Implementation of PID Adaptive Control System for Hexapod Robot Orientation

Yoga Adriansyah[2], Senanjung Prayoga[1*], Anugerah Wibisana[1]

[1]Jurusan Teknik Elektro, Prodi Teknik Robotika, Politeknik Negeri Batam, Batam, Indonesia
[2]Jurusan Teknik Elektro, Prodi Teknik Mekatronika, Politeknik Negeri Batam, Batam, Indonesia

[*]Email: senanjung@polibatam.ac.id

Received on 17-01-2025 | Revised on 01-04-2025 | Accepted on 25-06-2025

***Abstract*—This research aims to improve the stability and accuracy of hexapod robot motion in the SAR competition in Indonesia using an adaptive PID control system combined with fuzzy logic. PID control has advantages in linear systems but is less responsive to dynamic conditions, while fuzzy logic can produce a more flexible adaptive response. The proposed system uses IMU sensors to measure directional error, where this data is then processed by fuzzy control to adjust the PID parameters in real time. Tests show that this method results in an average recovery time of 3,06 seconds, a rise time of 1,99 seconds, and an overshoot of approximately 10,00% to 15,60%. These results show an increase in the efficiency of the robot in navigating in complex environments, thus improving the performance in real applications.**

Keywords: Adaptive PID Control, Fuzzy Logic, Hexapod Robot, IMU Sensor.

I. INTRODUCTION

INDONESIA SAR Robot Contest is a search and rescue legged autonomous robot. In last year there were new challenges that had to be overcome, such as broken roads, debris roads, muddy roads, sloping roads, and stairs. SAR robots must move quickly to save victims, overcome obstacles and move alone without an operator [1].

The lack of a control system to assist the robot in rescue operations causes the robot to move very slowly and less efficiently. Without an adequate control system, the robot will not be able to react quickly and appropriately to situations, thus reducing the effectiveness of rescue operations. In addition, the process of determining the robot's facing direction takes a considerable amount of time, which further slows down the Robot overall performance in critical situations [2].

The goal of this research is to implement and test an adaptive PID-based control system using an Inertia Measuring Unit (IMU) sensor to quickly and accurately improve robot motion. PID is known for its ease of implementation and reliability in controlling linear systems [3]. However, PID has limitations in responding to changing dynamic conditions in nonlinear systems, especially when disturbances or unexpected changes in system parameters occur [4], [5], like the previous research [6] use compass sensor to improve robot motion, the result is sensor giving 2-5% error, and PID result have little more time to control rotation of the robot.

Due to limitations, fuzzy logic presents a viable solution for enhancing control system performance by providing flexible and adaptive responses, especially in scenarios where conventional PID tuning methods may fall short [7]. Fuzzy logic has given good results in previous studies have applied control algorithms to fixed maze solved [8]. In contrast, this work focuses on a mobile robot where the inverse kinematics are continuously adjusted as the robot navigates through dynamic environments [9], [10]. This presents unique challenges and opportunities for real-time processing and control. In this research, fuzzy logic is integrated with PID control to optimize motion precision and stability. In addition, adaptive control strategies are implemented to manage variations in load and velocity, ensuring smoother and more accurate movements [11]. The proposed system is expected to increase the robot's efficiency in complex tasks and improve overall performance in real-world applications.

II. Method

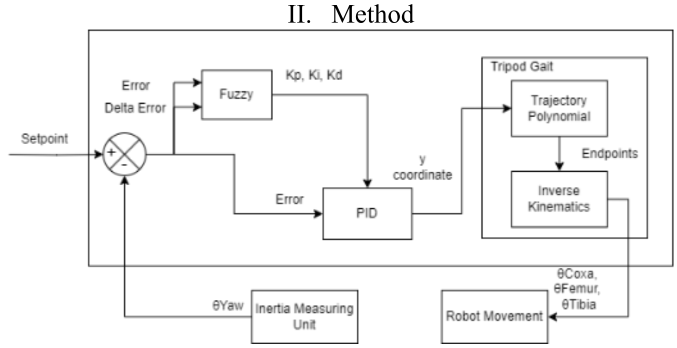

Figure 1. System Control Diagram

Figure 1 shows the block diagram of the control system design implemented for the robot using fuzzy logic control. Initially, the setpoint is introduced into the function, fuzzy get error and delta error are derived from the inertia measuring unit (IMU). These error values and the delta error are processed through fuzzy logic to generate PID parameters. The PID controller calculates the parameters based on the fuzzy output and the IMU error, which then produces the Y coordinate. This Y coordinate is then used as the input to a polynomial function, which produces the end points. Inverse kinematics is used to determine these end points, allowing the robot to move. The details of the system control design are discussed in the following sections.

