Optimization of robot stabilization response with PID control

Faikul Umam[1*], *Firmansyah* Adiputra[2], *Ach.* Dafid[1], *Hairil* Buadiarto[1], *Muhammad* Fuad[1], *Firman* Maolana[1], *Bintang* Nur Alif[3]

[1]Department of Mechatronics Engineering, Faculty of Engineering, Universitas Trunodjoyo, Madura
[2]Department of Information System, Faculty of Engineering, Universitas Trunodjoyo, Madura
[3]Department of Informatics Engineering, Faculty of Engineering, Universitas Trunodjoyo, Madura

Abstract. The stability of balance is needed on tools or workpieces, especially in the field of robotics. In this study, it tries to apply balance stability settings to a robot, namely a two-wheeled balancing robot. This two-wheeled robot balancing is an extension of the inverted pendulum model that is placed on a wheeled carriage. This study applies a control strategy to keep the robot in an upright position and has a fast response to return to a specified point. Robot balancing in this study discusses balancing a robot with two wheels as a balance driver which is located on the right and left of the robot. The concept of balance applied in this research is that the robot will try to balance itself, so that the robot is in an upright position which will continue to be repeated until it finds the desired set point. In order to add a robot that requires controlling a DC motor that has received input from the IMU sensor that can read the angle of the robot. The angular change in the robot is then calculated by the PID to produce a fast and precise response. This process is repeated until the robot is perpendicular to the plane. The test results of the balancing robot with the large PID method, the accuracy of the tool in the trial is believed to be 99.87%, so it can be said that the balancing robot can work well.

1 Introduction

The rapid development of robotics technology has significantly enhanced productivity across various industries, healthcare, and agriculture. These advancements are driven by constant innovation in robotics systems, which are increasingly integrated into everyday human activities. Notably, advancements in robotics are often a product of intense scientific competition to develop cutting-edge technologies that fulfill societal needs [1–6].

One of the most essential technologies in modern robotics is the ability of a robot to balance on two wheels. This ability, which allows a robot to remain upright and stable, is pivotal for its application in transportation, offering an innovative solution to mobility challenges. Achieving this balance requires synchronization between the robot's sensors and the control system used [7,8].

The development of such self-balancing systems has drawn attention due to the challenges of maintaining stability using only two wheels. These challenges are closely related to principles in physics, such as torque, angular momentum, and the center of mass. A two-wheeled robot must continuously adjust its position by using dynamic feedback to stay upright, which involves both mechanical and control systems innovations [9–12]. This research focuses on developing a self-balancing robot that utilizes artificial intelligence for improved performance. The robot, resembling an inverted pendulum, employs sensors such as an IMU (Inertial Measurement Unit) to detect angular displacement, which is then processed by a microcontroller (Arduino Nano) to adjust its position. The robot aims to balance itself by moving forward or backward to find the optimal center of gravity and remain perpendicular to the ground【19–24】. The control method employed is the Proportional-Integral-Derivative (PID) controller, which is specifically chosen for its ability to minimize error and ensure fast, responsive balancing [25–30].

Despite the progress in balancing robots, there remain challenges regarding optimal component layout, system design, and control strategies that hinder the robot's ability to maintain a stable position in dynamic environments[13–18]. Therefore, it is necessary to develop a balancing robot from the mechanical side and the control side.

This balancing robot has two wheels as a balance driver which is located on the right and left of the robot. The way this research works is that the robot detects changes in angle by utilizing the IMU sensor or tilt angle sensor. Then it will be processed by the Arduino Nano microcontroller which is none other than the controller, so that robot 2 will try to balance itself which continues to be done repeatedly until it finds the desired set point [19–24]. The change in the angle of the robot is then calculated and evaluated using the Proportional Integral Derivative (PID) method. The use of the PID control method aims to reduce errors in a balancing robot,

* Corresponding author: faikul@trunojoyo.ac.id

because this robot requires a fast response and requires a small error in order to keep the robot's position perpendicular to a flat plane [25–30].

The research contribution of this study is to design and implement a balancing robot with an advanced control system using PID. This research not only provides an effective solution to improve the robot's balancing ability but also presents a practical application of control theory in modern robotics. By focusing on the PID control method, this study aims to optimize the robot's stability, achieving a high precision in balancing, even under varying load conditions. This research bridges the gap between theoretical control methods and practical robotics applications, addressing real-world challenges in balancing systems.

