JOIV : Int. J. Inform. Visualization, 7(3) - September 2023 991-999

INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv

433Mhz based Robot Using Proportional Integral Derivative (PID) for Precise Facing Direction

M. Amin Hariyadi [a,*], Juniardi Nur Fadila [a], Hafizzudin Sifaulloh [a]

[a] *Department of Informatics Engineering, Faculty of Science and Technology, UIN Maulana Malik Ibrahim Malang, Malang, Indonesia*
Corresponding author: [*]adyt2002@uin-malang.ac.id

***Abstract*— This research endeavor aims to evaluate the effectiveness of the robot's direction control system by employing PID (Proportional Integral Derivative) output and utilizing wireless communication LoRa E32 433MHz. The experimental robot used in this study was a tank model robot equipped with 4 channels of control. LoRa was implemented in the robot control system, in conjunction with an Android control application, through serial data communication. The LoRa E32 module system was selected based on its established reliability in long-range communication applications. However, encountered challenges included the sluggishness of data transmission when using LoRa for transferring control data and the decreased performance of the robot under Non-Line of Sight conditions. To overcome these challenges, the PID method was employed to generate control data for the robot, thereby minimizing the error associated with controlling its movements. The PID system utilized feedback from a compass sensor (HMC5883L) to evaluate the setpoint data transmitted by the user, employing Kp, Ki, and Kd in calculations to enable smooth movements toward the setpoint. The findings of this study regarding the direct control of the robot using wireless LoRa E32 communication demonstrated an error range of 0.6% to 13.34%. A trial-and-error approach for control variables determined the optimal values for Kp, Ki, and Kd as 10, 0.1, and 1.5, respectively. Future investigations can integrate additional methodologies to precisely and accurately determine the PID constants (Kp, Ki, and Kd) mathematically.**

***Keywords*— Control system; LoRa E32 433MHz; PID.**

Manuscript received 16 May. 2023; revised 14 Jun. 2023; accepted 1 Aug. 2023. Date of publication 10 Sep. 2023.

I. INTRODUCTION

The presence of robots has emerged as a cornerstone of human civilization, significantly augmenting and facilitating various human activities [1], [2]. Robots, characterized as electro-mechanical or biomechanical devices, execute anatomical movements and motions in response to specific commands[3], [4]. Unmanned ground vehicles (UGVs) hold a prominent position among the prevailing robot types utilized in contemporary times. The development of a UGV entails the integration of a microcontroller and a motor, serving as the primary propellant for the robot [5].

To establish an effective control system for the UGV robot, the design process necessitates the utilization of a suitable methodology. Numerous approaches for robot control systems can be employed encompassing PID (Proportional Integral Derivative) controllers, Fuzzy Logic, genetic algorithms, artificial neural networks, and more [6]. Among these options, PID controllers are widely applied in control system applications, owing to their proven effectiveness [7].

A PID (Proportional Integral Derivative) controller represents a control system employed to achieve precise instrumentation control, incorporating feedback characteristics within the system [8], [9]. The widespread utilization of PID control is evident in various control system applications, including the research conducted by Ma'arif on mobile robots, aiming to accurately determine their directional positioning. The findings from the study demonstrate that the mobile robots successfully moved in accordance with readings obtained from compass sensors, yielding an average success rate of 76% [10].

In the design of a UGV robot, there arises a need for a control mechanism that surpasses the efficiency of conventional joystick-based controls [5]. This requirement can be met by employing a LoRa E32 433MHz wireless module, enabling the connection of an Android device as the controller for the robot [11], [12]. The LoRa E32 433MHz module encompasses a wireless module renowned for its expansive coverage area and its inclusion in the LPWAN (Low Power Wide Area Network) network group. According

to Fadila, the coverage range facilitated by the LoRa E32 module extends beyond 2 kilometers [13].

Building upon the aforementioned contextual foundation, the researcher investigated using a PID (Proportional Integral Derivative) control scheme integrated with a LoRa E32 433MHz wireless module. This research employed a compass sensor to capture data pertaining to the robot's orientation, with the overarching objective of enhancing the control system quality of the unmanned ground vehicle (UGV) robot.

II. Materials and Method

A. Proportional Integral Derivative (PID)

The PID control system represents an instrumental approach employed for attaining precise control in instrumentation, characterized by its feedback mechanisms within the system[14]. This control system comprises three distinct parameters, namely proportional, integral, and derivative, each exhibiting unique characteristics [2]. The formulation of the PID control system is denoted by the equation where u(t) denotes the resultant value [4], [15], [16].

$$u(t) = Kp\, e(t) + Ki \int_0^t e(\mathrm{T})\mathrm{dT} + \mathrm{Kd}\, \frac{de(t)}{dt} \quad (1)$$

Notation
Kp: Proportional Gain
Ki: Integral Gain
Kd: Derivative Gain
e: error (Setpoint – Parameter Value)
t: Time (Second)
T: a variable of integration, with values between 0 to the present state T

1) Proportional: The proportional characteristic enhances the transient response during the rise and settling times, thereby mitigating steady-state errors. However, a drawback of the proportional control lies in its propensity to yield a persistent offset value increase [5], [17]. The formulation of the proportional control system can be expressed as follows:

$$\mathrm{P} = \mathrm{Kp} * \mathrm{e(t)} \quad (2)$$

The P-factor also referred to as Kp, represents a constant value employed to either decrease or amplify the proportional impact [5], [9], [10].

2) Integral: The integral characteristic of the system response exhibits a longer duration than the proportional component. As the error value increases, the signal control changes faster. The integral component possesses the property of introducing system order, which can potentially lead to instability. However, a drawback of the integral component is its tendency to produce sluggish reactions [3], [5], [9], [10], [16]. The formulation of this control system is as follows:

$$I = Ki \int_0^t e(\mathrm{T})d\mathrm{T} \quad (3)$$

The I-factor, represented by the constant value Ki, is utilized to modulate the impact of the integral component, allowing for its increase or decrease [1], [9], [10], [14].