A. Fuzzy

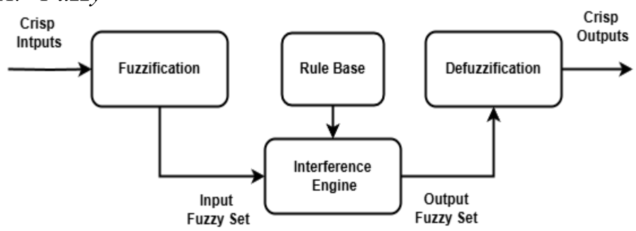

Figure 2. Fuzzy logic diagram

Figure 2 shows the fuzzy logic diagram, the input from IMU (error and delta error) going through fuzzification where the process of converting a non-fuzzy (crisp) set into a fuzzy set. Rule base from fuzzy is constructed, which connects the input variables with the output variable by means of if-then rules [12]. Rule base this research is show at figure below.

L = Low
M = Medium
H = High

		Error		
		L	M	H
Delta Error	L	L	M	L
	M	M	M	H
	H	H	H	L

Figure 3. Rules base fuzzy logic

Figure 3 shows the rules base for this research, these rules based on membership function that show at below.

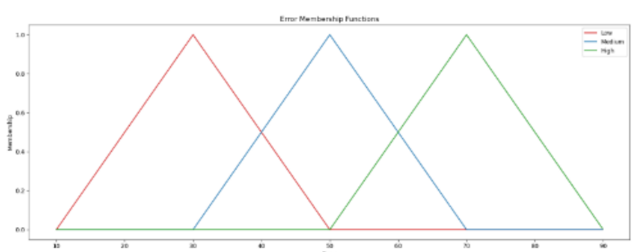

Figure 4. Input membership *error*

Figure 4 show input membership for error, the input has three level for producing different value to process defuzzification and producing the crisp out fuzzy.

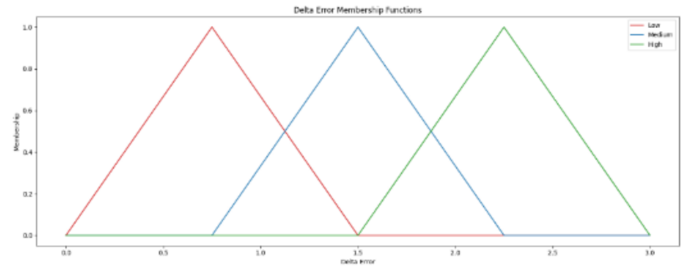

Figure 5. Input membership delta *error*

Figure 5 show input membership for delta error, delta error is the sum difference from error, has three level to producing the crisp out.

B. PID

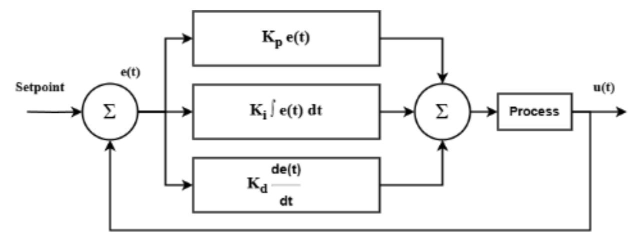

Figure 6. PID diagram

Figure 6 show how PID diagram works. PID control is a control method that uses three main elements, namely proportional controllers (P), integral controllers (I), and derivative controllers (D). Each of these controllers has its own advantages, such as proportional controllers (P) speeding up the rise time, integral controllers (I) minimizing the error generated by the system, and derivative controllers (D) reducing the overshoot or undershoot [13]. The PID equation that is obtained is as follows

$$u(t) = K_p\, e(t) + K_i \int_0^t et\, dt + K_d \frac{de(t)}{dt} \tag{1}$$

Where the symbol is

- $\boldsymbol{u(t)}$: PID controller total output
- $\boldsymbol{K_p}$: Gain Proportional
- $\boldsymbol{K_i}$: Gain Integral
- $\boldsymbol{K_d}$: Gain Derivative
- $\boldsymbol{e(t)}$: Error
- $\boldsymbol{\int_0^t et\, dt}$: Integral of error
- $\boldsymbol{\frac{de(t)}{dt}}$: Error derivation