2 Methods

Stabilization of the balancing robot response using the PID controller method. In this balancing robot, it has two wheels as balance drivers which are located on the left and right of the robot[31–35]. The robot will try to balance itself, so that the robot is in an upright position or until it finds the desired set point, which was previously known by the IMU sensor readings. The IMU sensor functions as a tilt angle reader and the DC motor functions as an actuator that will move back and forth to get the balance point according to the input from the Arduino Nano microcontroller by utilizing PID control as a control system that is able to balance the robot body with a fast and precise response, and can minimize error rate.[36–42]

2.1 Block diagram

The following is a block diagram on a balancing robot using a PID controller which can be seen in Fig. 1. The meaning of the symbols used in the block diagram shown in Fig. 1 are as follows:

a. Set point speed (Vs) and set point angle (θs)
b. Angle sensor to read the tilt of the robot body (θA)
c. Motor speed (V)
d. The difference between the set point and sensor readings (θ'and V')

Plant output after controlled (θ) and encoder sensor reading (V1).

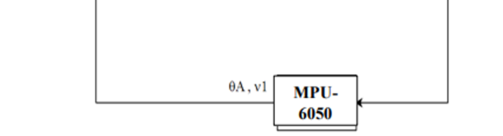

Fig. 1. Block diagram of a balancing robot using a PID controller.

Block diagram on balancing robot using PID controller. This system uses a close loop control system, because there is a feedback process to minimize errors and make the system approach the specified output [43–47]. The first process is input in the form of the expected set point angle (θs) and the set point of velocity (Vs). Then there are IMU sensors and encoder sensors where the readings will be stored as A. Furthermore, the sensor readings are stored in (V1). After that the readings of A will be compared with s and also the results of the encoder sensor readings will be compared with the motor speed (Vs). Then after the calculation is complete, it will be processed by the PID. The results of the PID control system are motor speed (V) and will be sent to a DC motor or balancing robot.

2.2 Flowchart system

The following is a system flowchart of a balancing robot using a PID controller shown in Fig. 2.

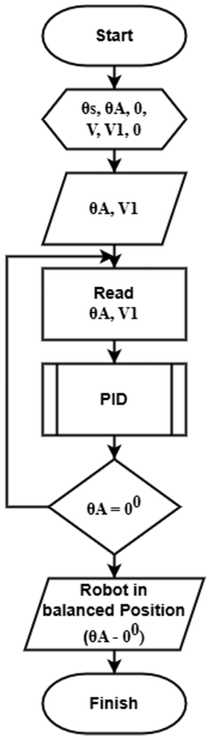

Fig. 2. Flowchart of a balancing robot system using a PID controller.

In Fig. 2 is a flowchart of a balancing robot system using a PID controller, in which in this case there is an initialization or introduction in the form of parameters s, A, ‘, V, V1. Where s is the set point angle and A is the IMU sensor angle, ' the difference between the set point angle and the encoder sensor reading angle, V DC motor speed, V1 encoder sensor reading, then plant output after being controlled. After the initialization process, there will be a reading of the IMU sensor and encoder sensor which will then be processed by the PID controller [48–52]. After the PID process is complete, the speed will move according to the input. Then if the robot is not in a straight line, it will be read back by the IMU sensor and encoder. If it is perpendicular or the robot is in balance, then the process is complete.

2.3 Flowchart PID controller

The PID controller flowchart of the balancing robot is shown in Fig. 3. In the PID controller flowchart shown in Fig. 3, it begins with the introduction of Kp, Ki, Kd, s, TimeSampling, LastError, and V. Then proceed to the values of Kp, Ki , Kd, s, A, then a condition will be carried out which if the input value is less than 0. If yes, then the error value calculation process is carried out, namely the error value (θ') is equal to the set point value minus the sensor input value as in equation 1.

$$\theta'(error)_{Forward} = \theta a - \theta s \quad (1)$$

If not or the input value is more than 0 then, the error value (θ') is equal to the sensor input value minus the set point angle as in equation 2.