3) Derivative: On the other hand, the derivative characteristic serves to rectify the transient response resulting from changes in the error. The magnitude of the generated signal is proportional to the rate of change of the error. Additionally, the derivative component enhances the damping effect when the system operates independently, thereby necessitating an increase in the Kp value [1], [8], [14], [15]. The formulation of this control system is as follows:

$$D = Kd\, \frac{de(t)}{dt} \quad (4)$$

The D-factor, also referred to as Kd, represents a constant value utilized to adjust the impact of the derivative component, enabling its reduction or amplification [1], [5], [8], [14].

B. LoRa E32 433MHz

LoRa E32 433MHz constitutes an integral component of the wide-area coverage network technology, which operates with low power consumption and belongs to the LPWAN (Low Power Wide Area Network) group [13], [18], [19]. The voltage requirement for LoRa E32 433MHz ranges from 2.3V to 5.5V, with caution advised as the maximum communication level of 3.3V for 5V carries a significant risk of burning [20], [21]. The module's power consumption is from 15mA to 106mA, subject to its usage requirements [22], [23]. Remarkably, the module exhibits a maximum range of up to 3 km under open area conditions (Line of Sight), facilitated by a 5dBi antenna gain and an antenna height of 2.5m [18], [20], [24]. The air data rate is also measured at 2.4kbps [25], [26]. A comprehensive table outlining the specifications of the LoRa E32 module is presented below for reference.

TABLE I
LoRa E32 specifications

Specifications	Performance			Information
	Min	Type	Max	
Operating Voltage (V)	3.3	5.0	5.2	≥ 5.0 V Output power
Communication level (V)	-	3.3	-	For 5V TTL, the maybe Risk of Burning
Operating temperature (°C)	-40	0	85	Industrial design
Maximum Tx *power* (dBm)	29.5	30	30.5	-
Receiving sensitivity (dBm)	-145	-147	-148	Air data rate is 2.4 kbps
Air data rate (bps)	0.3k	2.4k	19.2k	Control through user programming

The modulation of the LoRa spectrum involves representing each bit of payload data through the transmission of multiple chirps of information [27], [28]. A lower Spreading Factor (SF) signifies a higher frequency of chirps transmitted per second, enabling a greater amount of data to be encoded within the same time frame [29], [30]. Conversely, a higher SF corresponds to a reduced frequency of chirps per second, resulting in a lower data encoding rate. It is important to note that a higher SF necessitates a longer transmission

time, commonly referred to as "talk time" [23], [25], [30]. Prolonged talk time leads to increased operational duration for the modem, consequently resulting in higher power consumption [20].

Nevertheless, the advantage of employing a higher SF lies in the extended transmission time, allowing the receiver to sample the signal strength more frequently, enhancing overall sensitivity. Improved sensitivity enables the sensor to detect signals from greater distances, expanding the system's effective range[28], [30].

C. System Design

Based on Fig.1, the input diagram represents a control system that functions as a transceiver, facilitating communication with the controlling device via an Android mobile device. This connection is established through USB Serial Communication, where the Android device interfaces with an Arduino Nano board. The Arduino Nano board is integrated with a Transmitter Module Wireless device, enabling wireless transmission.

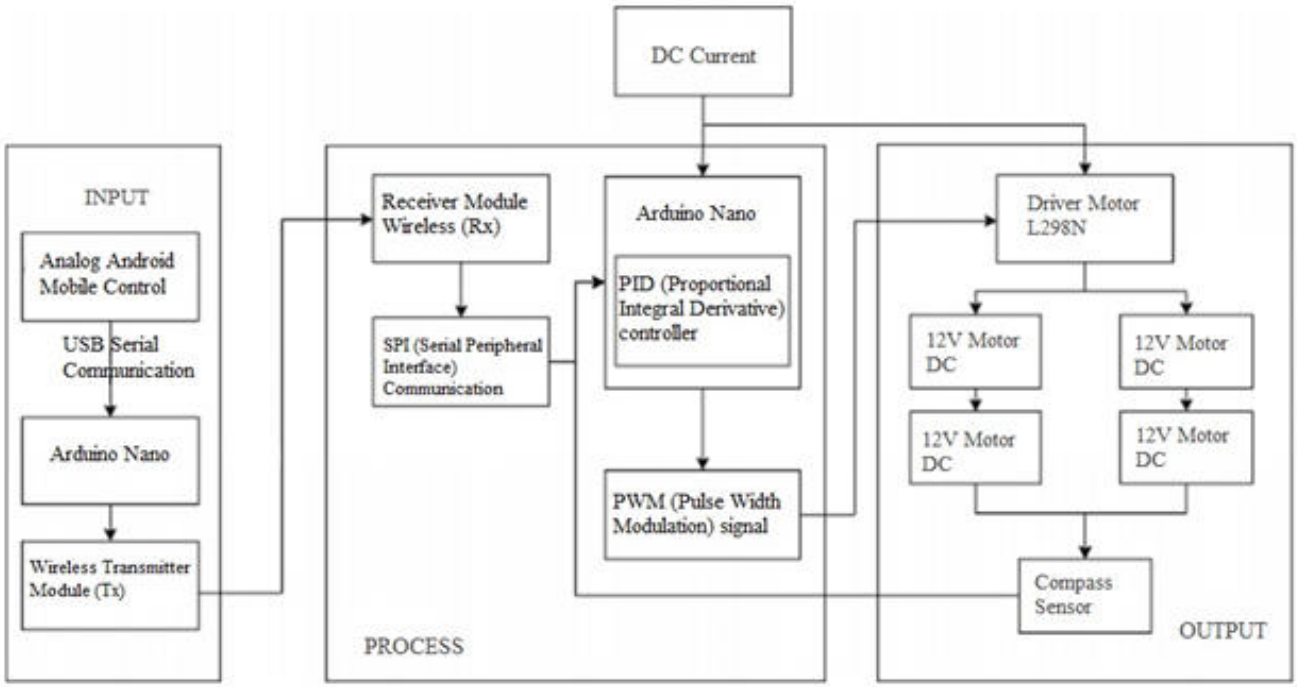


Fig. 1 System diagram of the robot

The process diagram illustrates the data flow from the transceiver module to the robot equipped with a wireless receiver module. This diagram encompasses the data-receiving process, wherein the microcontroller retrieves the data through the Serial Peripheral Interface (SPI) pin. Additionally, the diagram depicts the implementation of the Proportional Integral Derivative (PID) process. The PID process is implemented as a program syntax embedded within the microcontroller. The values of the constants Kp, Ki, and Kd are defined within this program syntax. Determining these constants involves calculations based on the previous error value, ensuring accurate and dynamic control.