C. Tripod Gait

The gait pattern of a hexapod robot has more variations compared to a quadruped robot because it can combine single or paired leg movements with more combinations. Each leg movement can be divided into two main phases: the support phase, where the leg provides thrust to the robot body while remaining attached to the ground, and the transfer phase, where the leg is lifted off the ground and swings forward to start the next phase [14]. The tripod step pattern has a fast phase to move between the support phase and the transfer phase.

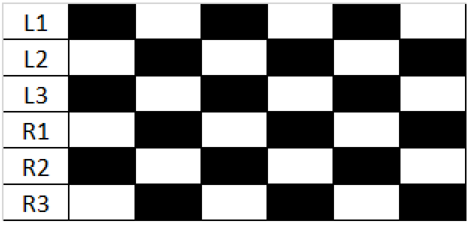

Figure 7. Gait cycle

Figure 7 show illustrates where the black color is which leg move. At that figure L1, L3, and R2 transfer phase. L2, R1, and R3 is support phase to help robot stable on the ground.

D. *Trajectory Polynomial*

The trajectory of the robot leg is the path of motion from one endpoint to another. This trajectory is needed so that the change from one point to another is the same and the movement of the robot is smoother.

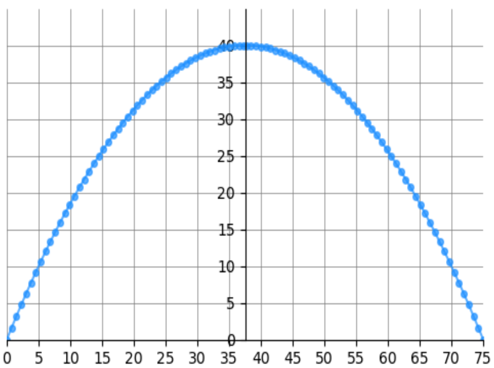

Figure 8. Polynomial curve

Figure 8 illustrate steps in legged robots from right figure to left figure. This called trajectory where is used to adjust the steps on the robot's legs. The trajectory used in the steps is a parabolic curve show at resulting from a polynomial equation. This polynomial equation produces trajectory points that are used as the trajectory of the robot's footsteps.

$$P(t)_{x,y,z} = (1-t)^3 P1_{x,y,z} + 3t(1-t)^2 P2_{x,y,z} + 3t^2(1-t) P3_{x,y,z} + t^3 P4_{x,y,z} \tag{2}$$

Where the P is the desired end point at iteration t, and t is the iteration factor. P1 through P4 are reference vector points on the x, y, and z coordinate axes. These points define the resulting trajectory. The value of t is the iteration factor, which increases from zero to one with time. The value of t is increased by 0.1 or as needed.

E. *Inverse Kinematics*

The leg of a hexapod robot has three main parts: the coxa (hip), the femur (thigh), and the tibia (shin). The coxa supports the robot's body weight during both static and dynamic motion. The femur connects the coxa to the tibia, forming a hinge joint that allows for smooth, stable walking. Together, these structures provide the leg with stability and flexibility for different tasks and environments.

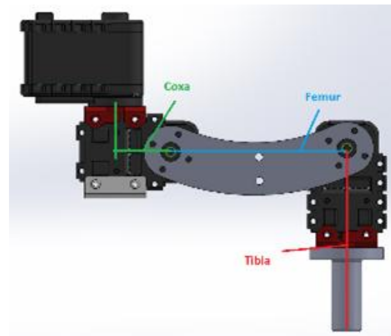

Figure 9. Robot leg structure

Figure 9 show how Coxa, Femur and Tibia assembly as Structure Leg. To determine the angles of the coxa, femur, and tibia joints on a hexapod robot so that the tip of the leg reaches the desired position, inverse kinematics is used [15]. This allows for precise determination of these angles to achieve the final position goal with accuracy and efficiency.