$$\theta'(error)_{Backward} = \theta s - \theta a \quad (2)$$

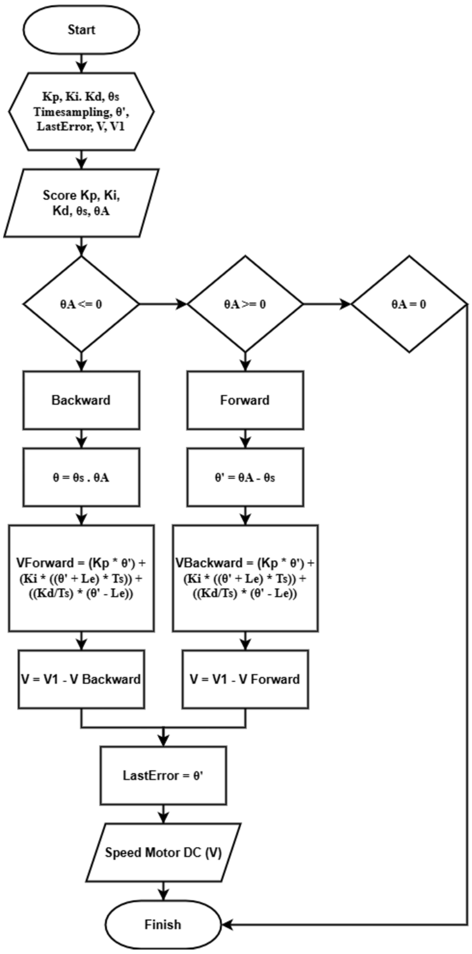

Fig. 3. Flowchart PID controller.

After the error value is obtained, of course it will be formulated in the PID controller formula by finding the speed value in each forward and backward condition as in equation 3 and equation 4.

$$V_{Forward} = (Kp * \theta') + (Ki * ((\theta' + Last\ Error) * Ts)) + ((Kd/Ts) * (\theta' - Last\ Error)) \quad (3)$$

$$V_{Backward} = (Kp * \theta') + (Ki * ((\theta' + Last\ Error) * Ts)) + ((Kd/Ts) * (\theta' - Last\ Error)) \quad (4)$$

Then after obtaining the control results for the value of V from the PID in each forward and backward condition, the V calculation process will be carried out, namely the difference between V1 and V both forward and backward positions as in equation 5 and equation 6.

$$V = V1 - V_{Forward} \quad (5)$$

$$V = V1 - V_{Backward} \quad (6)$$

After that, it is continued with the last error process equal to '. The last step is to obtain the results of the DC motor speed (V) in each forward and reverse position.

3 Result and discussion

3.1 Tuning PID test

PID tuning test is done to get the most appropriate formula or value of Kp, Ki, and Kd. The right values of Kp, Ki, and Kd can provide a good response to the balancing robot system. Where the motor output in the form of motor rotation speed will be stable, so that the balancing robot can move and respond well to achieve balanced or balanced conditions. When affected by a disturbance, the PID will process the disturbance with an output in the form of a motor speed value, so that the robot will respond quickly to reach a stable position again. The results of the test looking for the PID value are shown in Table 1. From the experiments that have been carried out, it can be analyzed that the experiment was carried out 10 times by giving different Kp, Ki, Kd values and the same set point value of the robot position. The values of Kp, Ki, and Kd are given randomly to find a value that responds well to motor movement, so that the robot returns to a stable position. The first experiment by giving a value of Kp = 0, Ki = 0, and Kd = 0 with the robot position parallel to 180°, the result is that the robot cannot move because the values of Kp, Ki, Kd are both 0 or empty.

Table 1. Tuning PID test.

No.	Robot Position Values Kp, Ki, Kd	Response
1.	180° Kp = 0 Ki = 0 Kd = 0	Robot can't move
2.	180° Kp = 69.6 Ki = 0 Kd = 0	The robot does not stand balanced and still moves back and forth wildly
3.	180° Kp = 69.6 Ki = 250 Kd = 0	Robot can stand and still move uncontrollably
4.	180° Kp = 69.6 Ki = 0 Kd = 62.5	Robot can stand and still move uncontrollably
5.	180° Kp = 0 Ki = 250 Kd = 62.5	Robot moves, but can't stand up yet
6.	180° Kp = 15.9 Ki = 0.98 Kd = 36.00	Robot can stand balanced
7.	180° Kp = 0 Ki = 23 Kd = 56	Robot moves wildly and is not balanced
8.	180° Kp = 69.4 Ki = 34 Kd = 67	Robot can stand but still not balanced
9.	180° Kp = 69.7 Ki = 25 Kd = 45	Robot can stand and still move uncontrollably
10.	180° Kp = 75.5 Ki = 0 Kd = 63	Robot can stand but still not balanced

