization Process," *International Advanced Researches and Engineering Journal*, vol. 7, no. 3, 2023, doi: 10.35860/iarej.1256107.

[88] M. O. Okelola, D. O. Aborisade, and P. A. Adewuyi, "Performance and Configuration Analysis of Tracking Time Anti-Windup PID Controllers," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 6, no. 2, pp. 20-29, 2020, doi: 10.26555/jiteki.v6i2.18867.

[89] B. Tomar, N. Kumar, and M. Sreejeth, "Real-Time Balancing and Position Tracking Control of 2-DOF Ball Balancer Using PID with Integral ANTI-WINDUP Controller," *Journal of Vibration Engineering & Technologies*, vol. 12, no. 1, 2023, doi: 10.1007/s42417-023-01179-x.

[90] D. Peng, B. Huang, and H. Huang, "Design of an Anti-Windup PID Algorithm for Differential Torque Steering Systems," *Shock and Vibration, Shock and Vibration*, no. 16, pp. 1-14, 2022, doi: 10.1155/2022/9973379.

[91] K. Premkumar, T. Thamizhselvan, M. V. Priya, S. R. Carter, and L. P. Sivakumar, "Fuzzy Anti-Windup PID Controlled Induction Motor," *International Journal of Engineering and Advanced Technology*, vol. 9, no.1, pp. 184-189, 2019, doi: 10.35940/ijeat.A1113.109119.

[92] A. Rios and O. L.-Santiago, "Robust PID Control with Anti-Windup Compensation for ULDS," in *Proceedings of 19th Latin American Control Congress (LACC 2022)*, pp. 23-33, 2023, doi: 10.1007/978-3-031-26361-3_3.

[93] K. K. C. Yapp, H. C. Lih, and C. H. Lai, "New anti-windup Proportional-Integral-Derivative for motor speed control," *Asian Journal of Control*, 2024, doi: 10.1002/asjc.3390.

[94] M. Filo, A. Gupta, and M. Khammash, "Anti-Windup Protection Circuits for Biomolecular Integral Controllers," *bioRxiv*, 2023, doi: 10.1101/2023.10.06.561168.

[95] M. M. Ghazaly, S. P. Tee, and N. Zainal, "Anti-windup modified proportional integral derivative controller for a rotary switched reluctance actuator," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 6, pp. 3311-3324, 2023, doi: 10.11591/eei.v12i6.6027.

[96] L. Magnani, "Heuristic Reasoning," *Studies in Applied Philosophy, Epistemology and Rational Ethics*, vol. 16, 2015, doi: 10.1007/978-3-319-09159-4.