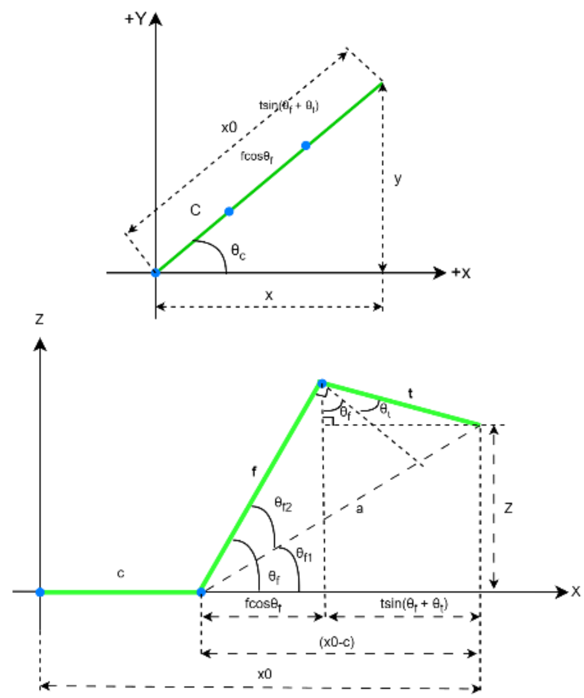

Figure 10. Viewing angle of joints in the XY and XZ plane

Figure 10 illustrate leg structure in XY and XZ plane. The values for the coxa angle (θc) and femur angle (θf) can be determined, while the tibia angle (θt) and leg span can be calculated using the following equations for the angles.

$$\theta coxa = tan^{-1}\left(\frac{y}{x}\right) \quad (3)$$

$$x0 = \sqrt{x^2 + y^2} \quad (4)$$

$$a = \sqrt{z^2 + (x0 - c)^2} \quad (5)$$

$$\theta f1 = tan^{-1}\left(\frac{z}{(x0 - c)}\right) \quad (6)$$

$$\theta f2 = cos^{-1}\left(\frac{f^2 + a^2 + t^2}{2\,a\,f}\right) \quad (7)$$

$$\theta femur = \theta f1 + \theta f2 \quad (8)$$

$$\theta tibia = cos^{-1}\left(\frac{f^2 + t^2 + a^2}{2\,f\,t}\right) - 90° \quad (9)$$

F. Hexapod Robot

Figure 11. Hexapod robot

Figure 11 show the robot used in this study is a six-legged robot. It is equipped with a camera, the Huskylens. In addition, the robot is also equipped with a claw system and proximity sensors. The claw system uses a gripper mechanism while the wall distance sensor uses four Time of Flight (ToF) sensors.

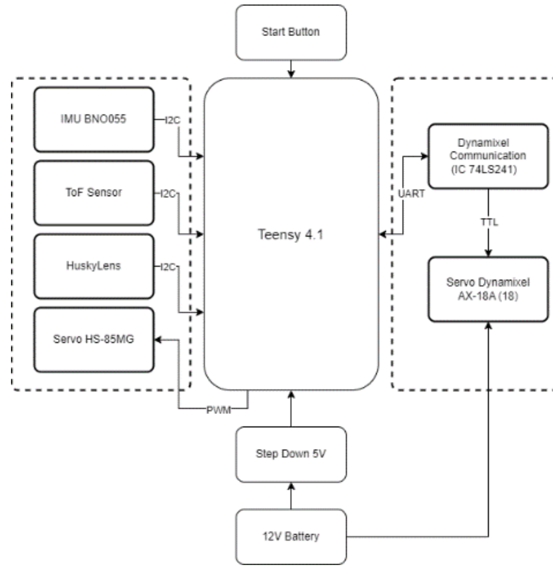

Figure 12. Hardware block diagram

Figure 12 show robot diagram system that use IMU BNO055 as Inertia Measurement Unit for this robot with I2C protocol.

III. Result

A. Angle testing of the IMU sensor

The robot is placed on a protractor with a degree sequence of 0° to 90°, the robot is manually rotated per 10° based on the protractor starting from 0° to 90° and the other side from - 0° to -90°.