The second experiment by giving a value of Kp = 69.6, Ki = 0, and Kd = 0 with the robot position parallel to 180°, the result is that the robot cannot stand in balance and still moves back and forth wildly. This is because the values of Ki and Kd are both 0 or empty. The third experiment by giving a value of Kp = 69.6, Ki = 250, and Kd = 0 with the robot position parallel to 180°, 43 the results are that the robot can stand but the robot is still moving uncontrollably. This is because the Kd value is 0 or empty, where the response after the error is not updated. The fourth experiment by giving a value of Kp = 69.6, Ki = 0, and Kd = 62.5 with the robot position parallel to 180°, the result is that the robot can stand but the robot is still moving uncontrollably. This is because the value of Ki is 0 or empty. The fifth experiment by giving a value of Kp = 0, Ki = 250, and Kd = 62.5 with the robot position parallel to 180°, the result is that the robot can move but the robot cannot stand. This is because the value of Kp is 0 or empty. The sixth experiment by giving a value of Kp = 15.9, Ki = 0.98, Kd = 36.00 with the robot position parallel to 180°, the result is that the robot can stand in a balanced and stable manner. The seventh experiment by giving a value of Kp = 0, Ki = 23, Kd = 56 with the robot position parallel to 180°, the result is that the robot moves uncontrollably so that it has not reached a balanced position. This is because the value of Kp 0 where the motor response is not controlled. The eighth experiment by giving a value of Kp = 69.4, Ki = 34, Kd = 67 with the robot position parallel to 180°, the result is that the robot can stand but the motor response is still not controlled so that the robot cannot reach a balanced position. Of the 10 experiments that have been carried out, the correct PID tuning value was obtained in the sixth experiment with the value of Kp = 15.9, Ki = 0.98, Kd = 36.00. In the sixth experiment, the motor can move well to respond to the robot's body in an effort to achieve a balanced condition.

3.2 Robot balance angel test

Testing the balance of the robot by providing different set point robot angles. The test was carried out 10 times with different set points and the same KP, Ki, Kd values. This test is carried out to observe the robot's response with the values of KP, Ki, Kd from the PID tuning results that have been obtained in the previous test, whether the robot can achieve a balanced condition. The following are the results of testing the balance of the robot with different set point values which are shown in Table 2. From Table 2, it is obtained data on how long the robot can reach a stable condition again. The test is done by giving a different set point value, then the robot will try to reach a stable position perpendicular to (180°) again. Observe how long it takes the robot to work. The following is an explanation of the first test to the 10th. In the first experiment, a set point value of 180° was given with a value of Kp = 15.9, Ki = 0.98, Kd = 36.00. The test is carried out by observing how long the robot reaches a stable condition perpendicular to the position of the set point (180°). The robot's response when turned on can reach a balanced position within 7.34 seconds. The following is a picture of the first experiment with a set point value of 180° which is shown in Fig. 4.

Table 2. Robot balance angle testing.

No	Set Point	Kp	Ki	Kd	Respon (second)
1	180°	15.9	0.98	36	7,34
2	175°	15.9	0.98	36	6,63
3	170°	15.9	0.98	36	7,48
4	165°	15.9	0.98	36	6,20
5	160°	15.9	0.98	36	6,55
6	0°	15.9	0.98	36	7,46
7	5°	15.9	0.98	36	6,67
8	10°	15.9	0.98	36	7,54
9	15°	15.9	0.98	36	5,91
10	20°	15.9	0.98	36	6,48

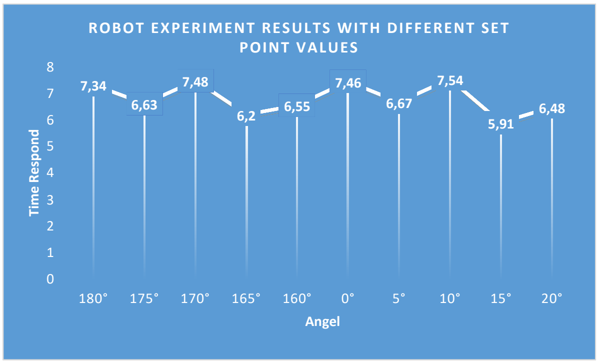

Fig. 4. Robot experiment results with different set point values.

3.3 Robot load testing

Load testing on the balancing robot is carried out by providing additional load on the robot body. This test was conducted to observe how long the robot response to reach a stable condition and to test the robot's resistance when given additional load. The balancing robot is given additional loads with varying weights. The following table shows the experimental results of testing robot loads with varying weights which are shown in Table 3.

Table 3. Additional load testing.