Moreover, the subsequent section presents the program's implementation of the transfer function. The transfer function calculates the error, which is obtained by subtracting the setpoint value from the sensor reading. This error value is a fundamental parameter for the PID control algorithm, enabling precise control adjustments within the system.

$$error = Sensor - Setpoint \quad (5)$$

The transfer function presented in equation (5) plays a fundamental role in determining the error within the robot control system. Specifically, it seeks to quantify the discrepancy between the sensor reading and the desired setpoint value, as defined by equation (1). This transfer function serves as a crucial analytical tool, enabling a comprehensive investigation into the performance of the robot control system. By employing this transfer function, we can effectively assess the extent of deviation between the actual sensor measurements and the intended target, thereby facilitating the identification and implementation of appropriate corrective measures to enhance system accuracy and performance.

$$iterm += (Ki \text{ x } error \text{ x } dt) \quad (6)$$

As defined in equation (6), the transfer function entails the computation of the integral component within the PID parameter. Specifically, it involves the calculation of the i-term variable, which arises from the multiplication of Ki (the integral gain) with the error and time variables, in accordance with equation (3). This transfer function holds significant importance in analyzing and evaluating the PID control system. By integrating the error over time with the appropriate weighting factor, represented by Ki, this transfer function plays a pivotal role in shaping the control system's response. Consequently, we can utilize this transfer function to gain insights into the system's dynamic behavior and make informed decisions regarding adjusting integral control parameters to optimize system performance.

$$dinput = error - last\_error \quad (7)$$

The transfer function, denoted as (3.3), encompasses the computation of the derivative component within the Proportional Integral Derivative (PID) control system. Specifically, it involves the calculation of the input variable, which is derived from the difference between the current error and the previous error, as indicated by equation (4). This transfer function is pivotal in capturing the error signal's rate of change over time. By quantifying this rate of change, we can gain valuable insights into the dynamic behavior of the PID control system. This transfer function facilitates evaluating and adjusting the derivative control parameter to effectively respond to rapid changes in the error signal, thereby enhancing the control system's responsiveness and stability.

$$Result = (Kp \text{ x } error) + (iterm) + (Kd \text{ x } (dinput \; / \; dt)) \quad (8)$$

The transfer function described in equation (8) serves as a mathematical representation of the PID control system, encompassing the calculation of all PID parameters. These parameters include the iterm (Ki), obtained by multiplying the input (Kd) divided by time with Kd and Kp multiplied by the error. The derivation of these values is based on equations (1) and (2), which establish the foundational principles of the PID control algorithm.

The output chart illustrates the integration of an L298N motor driver with an Arduino nano microcontroller. This motor driver receives a Pulse Width Modulation (PWM) signal from the Arduino nano, facilitating the control of a four-motor DC system operating at a voltage of 12V. The power required for the motor driver is sourced from a battery in the form of direct current (DC). In the form of movement commands, the PWM signal precisely regulates the motor DC, dictating its specific movements according to the desired control objectives.

To determine the direction faced by the robot system, the HMC588L compass sensor plays a crucial role. This sensor detects and captures the orientation of the robot system accurately. The resulting sensor readings are transmitted back to the Arduino nano microcontroller, which is programmed with the PID control algorithm. This integration enables the Arduino Nano to utilize the sensor data effectively, allowing the PID control system to execute appropriate control actions to accurately regulate and maintain the desired direction of the robot system.

In summary, the transfer function described in equation (9) enables the calculation of PID parameters to achieve optimal control performance. Integrating the L298N motor driver, Arduino nano microcontroller, HMC588L compass sensor, and PID control algorithm forms a comprehensive control system. This system ensures precise motor control, driven by the PWM signal, and accurately maintains the desired direction of the robot system based on sensor readings.

D. Robot Circuit

Fig. 2 illustrates the schematic design of the receiver electronic circuit, showcasing the interconnections between various components. This design encompasses the relationship between the Arduino nano microcontroller and the wireless Rx module, the connection between the Arduino nano and the motor driver, as well as the association between the L298N motor driver, the motor DC, and the 12-volt DC battery.

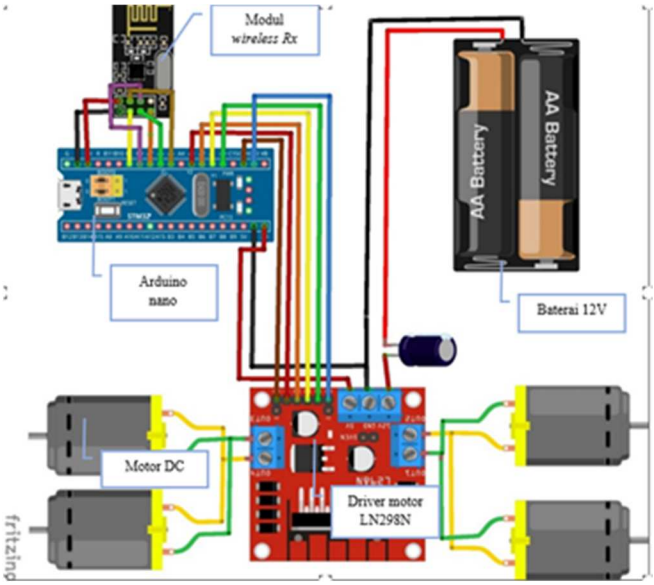


Fig. 2 Circuit receiver design

To ensure proper voltage supply, the Arduino nano derives its power from the motor driver, which provides a stable 5-volt output. On the other hand, the wireless Rx module obtains its voltage supply from the VCC pin of the Arduino nano, receiving a regulated 3.3 volts. Several pin connections link The wireless Rx module to the Arduino nano. These connections include the CE pin, which is connected to pin B1; the MOSI pin, connected to pin B0; the MISO pin, linked to pin A7; the SCK pin, connected to pin A6; and the CSN pin, which is connected to pin A5.