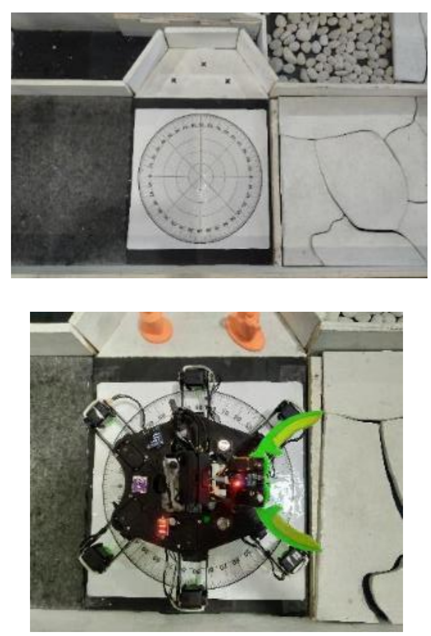

Figure 13. Sensor testing

Figure 13 show how testing sensor on IMU with placed in robot. Here are the results of Sensor testing.

TABLE I
IMU TESTING SENSOR

Degree	Experiment			Mean	Highest Deviation
	1	2	3		
-90	-89,62	-90,19	-90,06	-89,96	0,35
-80	-80,24	-79,94	-80,00	-80,04	0,20
-70	-69,69	-70,37	-69,50	-69,89	0,48
-60	-59,69	-60,25	-60,00	-59,98	0,30
-50	-50,19	-50,69	-49,94	-50,20	0,49
-40	-39,38	-40,56	-40,06	-40,00	0,62
-30	-30,12	-30,75	-29,81	-30,17	0,58
-20	-19,87	-20,50	-20,06	-20,10	0,40
-10	-10,56	-10,14	-9,81	-10,13	0,43
0	0,06	0,06	0,06	0,05	0,02
10	9,56	10,19	10,44	10,05	0,49
20	20,06	19,62	20,44	20,03	0,41
30	30,19	29,56	30,50	30,06	0,51
40	40,02	39,94	40,05	40,01	0,06
50	50,44	49,88	50,24	50,14	0,30
60	60,19	59,69	60,50	60,10	0,41
70	70,12	69,81	70,19	70,03	0,22
80	79,87	80,06	80,75	80,17	0,58
90	90,81	89,31	90,62	90,19	0,88

From the inspection of the IMU sensor readings shown in table 1, it can be seen that the average sensor reading of each angle has a difference of 0,01 to 0,1 degrees from the actual value on the protractor reading. In the Highest Deviation column, the largest deviation values are within a relatively small range, indicating that the measurement results are fairly consistent. As seen in the table, the largest measured deviation ranges from 0,02 to 0,62.

Once applied to the robot, these measurement differences did not have a significant effect on the robot's facing direction. Therefore, the authors decided not to apply additional filtering to correct the difference deviation, as the resulting accuracy was sufficient for the needs of controlling the robot's direction in the context of the tested application.

B. *Adaptive PID Testing*

This PID test is carried out by combining other algorithms, namely Fuzzy, the logic will function to adjust the value of the PID constant automatically, then the response will be seen to meet the values in the table below. The robot will be placed in room 1 in front of obstacle 1 (broken road), then the robot will rotate before passing obstacle 1 (broken road), then the author will see the response of the PID and the robot whether it can pass the obstacle or not.

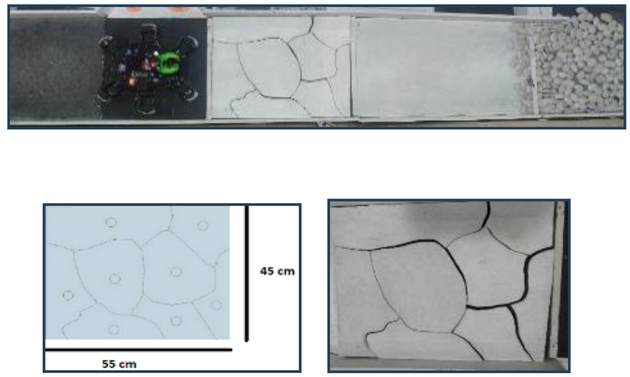

Figure 14. Route testing PID and broken road

Figure 14 show the robot facing the route testing and move with Adaptive PID. Here is the response table.