No	Load	Start Time	End Time	Result (Second)
1	46 gram	15:18:43.188	15:18:44.798	1,61
2	20 gram	15:23:15.002	15:23:16.680	1,67
3	45 gram	15:27:36.001	15:27:37.168	1,167
4	33 gram	15:30:17.278	15:30:18.438	1,16
5	90 gram	15:46:44.020	15:46:45.821	1,81
6	36 gram	15:33:44.282	15:33:45.624	1,342
7	20 gram	15:36:30.057	15:36:31.498	1,441
8	81 gram	15:40:47.160	15:40:48.545	1,385
9	34 gram	15:42:00.182	15:42:01.354	1,172
10	10 gram	15:57:29.146	15:57:30.657	1,511

From the tests that have been carried out, it can be analyzed that the balancing robot can still work even though it is given an additional load on the robot body. In the 5th test, the largest additional load was in the form of a li-po battery with a weight of 90 grams. The experiment, which has been conducted 10 times, has the set point set at 180°, which means that the robot is in a straight or balanced state. The balancing robot takes 1.81 seconds to reach a stable state again after being given an additional load. From the test with the heaviest load, it takes the longest time than testing with other loads.

In this test, it provides the largest additional load in the form of a li-po battery with a weight of 90 grams. In this test, the balancing robot takes the longest time, which is 1.81 seconds to reach a stable state again after being given an additional load.

Fig. 5 is the result of testing the balance of the balancing robot by providing an additional load. The test was carried out 10 times with different loads. The test is carried out by placing a load on the robot and observing the robot's position from being perpendicular (180°) to returning to an upright position (180°). In other words, how long does it take for the robot to reach a stable position again when given an additional load. 51 The longest time for the balancing robot to reach a stable condition is 1.81 seconds with a load of 90 grams.

In this test, it provides the largest additional load in the form of a li-po battery with a weight of 90 grams. In this test, the balancing robot takes the longest time, which is 1.81 seconds to reach a stable state again after being given an additional load.

,

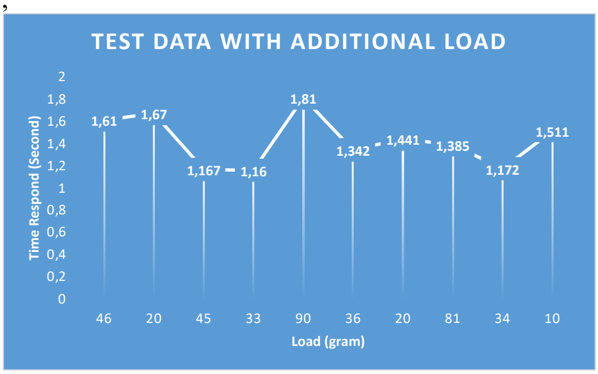

Fig. 5. Test graph with additional load.

Fig. 5 above is the result of testing the balance of the balancing robot by providing an additional load. The test was carried out 10 times with different loads. The test is carried out by placing a load on the robot and observing the robot's position from being perpendicular (180°) to returning to an upright position (180°). In other words, how long does it take for the robot to reach a stable position again when given an additional load. The longest time for balancing the robot to reach a stable condition is 1.81 seconds with a load of 90 grams.

4 Conclusion

After doing some testing and analyzing the data obtained from this thesis, it can be concluded that:

a. The balancing robot takes a maximum of 7.54 seconds to reach a state of balance. The set point value of 10° takes the longest to reach a balanced position.
b. The use of the MPU-6050 sensor is very appropriate for robot balancing systems, as evidenced by the success rate of 99.87%.
c. The PID controller method produces a fairly good response. It is proven from the test results that the PID controller has the longest time to reach a stable condition, only 7.54 seconds.
d. PID tuning is done 10 times with different tunings, so that it can produce the right tuning.

References

1. M. V. de Souza Almeida, A. B. Alves, E. S. Carvalho, E. C. C. da Silva, B. D. R. de Mesquita, Educational Robotics as a Teaching Field and Technology Integration: Application of CAD, CAM and 3D printing in structural robot construction. IEEE World Conference on Engineering Education (EDUNINE). 1–5 (2020). https://doi.org/10.1109/EDUNINE48860.2020.9149541
2. N. Al-Mouh *et al.*, A professional development workshop on advanced computing technologies for high and middle school teachers. 15th International Conference on Information Technology Based Higher Education and Training (ITHET). 1–4 (2016). https://doi.org/10.1109/ITHET.2016.7760696
3. A. C. T. de Vasconcelos *et al.*, Utilizing Educational Robotics for Learning About Sustainability Issues. Latin American Robotics Symposium (LARS), Brazilian Symposium on Robotics (SBR), and Workshop on Robotics in Education (WRE). 384–388 (2022). https://doi.org/10.1109/LARS/SBR/WRE56824.2022.9995840
4. A. Chopra, J. K. Chandel, V. K. Panchal, S. Sinha, Understanding of cognitive robotics. International Conference on Computing for Sustainable Global Development (INDIACom). 317–320 (2014). https://doi.org/10.1109/IndiaCom.2014.6828151
5. J. Liu *et al.*, Development And Application of Robotics Technologies in Future Intelligent Soldier Squad. 35th Chinese Control and Decision Conference (CCDC). 845–850 (2023). https://doi.org/10.1109/CCDC58219.2023.10326681
6. D. Deeb, I. Merkuryev, V. Pakhomov, New Educational Technologies for Training Specialists in the Field of Mechatronics and Robotics. 7th International Conference on Information Technologies in Engineering Education (Inforino). 1–4 (2024). https://doi.org/10.1109/Inforino60363.2024.10552009
7. Z. J. Belmonte, C. Bandola, R. Decapia, E. C. Gonzaga, Do Technological Developments Reduce