Furthermore, the pinout configuration of the L298N motor driver, when connected to the Arduino nano, involves specific pin assignments. The ENA pin is connected to pin C13, the IN1 pin is linked to pin A0, the IN2 pin is connected to pin A1, the IN3 pin is associated with pin A2, the IN4 pin is connected to pin A3, and the ENB pin is linked to pin C13. Overall, the schematic design depicted in Figure 3.2 showcases the intricate connections between the Arduino nano microcontroller, the wireless Rx module, and the L298N motor driver. These interconnections enable the seamless integration and control of the motor DC system, facilitating the realization of precise and efficient robotic operations.

Fig. 3 illustrates the electronic circuit design of the transceiver, showcasing the interconnections between the Arduino nano microcontroller and the wireless Tx module. In this design, the voltage supply for the Arduino nano is obtained from the connected device itself, ensuring proper power provision. On the other hand, the wireless Tx module derives its voltage supply from the VCC pin of the Arduino nano, receiving a regulated 3.3 volts.

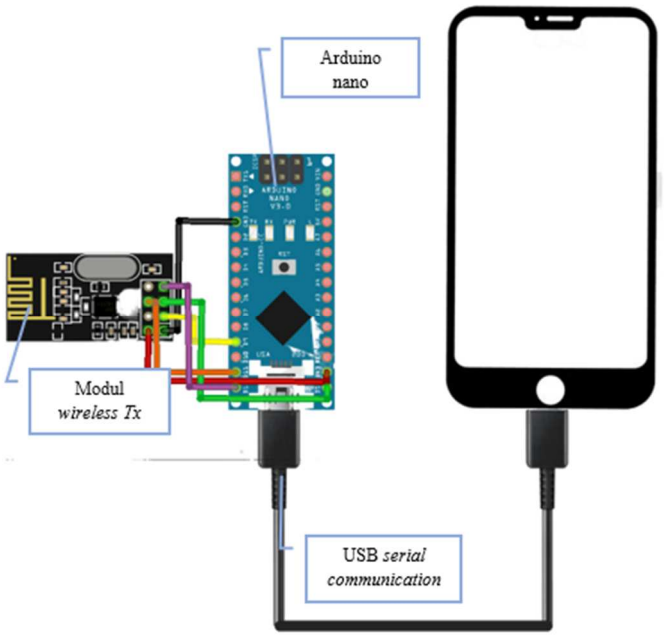


Fig. 3 Transceiver circuit design

Specific pin assignments are involved when establishing the connections between the wireless Tx module and the Arduino nano. These connections include linking the CE pin to the D7 pin, the MOSI pin to the D11 pin, the MISO pin to the D12 pin, the SCK pin to the D13 pin, and the CSN pin to the D8 pin. The meticulous design exemplifies the intricate interrelationships between the Arduino nano microcontroller and the wireless Tx module. These connections facilitate the transceiver system's seamless integration and communication capabilities, enabling efficient and reliable wireless data transmission.

III. Result And Discussions

The study's findings are illustrated in Fig. 4, depicting a series of UGV robots equipped with the LoRa E32 wireless module serving as the receiver (Rx). Subsequently, Fig. 5 showcases the transceiver (Tx) series, functioning as the control medium for the UGV robot.

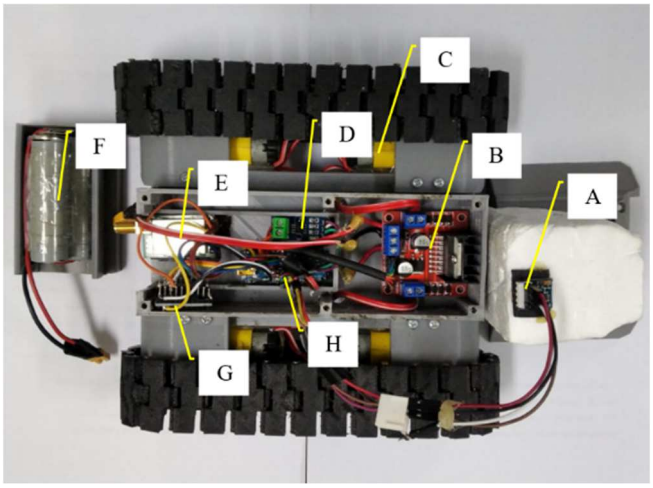


Fig. 4 LoRa E32 Rx series results

The receiver circuit (Rx) encompasses various components, including the L298N motor driver (designated as label B), Arduino nano (labeled as H), motor DC (labeled as C), GY271 compass sensor (labeled as A), ACS712 current sensor (designated as D), TXS0108E logic level (labeled as G), 12volt battery (designated as F), and LoRa E32 433MHz module (labeled as E). The study's findings are presented in Fig. 4, illustrating a series of UGV robots that have been equipped with the LoRa E32 wireless module functioning as a receiver (Rx). Fig. 5, on the other hand, displays the transceiver (Tx) series, which serves as the control medium for the UGV robot.

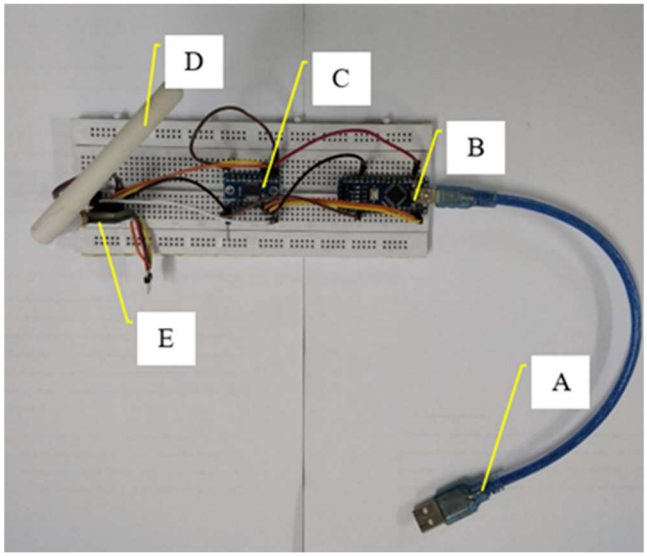


Fig. 5 Tx Circuit results with LoRa E32

The transceiver (Tx) circuit comprises several components, including USB serial communication (labeled as A), logic level TXS0108E (labeled as C), 433MHz antenna (labeled as D), Arduino nano (labeled as B), and LoRa E32 wireless module (labeled as E).