TABLE II
SETPOINT 0° ADAPTIVE PID

Experiment	Setpoint (θ)	Settling Time(s)	Rise Time(s)	Overshoot (degree)	Success
1	0°	2,40	0,80	9,31	Success
2	0°	3,40	1,80	13,81	Success
3	0°	3,50	1,90	11,06	Success
4	0°	3,40	1,80	4,06	Success
5	0°	3,40	1,80	12,31	Success
Mean		3,22	1.62	10,11	

TABLE III
SETPOINT 90° ADAPTIVE PID

Experiment	Setpoint (θ)	Settling Time(s)	Rise Time(s)	Overshoot (degree)	Success
1	90°	3,40	1,80	19,65	Success
2	90°	3,40	1,80	17,07	Success
3	90°	2,10	0,50	8,10	Success
4	90°	3,00	1,50	15,90	Success
5	90°	3,10	1,50	15,34	Success
Mean		3,00	1,42	10,21	

TABLE IV
SETPOINT -90° ADAPTIVE PID

Experiment	Setpoint (θ)	Settling Time(s)	Rise Time(s)	Overshoot (degree)	Success
1	-90°	3,70	3,40	17,63	Success
2	-90°	3,10	2,70	5,70	Success
3	-90°	2,10	1,80	19,85	Success
4	-90°	3,70	3,40	17,01	Success
5	-90°	4,50	4,00	18,18	Success
Mean		3,42	1.42	15,68	

From the three data tables above, the PID constants are set using the Adaptive PID method, which produces varying constants with Kp values ranging from 0 to 0,75, Ki = 0 to 0,0003 and Kd = 0 to 0,003. Using these constant values, the system response produces an average settling time of 3,06 seconds, an average rise time of 1,99 seconds, and an average overshoot of 10 – 15,6%.

Figure 14. Route testing PID and broken road

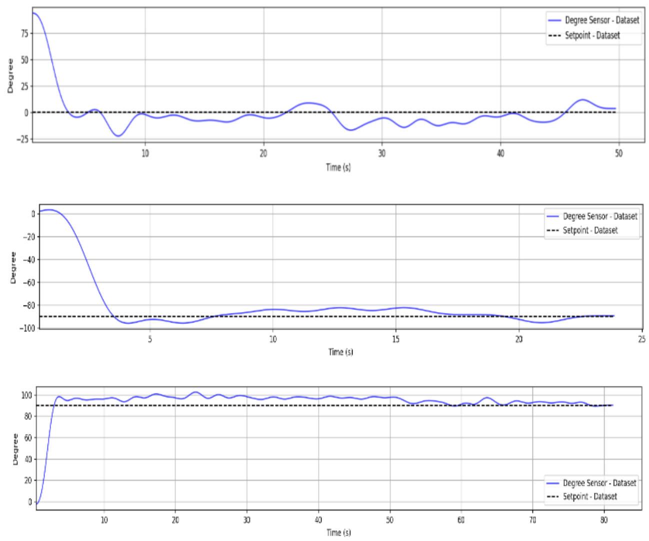

Figure 15. Curve response from adaptive PID

Figure 15 show curve response from adaptive PID with setpoint 0, -90, and 90 Degree. From the three graphs, adaptive PID produces oscillations that are quite responsive. although there is some responsiveness, the PID system is well integrated into the system. the robot can correct its direction of rotation.

IV. CONCLUSION

The study shows that the sensor used has a very low angular error rate, with errors between 0,01% and 0,1%, highlighting its reliability for real-time applications. The adaptive PID method significantly accelerates robot stabilization and minimizes overshoot by adjusting dynamically to changing conditions. This combination of precise sensing and adaptive control enhances navigation and stability across complex environments. Performance metrics include an average settling time of 3,06 seconds, rise time of 1,99 seconds, and overshoot between 10,00% and 15,60%. Future work could involve refining adaptive PID and testing alternative membership functions and fuzzy rules for improved response.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude for the financial support from Barelang Robotics and Artificial Intelligence (BRAIL) and the Electrical Engineering Department of Batam State Polytechnic. We are particularly grateful to the dedicated members of the Search and Rescue Robotics Research Team at BRAIL for their critical support and insights. Special appreciation is extended to Mr. Senanjung Prayoga and Mr. Anugerah Wibisana, the Head Advisors, for their exceptional mentorship throughout this project. Finally, the authors express deep gratitude to their families, especially their parents, for being a continuous source of motivation and support in achieving their academic goals.

REFERENCES

[1] B. Pengembangan Talenta, I. Pusat, P. Nasional, K. Pendidikan, And D. Teknologi, "Pedoman Kontes Robot Indonesia (Kri) Pendidikan Tinggi Tahun 2024."