Unemployment in the Philippines?. 62nd International Scientific Conference on Information Technology and Management Science of Riga Technical University (ITMS). 1–5 (2021). https://doi.org/10.1109/ITMS52826.2021.9615294

8. L. Y. Y. Lu, Y. L. Lan, J. S. Liu, A novel approach for exploring technological development trajectories. IEEE International Conference on Management of Innovation & Technology (ICMIT). 504–509 (2012). https://doi.org/10.1109/ICMIT.2012.6225857
9. A. Ivoilov, V. Trubin, V. Zhmud, L. Dimitrov, The Power Consumption Decreasing of the Two-Wheeled Balancing Robot. International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon). 1–8 (2018). https://doi.org/10.1109/FarEastCon.2018.8602775
10. K. Mitobe, G. Capi, A global stability analysis of legged balance controllers under ZMP constraints. IEEE International Symposium on Robotic and Sensors Environments (ROSE). 1–6 (2023). https://doi.org/10.1109/ROSE60297.2023.10410748
11. M. A. Imtiaz *et al.*, Control System Design, Analysis & Implementation of Two Wheeled Self Balancing Robot (TWSBR). IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON). 431–437 (2018). https://doi.org/10.1109/IEMCON.2018.8614858
12. P. Suebsaiprom, C. L. Lin, 2-DOF barycenter mechanism for stabilization of fish-robots. IEEE 8th Conference on Industrial Electronics and Applications (ICIEA). 1119–1122 (2013). https://doi.org/10.1109/ICIEA.2013.6566534
13. X. Wang *et al.*, Teleoperation control for bimanual robots based on RBFNN and wave variable. 9th International Conference on Modelling, Identification and Control (ICMIC). 1–6 (2017). https://doi.org/10.1109/ICMIC.2017.8321519
14. G. Masengo *et al.*, A Design of Lower Limb Rehabilitation Robot and its Control for Passive Training. 10th Institute of Electrical and Electronics Engineers International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER). 152–157 (2020). https://doi.org/10.1109/CYBER50695.2020.9278952
15. M. A. Arteaga-Peréz, J. Pliego-Jiménez, J. G. Romero, Experimental Results on the Robust and Adaptive Control of Robot Manipulators Without Velocity Measurements. IEEE Transactions on Control Systems Technology. **28**, 6, 2770–2773 (2020). https://doi.org/10.1109/TCST.2019.2945915
16. C. S. Putra *et al.*, Multilayer Control for Coordinating Three - Wheeled Omnidirectional Mobile Robots. 6th International Conference on Instrumentation, Control, and Automation (ICA). 182–187 (2019). https://doi.org/10.1109/ICA.2019.8916737
17. R. P. Saputra *et al.*, Decomposed-Objective Model Predictive Control for Autonomous Casualty Extraction. IEEE Access. **9**, 39656–39679 (2021). https://doi.org/10.1109/ACCESS.2021.3063782
18. M. Zarghami *et al.*, Fast and precise positioning of wheeled Omni-directional robot with input delay using model-based predictive control. In: Proceedings of the 33rd Chinese Control Conference. 7800–7804 (2014). https://doi.org/10.1109/ChiCC.2014.6896302
19. Q. A. Al-Haija *et al.*, Cryptosystem based on Arduino Microcontroller Useful for Small Scale Networks. Procedia Comput Sci. **34**, 639–646 (2014). https://doi.org/10.1016/j.procs.2014.07.091
20. A. A. Sulayman *et al.*, Arduino microcontroller based real-time monitoring of haemodialysis process for patients with kidney disease. e-Prime - Advances in Electrical Engineering, Electronics and Energy. **7**, 100403 (2024). https://doi.org/10.1016/j.prime.2023.100403
21. P. Itterheimová, F. Foret, P. Kubáň, High-resolution Arduino-based data acquisition devices for microscale separation systems. Anal Chim Acta. 1153, 338294 (2021). https://doi.org/10.1016/j.aca.2021.338294
22. R. Cuervo *et al.*, Low-cost and open-source neonatal incubator operated by an Arduino microcontroller. HardwareX. **15**, e00457 (2023). https://doi.org/10.1016/j.ohx.2023.e00457
23. N. A. Arsyad *et al.*, Breast milk volume using portable double pump microcontroller Arduino Nano. Enferm Clin. **30**, 555–558 (2020). https://doi.org/10.1016/j.enfcli.2019.07.159
24. J. A. Marín-Marín, P. A. García-Tudela, P. Duo-Terrón, Computational thinking and programming with Arduino in education: A systematic review for secondary education. Heliyon. **10**, 8, e29177 (2024). https://doi.org/10.1016/j.heliyon.2024.e29177
25. B. Zhao, B. Y. Wang, J. Cheng, D. Xu, Study on Hybrid PID Control of Multi-Terrain Mobile Robot Based on Moment Balance Mechanism. 2nd International Conference on Automation, Robotics and Computer Engineering (ICARCE). 1–6 (2023). https://doi.org/10.1109/ICARCE59252.2024.10492547
26. L. Zhang, X. Ren, Q. Guo, Balance Control of a Wheeled Hopping Robot. 39th Chinese Control Conference (CCC). 3801–3805 (2020). https://doi.org/10.23919/CCC50068.2020.9189592
27. Z. Chen, X. Ruan, Y. Li, Balancing control of a cubical robot balancing on its corner. IEEE 15th International Workshop on Advanced Motion Control (AMC). 631–636 (2018). https://doi.org/10.1109/AMC.2019.8371167
28. F. F. Rabbany, A. Qurthobi, A. Suhendi, Design of Self-Balancing Virtual Reality Robot Using PID Control Method and Complementary Filter. IEEE International Conference on Industry 40, Artificial Intelligence, and Communications Technology