A. Wireless Module Testing

Wireless module testing was performed to compare the performance of LoRa E32 with NRF24 under two different conditions: Line of Sight (LOS) and Non-Line of Sight (NLOS). The testing involved evaluating the wireless modules' distance coverage, power consumption, and transmission speed. The error formula, which captures the discrepancies between the test results of NRF24 and LoRa E32, is presented below [16].

$$E = \left| \frac{(Mx - My)}{My} \, x \, 100\% \right| \quad (9)$$

Notation
E : Error (Absolute Value)
Mx: NRF24 Test Results
My: LoRa E32 Test Results
The following is a table of the results of the wireless module test.

TABLE II
TESTING ERRORS WHEN LOS (LINE OF SIGHT)

Distance Test (Meter)	Power Consumption x̄(mA)		Error (%)	Transmission Speed x̄ (Microseconds)		Error (%)
	NRF24 Test	LoRa E32 Test		NRF24 Test	LoRa E32 Test	
100	180,13	189,52	4,95	10131,1	132081	92,32
200	184,36	3435,66	94,63	10647,6	132810	91,98
350	201,25	4866,15	95,86	14030,4	145243	90,34
500	-	5469,21	-	-	121240	-
1 K	-	5338,008	-	-	182320	-
1,5 K	-	6933,93	-	-	281909	-

The test results presented in Table II pertain to the Line of Sight (LOS) conditions. In terms of transmission speed, the NRF24 module outperforms the LoRa E32 module, exhibiting an average error percentage of 92.32% at a distance of 100 meters, 91.98% at 200 meters, and 90.34% at 350 meters. On the other hand, the power consumption of the LoRa E32 module is nearly identical to that of the NRF24 module within a 100-meter range, with an average error percentage of 4.95%. However, notable differences arise over longer distances. At a distance of 200 meters, the LoRa E32 module demonstrates a significant average error percentage of 94.63%; at 350 meters, it reaches 95.86%. Notably, the LoRa E32 module surpasses the NRF24 module in terms of maximum coverage, with a range of up to 1.5 km compared to the NRF24's maximum range of only 350 meters.

Table III presents the test results corresponding to the Non-Line of Sight (NLOS) conditions. In this scenario, the LoRa E32 module demonstrates superior performance, achieving a distance of up to 411 meters compared to the NRF24 module's limited reach of 87 meters. However, the NRF24 module outperforms the LoRa E32 module when considering transmission speed. At a distance of 87 meters, the NRF24 module exhibits a lower average error percentage of 83.05%. It is worth noting that despite the difference in transmission speed, both modules consume nearly the same amount of power at this distance. The LoRa E32 module operates with an average error percentage of 11%, comparable to that of the NRF24 module.

TABLE IIII
TESTING ERRORS WHEN NLOS (NON-LINE OF SIGHT)

Distance Test (Meter)	Power Consumption x̄(mA)		Error (%)	Transmission Speed x̄ (Microseconds)		Error (%)
	NRF24 Test	LoRa E32 Test		NRF24 Test	LoRa E32 Test	
87	184,36	164,07	11	14818	87459,45	83,05
235	-	5721,55	-	-	128216,21	-
411	-	8151,83	-	-	211243,24	-

B. PID testing

The PID (Proportional Integral Derivative) testing focuses on evaluating the orientation of the robot's face. During the testing process, the level of error values originating from the direction of the UGV robot is measured. The trial-and-error method is employed to determine the optimal constant values for the PID parameters. This iterative process enables the derivation of the formula that represents the facing error [16].

$$E = |\frac{(Cx - Sp)}{Sp} \text{ x } 100\%| \quad (10)$$

Notation
E : Error (Absolute Value)
Cx: NRF24 Test Results
Sp: Setpoint

The obtained results of this test include a comprehensive comparison of various control combinations, namely P (Proportional), PI (Proportional Integral), PD (Proportional Derivative), and PID (Proportional Integral Derivative) controls, all of which have been implemented within the program. These combinations were thoroughly evaluated and analyzed to assess their respective performance and effectiveness.

1) Using the P (Proportional) controller

Fig. 6 illustrates the graph obtained from the experiment using the Kp value of 2, which shows a very poor control response and a high number of steady-state errors. Hence, an additional Kp value is necessary. Subsequently, employing a Kp value of 6 yields desirable graphic results from the control system using the P control. However, the response is still slow.

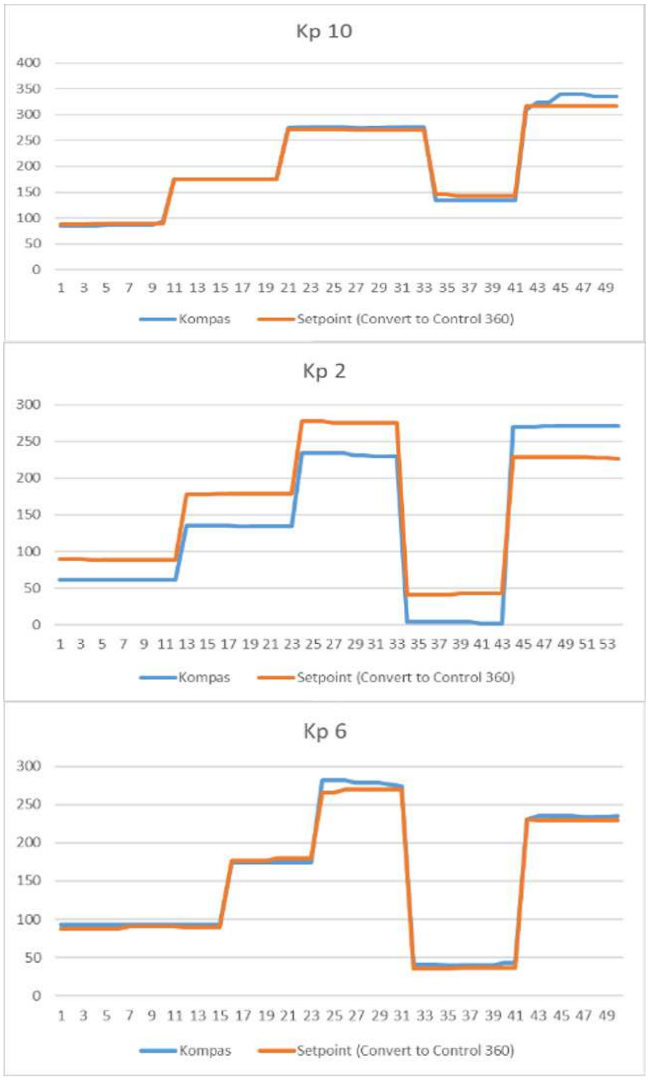


Fig. 6 Test graph using Kp

On the other hand, the graphic results obtained using the Kp value of 10 results in a fast response, but there is an overshot that may affect the control system.