[2] N. F. Lawrence, A. Qurthobi, And M. H. Barri, "Perancangan Dan Implementasi Kendali Lingkar Tertutup Gait Robot Hexapod Untuk Mendeteksi, Menaiki Dan Menuruni Tangga Design And Implementation Of Hexapod Gait Close-Loop Control For Detecting, Ascending And Descending Stairs," Vol. 8, No. 5, P. 5843, 2021.

[3] A. Febriawan And W. Sapto Aji, "Rotating Control On Robots Indonesian Abu Robot Contest With Pid And Imubno055 Controls Kendali Berputar Pada Robot Kontes Robot Abu Indonesia Dengan Kendali Pid Dan Imubno055," *Buletin Ilmiah Sarjana Teknik Elektro*, Vol. 2, No. 1, Pp. 14–23, 2020, Doi: 10.12928/Biste.V2i1.987.

[4] M. S. Kurnia, A. Pracoyo, And L. Kamajaya, "Kontrol Pid Untuk Navigasi Robot Berkaki Berbasis Sensor Tof Dan Bno055," *Jurnal Elektronika Dan Otomasi Industri*, Vol. 10, No. 3, Pp. 462–470, Sep. 2023, Doi: 10.33795/Elkolind.V10i3.1952.

[5] M. Fajar Kesuma And Z. S. T. Mirza, "Perbandingan Sistem Kendali Pid Dan Fuzzy Logic Pada Quadcopter." [Online]. Available: Https://Raharja.Ac.Id/2020/04/06/Logika-Fuzzy/.

[6] Y. Pratama *Et Al.*, "Jurnal Media Informatika Budidarma Implementasi Kontrol Pid Untuk Percepatan Rotasi Pada Robot Hexapod," Vol. 7, Pp. 1869–1876, 2023, Doi: 10.30865/Mib.V7i4.6433.

[7] U. Kabir, M. F. Hamza, A. Haruna, And G. Sani, "Performance Analysis Of Pid, Pd And Fuzzy Controllers For Position Control Of 3-Dof Robot Manipulator."

[8] I. M. Sidiq, E. Ihsanto, And Y. Yuliza, "Rancang Bangun Sistem Navigasi Robot Hexapod 3 Dof Di Ruangan Labirin Menggunakan Metode Fuzzy Sugeno Berbasis Arduino," *Jurnal Teknologi Elektro*, Vol. 11, No. 3, P. 133, Oct. 2020, Doi: 10.22441/Jte.2020.V11i3.004.

[9] S. Sukardi And S. Gunawan Zain, "Pengembangan Algoritme Gerak Robot Hexapod Menggunakan Logika Fuzzy." [Online]. Available: Https://Ojs.Unm.Ac.Id/Jessi/Index

[10] "Heading Control For Quadruped Stair Climbing Based On Pd Controller For The Krsri Competition".

[11] J. Subur, S. Suryadhi, And E. Arohman, "Penerapan Metode Fuzzy Logic-Pid Sebagai Kontrol Gerak Robot Berkaki Enam (Hexapod)," *Proceedings Series On Physical & Formal Sciences*, Vol. 6, Pp. 98–108, Oct. 2023, Doi: 10.30595/Pspfs.V6i.858.

[12] L. Ode *Et Al.*, "Sistem Kendali Hybrid Fuzzy-Pid Pada Kinematika Robot Berkaki 4 Menggunakan Sensor Gyroscope," 2023.

[13] D. S. Pamungkas *Et Al.*, "Menggunakan Sensor Imu Sebagai Penstabil Robot Exoskeleton Assistive Walker Dengan Metode Pid," *Jurnal Simetris*, Vol. 14, No. 1, 2023.

[14] R. Wang, "Hybrid Gait Planning Of A Hexapod Robot," *Modern Electronic Technology*, Vol. 4, No. 2, P. 11, Dec. 2020, Doi: 10.26549/Met.V4i2.5075.

[15] Y. Pratama, C. Saputra, A. N. Toscany, And M. I. Bustami, "Kontrol Navigasi Robot Hexapod Berbasis Inverse Kinematic Dan Body Kinematic Untuk Stabilitas Optimal Di Medan Ekstrem," Vol. 6, No. 3, Pp. 1898–1909, 2024, Doi: 10.47065/Bits.V6i3.6007.