(IAICT). 15–19 (2021). https://doi.org/10.1109/IAICT52856.2021.9532576

29. T. A. Mai, T. S. Dang, D. N. Anisimov, E. Fedorova, Fuzzy-PID Controller for Two Wheels Balancing Robot Based on STM32 Microcontroller. International Conference on Engineering Technologies and Computer Science (EnT). 20–4 (2019). https://doi.org/10.1109/EnT.2019.00009
30. N. Hasanah, A. H. Alasiry, B. Sumantri, Two Wheels Line Following Balancing Robot Control using Fuzzy Logic and PID on Sloping Surface. International Electronics Symposium on Engineering Technology and Applications (IES-ETA). 210–215 (2018). https://doi.org/10.1109/ELECSYM.2018.8615483
31. P. Grelewicz *et al.*, Application of Machine Learning to Performance Assessment for a Class of PID-Based Control Systems. IEEE Trans Syst Man Cybern Syst. **53**, 7, 4226–4238 (2023). https://doi.org/10.1109/TSMC.2023.3244714
32. H. Sira-Ramírez, E. W. Zurita-Bustamante, C. Huang, Equivalence Among Flat Filters, Dirty Derivative-Based PID Controllers, ADRC, and Integral Reconstructor-Based Sliding Mode Control. IEEE Transactions on Control Systems Technology. **28**, 5, 1696–1710 (2020). https://doi.org/10.1109/TCST.2019.2919822
33. W. Zhang, H. Lv, Disturbance Observer-Based PID Control System Using DNA Strand Displacement and Its Application in Exponential Gate. IEEE Access. **11**, 113160–113175 (2023). https://doi.org/10.1109/ACCESS.2023.3320956
34. S. Gao *et al.*, Global Nested PID Control of Strict-Feedback Nonlinear Systems With Prescribed Output and Virtual Tracking Performance. IEEE Transactions on Circuits and Systems II: Express Briefs. **67**, 2, 325–329 (2020). https://doi.org/10.1109/TCSII.2019.2907141
35. W. Tan, W. Han, J. Xu, State-Space PID: A Missing Link Between Classical and Modern Control. IEEE Access. 10, 116540–116553 (2022). https://doi.org/10.1109/ACCESS.2022.3218657
36. W. Wei, K. Kurita, J. Kuang, A. Gao, Real-Time 3D Arm Motion Tracking Using the 6-axis IMU Sensor of a Smartwatch. IEEE 17th International Conference on Wearable and Implantable Body Sensor Networks (BSN). 1–4 (2021). https://doi.org/10.1109/BSN51625.2021.9507012
37. N. Ruangpayoongsak, Stride Length Estimation using Fuzzy Logic and a Single Shoe Mounted IMU Sensor. Research, Invention, and Innovation Congress: Innovative Electricals and Electronics (RI2C). 120–125 (2023). https://doi.org/10.1109/RI2C60382.2023.10355946
38. N. Petranon, S. Umchid, Design and Development of Hand Movement Detection Device for Sign Language using IMU and sEMG Sensors. Research, Invention, and Innovation Congress: Innovative Electricals and Electronics (RI2C). 1–4 (2023). https://doi.org/10.1109/RI2C60382.2023.10356033
39. Li C, Yu L, Fei S. Real-Time 3D Motion Tracking and Reconstruction System Using Camera and IMU Sensors. IEEE Sens J. **19**, 15, 6460–6466 (2019). https://doi.org/10.1109/JSEN.2019.2907716
40. N. Verdel *et al.*, Time Synchronization in Wireless IMU Sensors for Accurate Gait Analysis During Running. IEEE International Workshop on Sport, Technology and Research (STAR). 126–129 (2023). https://doi.org/10.1109/STAR58331.2023.10302652