2) Using a Proportional Integral (PI) controller

This PI test aims to mitigate the steady-state error present in the system, as evidenced in Fig. 7. By utilizing the values of Kp 10 and Ki 0.3. There is a noticeable difference between the compass sensor readings and the setpoint (converted to 360). Conversely, the graphic results obtained using the Kp 10 and Ki 0.2 values yield a difference that is not significantly deviating from the setpoint. In this scenario, the Ki value in the PI combination is appropriately reduced, thereby effectively reducing the presence of steady-state error. Conversely, in the graphic results obtained using Kp 10 and Ki 2, the Ki values are excessively added, leading to irregular control of the robot.

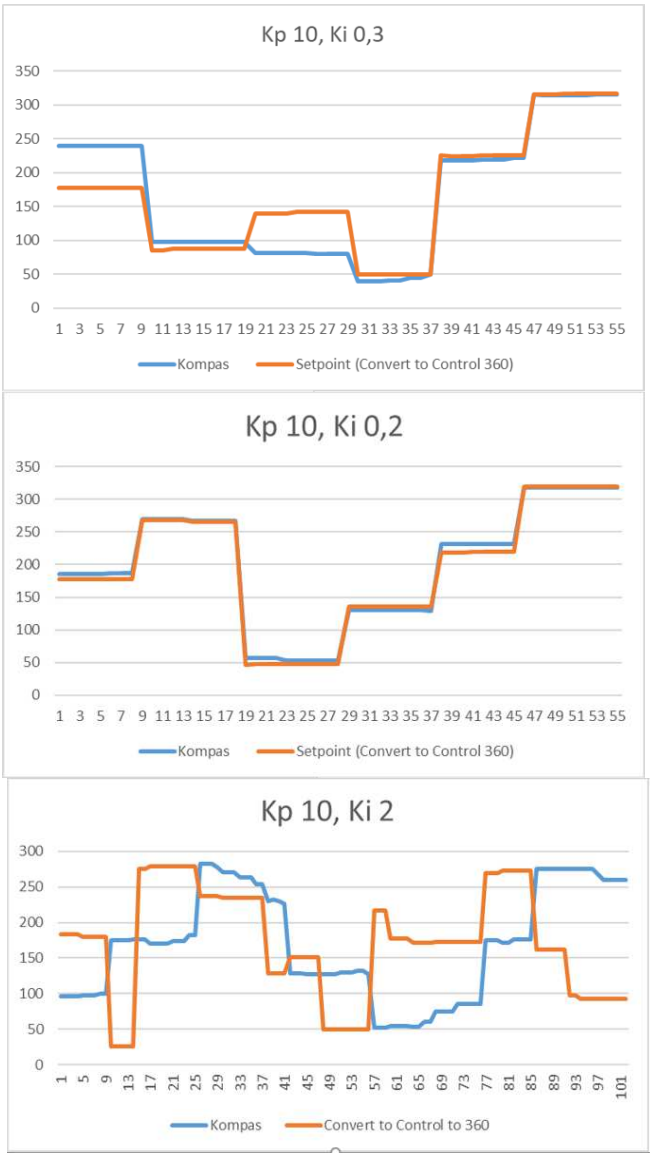


Fig. 7 Test graphs using a combination of Kp and Ki

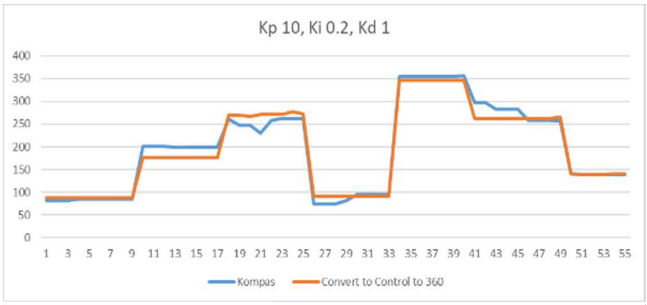
3) Using a PD (Proportional Derivative) controller

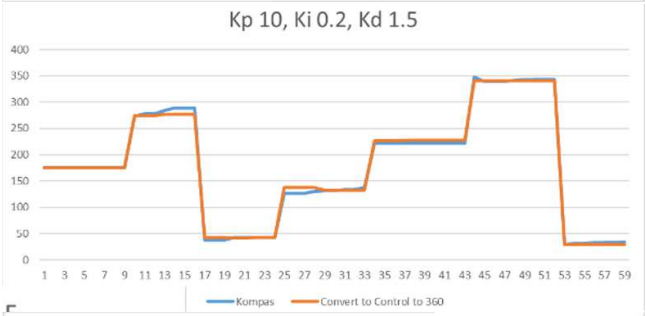
The graph depicted in Fig. 8 illustrates the results of utilizing the P constant in combination with D, indicating a decrease in the overshot value produced by the magnitude of the Kp value. The Kd value test was carried out by assigning a value of 1, as shown in Fig. 4.5, where there is still a slight overshot value. Consequently, a trial was conducted with a Kd value of 1.5, which produced superior results. However, the charts using Kp 10 and Kd 0.8 yielded unsatisfactory results due to excessive overshoot, indicating the unsuitability of the Kd value of 0.8.

Fig. 8 Test graphs using a combination of Kp and Kd

4) Using a PID (Proportional Integral Derivative) controller

The test results employing all values of the PID constants demonstrate a favorable graph when utilizing Kp 10, Ki 0.2, or 0.1, and Kd 1.5, as illustrated in Fig. 9. This combination yields a rapid response, eliminates overshoot, and mitigates steady-state error in robot control. Conversely, employing Kp 10, Ki 0, and Kd 1.25 results in an unfavorable graph characterized by the persistence of overshoot and steady-state errors.