41. M. Ajčević *et al.*, Assessment of mobility deficit and treatment efficacy in adhesive capsulitis by measurement of kinematic parameters using IMU sensors. IEEE International Symposium on Medical Measurements and Applications (MeMeA). 1–5 (2020). https://doi.org/10.1109/MeMeA49120.2020.9137157
42. H. M. Saputra *et al.*, Effect of IMU Sensor Positioning on 1-DOF Angle Measurement Accuracy for Robotic Charging Station (RoCharg-v1) Manipulator. International Conference on Computer, Control, Informatics and its Applications (IC3INA). 96–101 (2023). https://doi.org/10.1109/IC3INA60834.2023.10285798
43. H. E. Kiran, A. Akgul, O. Yildiz, E. Deniz, Lightweight encryption mechanism with discrete-time chaotic maps for Internet of Robotic Things. Integration. **93**, 102047 (2023). https://doi.org/10.1016/j.vlsi.2023.06.001
44. H. Mohamad, S. Ozgoli, Online gait generator for lower limb exoskeleton robots: Suitable for level ground, slopes, stairs, and obstacle avoidance. Rob Auton Syst. **160**, 104319 (2023). https://doi.org/10.1016/j.robot.2022.104319
45. C. Qin *et al.*, Development and application of an intelligent robot for rock mass structure detection: A case study of Letuan tunnel in Shandong, China. International Journal of Rock Mechanics and Mining Sciences. **169**, 105419 (2023). https://doi.org/10.1016/j.ijrmms.2023.105419
46. F. Khadivar, K. Yao, X. Gao, A. Billard, Online active and dynamic object shape exploration with a multi-fingered robotic hand. Rob Auton Syst. **166**, 104461 (2023). https://doi.org/10.1016/j.robot.2023.104461
47. J. Zhang *et al.*, Design of parallel multiple tuned mass dampers for the vibration suppression of a parallel machining robot. Mech Syst Signal Process. **200**, 110506 (2023). https://doi.org/10.1016/j.ymssp.2023.110506
48. J. Lu *et al.*, A New Calibration Method of MEMS IMU Plus FOG IMU. IEEE Sens J. **22**, 9, 8728–

8737 (2022). https://doi.org/10.1109/JSEN.2022.3160692

49. A. R. Parrish, C. S. Jao, A. M. Shkel, Stance Phase Detection for ZUPT-Aided INS Using Knee-Mounted IMU in Crawling Scenarios. IEEE Sens Lett. **6**, 5, 1–4 (2022). https://doi.org/10.1109/LSENS.2022.3165760
50. A. Tripathi, A. K. Mondal, L. Kumar, A. P. Prathosh, ImAiR: Airwriting Recognition Framework Using Image Representation of IMU Signals. IEEE Sens Lett. **6**, 10, 1–4 (2022). https://doi.org/10.1109/LSENS.2022.3206307
51. H. Jeon, J. Min, H. Bang, W. Youn, Quaternion-Based Iterative Extended Kalman Filter for Sensor Fusion of Vision Sensor and IMU in 6-DOF Displacement Monitoring. IEEE Sens J. **22**, 23, 23188–23199 (2022). https://doi.org/10.1109/JSEN.2022.3214580
52. H. Zhang, Y. Li, Z. Li, 6-D Spatial Localization of Wireless Magnetically Actuated Capsule Endoscopes Based on the Fusion of Hall Sensor Array and IMU. IEEE Sens J. **22**, 13, 13424–33 (2022). https://doi.org/10.1109/JSEN.2022.3175919