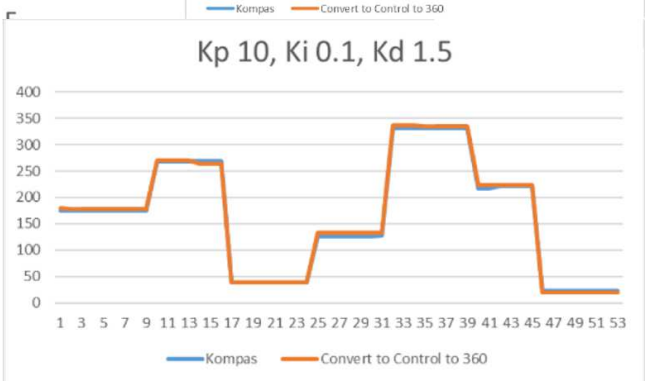


Fig. 9 Test graphs using a combination of Kp, Ki, and Kd

5) Testing without PID

Testing without PID control involved manually programming the robot's direction without utilizing the PID or other algorithms. The test aimed to determine the error values from various facing directions, including 175°, 275°, 42°, 135°, 230°, and 30°. The obtained data was then used to calculate the average and determine the error percentage through relative error calculations. Table 4.3 presents the results obtained from the test conducted without PID control. The tests' results without PID control exhibited inferior graphics, as exemplified by the graph in Fig. 10. This graph depicted a sluggish response, excessive overshot, and significant steady-state errors in robot control.

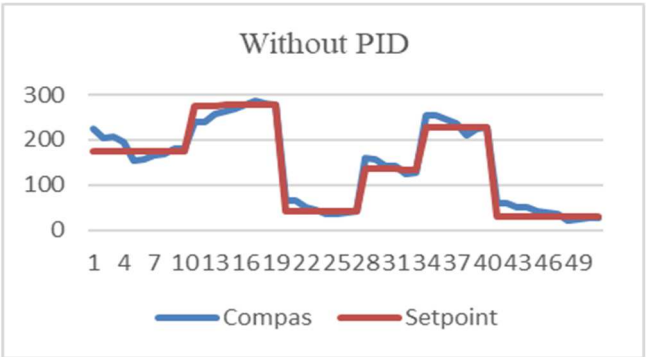


Fig. 10 Test graph without PID

IV. Conclusion

Based on the tools and experiments conducted, the following conclusions can be drawn from the test results for wireless modules show that when in LOS (Line of Sight) conditions, the NRF24 wireless module is more efficient regarding power consumption. Additionally, the transmission speed of the NRF24 module is superior to that of the LoRa E32 module. The test results for NLOS (Non-Line of Sight) conditions were similar to LOS conditions, as the power consumption and transmission speed continued to increase as the distance between wireless modules increased. However, obstacles, or NLOS conditions, can affect the maximum distance of each wireless module. The NRF24 module has a maximum distance of 80 meters, while the LoRa E32 module can reach up to 411 meters. The use of PID controllers in robot control systems is highly accurate, with an error percentage of 0.57% to 12.51% when using Kp 10, Ki 0.2, and Kd 1.5. This combination also reduces overshoot and steady-state error.

However, using the P controller with Kp 10 results in an error percentage of 1% to 7.5% with some overshot. Furthermore, the use of a combination of PI controllers with Kp 10 and Ki 0.2 produces an error percentage of 0.18% to 19% but with a slight steady-state error. Using a combination of PD controllers with Kp 10 and Kd 1.5 results in an error percentage of 0.06% to 8.34%, but there is still a slight overshoot. In contrast, testing without PID control resulted in less accurate robot movements, with an error percentage ranging from 13.29% to 49.22%, and indirect robot movements with a high error rate.

This research demonstrates that PID control can enhance the robot's facing direction system. The LoRa control system exhibits exemplary performance when integrated with PID, particularly over longer distances, rendering the robot more resilient and stable. However, achieving perfection requires further investigation, such as exploring obstacle avoidance strategies based on PID control and developing an efficient speed and battery management system. The latter is crucial for extended-distance control, as it necessitates a prolonged battery lifetime.

Acknowledgment

This research was supported by the Ministry of Religion of Indonesia, the Research and Community Service Board (LP2M), and the Faculty of Sains and Technology UIN Maulana Malik Ibrahim Malang.

References

[1] A. Latif, K. Shankar, and P. T. Nguyen, "Legged fire fighter robot movement using PID," *Journal of Robotics and Control (JRC)*, vol. 1, no. 1, 2020, doi: 10.18196/jrc.1104.

[2] A. Habibi and R. T. S. Haryati, "Artificial Intellegence in Nursing : A Literatur Riview," *JKFT: Universitas Muhamadiyah Tangerang Vol.*, vol. 6, no. 2, 2021.

[3] P. Huang, Z. Zhang, and X. Luo, "Feedforward-plus-proportional–integral–derivative controller for agricultural robot turning in headland," *Int J Adv Robot Syst*, vol. 17, no. 1, 2020, doi: 10.1177/1729881419897678.

[4] I. Barus, D. A. Prasetya, and N. Rochman, "Optimasi Realtime Control System Pada Navigasi Mobile Robot," *Prosiding SNATIF*, 2017.

[5] W. Purbowaskito and C.-H. Hsu, "Sistem Kendali PID untuk Pengendalian Kecepatan Motor Penggerak Unmanned Ground Vehicle untuk Aplikasi Industri Pertanian," *Jurnal Infotel*, vol. 9, no. 4, 2017.

[6] E. A. Chater, H. Housny, and H. El Fadil, "Adaptive proportional integral derivative deep feedforward network for quadrotor trajectory-tracking flight control," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 4, 2022, doi: 10.11591/ijece.v12i4.pp3607-3619.

[7] M. Lamatenggo, I. Wiranto, and W. Ridwan, "Perancangan Balancing Robot Beroda Dua Dengan Metode Pengendali PID Berbasis Arduino Nano," *Jambura Journal of Electrical and Electronics Engineering*, vol. 2, no. 2, 2020, doi: 10.37905/jjeee.v2i2.6906.

[8] S. J. Hammoodi, K. S. Flayyih, and A. R. Hamad, "Design and implementation speed control system of DC motor based on PID control and matlab simulink," *International Journal of Power Electronics and Drive Systems*, vol. 11, no. 1, 2020, doi: 10.11591/ijpeds.v11.i1.pp127-134.

[9] I. P. Adinata, M. Pratama, I. N. Suweden, and I. B. A. Swamardika, "Sistem Kontrol Pergerakan Pada Robot Line Follower Berbasis Hybrid PID-Fuzzy Logic," *Prosiding Conference on Smart-Green Technology in Electrical and Information Systems*, no. November, 2013.

[10] R. Dikairono, T. A. Sardjono, and L. Yulianto, "Sistem Navigasi Dan Penghindar Rintangan Pada Mobile Robot Menggunakan Gps Dan Pengukur Jarak Ultrasonik," *JAVA Journal of Electrical and Electronics Engineering*, vol. 11, no. 1, 2016.

[11] J. M. Sole, R. P. Centelles, F. Freitag, and R. Meseguer, "Implementation of a LoRa Mesh Library," *IEEE Access*, vol. 10, 2022, doi: 10.1109/ACCESS.2022.3217215.

[12] A. S. Milewski, Ł. Mierzejewski, and J. Tołstoj-Sienkiewicz, "Differential control of six-wheeled robot using a mobile application," in *Solid State Phenomena*, 2017. doi: 10.4028/www.scientific.net/SSP.260.45.

[13] J. N. Fadila, "Improving UAV Radio Control System with 433 MHz Radio Wave Using Lo-Ra based on QCZEK Model Communication System," *MATICS*, vol. 14, no. 1, 2022, doi: 10.18860/mat.v14i1.16370.

[14] H. Maghfiroh, M. Ahmad, A. Ramelan, and F. Adriyanto, "Fuzzy-PID in BLDC Motor Speed Control Using MATLAB/Simulink," *Journal of Robotics and Control (JRC)*, vol. 3, no. 1, 2022, doi: 10.18196/jrc.v3i1.10964.

[15] A. Riansyah, S. Mulyono, and M. Roichani, "Applying fuzzy proportional integral derivative on internet of things for figs greenhouse," *IAES International Journal of Artificial Intelligence*, vol. 10, no. 3, pp. 536–544, Sep. 2021, doi: 10.11591/ijai.v10.i3.pp536-544.

[16] A. M. Abed *et al.*, "Trajectory tracking of differential drive mobile robots using fractional-order proportional-integral-derivative controller design tuned by an enhanced fruit fly optimization," *Measurement and Control (United Kingdom)*, vol. 55, no. 3–4, 2022, doi: 10.1177/00202940221092134.

[17] M. Fliess and C. Join, "An alternative to proportional-integral and proportional-integral-derivative regulators: Intelligent proportional-derivative regulators," *International Journal of Robust and Nonlinear Control*, vol. 32, no. 18, 2022, doi: 10.1002/rnc.5657.

[18] O. A. Saraereh, A. Alsaraira, I. Khan, and P. Uthansakul, "Performance evaluation of UAV-enabled LoRa networks for disaster management applications," *Sensors (Switzerland)*, vol. 20, no. 8, 2020, doi: 10.3390/s20082396.

[19] J. P. Shanmuga Sundaram, W. Du, and Z. Zhao, "A Survey on LoRa Networking: Research Problems, Current Solutions, and Open Issues," *IEEE Communications Surveys and Tutorials*, vol. 22, no. 1, 2020, doi: 10.1109/COMST.2019.2949598.

[20] R. Liang, L. Zhao, and P. Wang, "Performance evaluations of lora wireless communication in building environments," *Sensors (Switzerland)*, vol. 20, no. 14, 2020, doi: 10.3390/s20143828.

[21] C. Bouras, A. Gkamas, and S. A. K. Salgado, "Energy efficient mechanism for LoRa networks," *Internet of Things (Netherlands)*, vol. 13, 2021, doi: 10.1016/j.iot.2021.100360.

[22] R. Islam, M. W. Rahman, R. Rubaiat, M. M. Hasan, M. M. Reza, and M. M. Rahman, "LoRa and server-based home automation using the internet of things (IoT)," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, 2022, doi: 10.1016/j.jksuci.2020.12.020.

[23] A. J. Wixted, P. Kinnaird, H. Larijani, A. Tait, A. Ahmadinia, and N. Strachan, "Evaluation of LoRa and LoRaWAN for wireless sensor networks," in *Proceedings of IEEE Sensors*, 2017. doi: 10.1109/ICSENS.2016.7808712.

[24] Q. Zhou, K. Zheng, L. Hou, J. Xing, and R. Xu, "Design and implementation of open LORa for IoT," *IEEE Access*, vol. 7, 2019, doi: 10.1109/ACCESS.2019.2930243.

[25] A. Augustin, J. Yi, T. Clausen, and W. M. Townsley, "A study of Lora: Long range & low power networks for the internet of things," *Sensors (Switzerland)*, vol. 16, no. 9, 2016, doi: 10.3390/s16091466.

[26] T. Antoine-Santoni, B. Poggi, D. Araujo, and C. Babatounde, "Factors Influencing LoRa Communication in IoT Deployment: Overview and Experience Analysis," 2022. doi: 10.5220/0011102600003194.

[27] R. Anzum *et al.*, "A Multiwall Path-Loss Prediction Model Using 433 MHz LoRa-WAN Frequency to Characterize Foliage's Influence in a Malaysian Palm Oil Plantation Environment," *Sensors*, vol. 22, no. 14, 2022, doi: 10.3390/s22145397.

[28] L. Moiroux-Arvis, C. Cariou, and J. P. Chanet, "Evaluation of LoRa technology in 433-MHz and 868-MHz for underground to aboveground data transmission," *Comput Electron Agric*, vol. 194, 2022, doi: 10.1016/j.compag.2022.106770.

[29] Hudiono, M. Taufik, R. H. Y. Perdana, and A. E. Rakhmania, "Digital centralized water meter using 433 mhz lora," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 4, 2021, doi: 10.11591/EEI.V10I4.2950.

[30] M. O. Ojo, D. Adami, and S. Giordano, "Experimental evaluation of a Lora wildlife monitoring network in a forest vegetation area," *Future Internet*, vol. 13, no. 5, 2021, doi: 10.3390/fi13050115.