

# Desain dan Kontrol Modular *Independent Drive Independent Steering Mobile Robot* Aktuator

Brahmantya Bagus Irsa Adinugraha<sup>1\*</sup>, Indrazno Siradjuddin<sup>2</sup>, Leonardo Kamajaya<sup>3</sup>  
<sup>1,2,3</sup>Politeknik Negeri Malang, Malang,

\*Penulis Korespondensi, e-mail: [brahmantyaabian@gmail.com](mailto:brahmantyaabian@gmail.com)

Received: 11/10/2023

Revised: 26/10/2023

Accepted: 28/10/2023

**Abstract.** This study explores and implements the Swerve drive as an actuator on a mobile robot with a modular independent drive and independent steering system. This innovation allows each wheel on the robot to move freely and face any direction, controllable independently for both steering and driving, to achieve optimal performance. The Swerve drive enables independent wheel rotation and swift movement, enhancing the robot's manoeuvrability and performance in robotic competitions. However, the complexity and higher costs pose challenges in implementing the Swerve drive on mobile robots. The challenge of controlling the Swerve drive lies in using two motors, requiring precise control to adjust motor rotation according to target values. The proposed solution is the highly effective PID control implementation, with low average errors, specifically 2.54% for driving speed and 0.91% for steering position. The first type of PID control is chosen to regulate driving speed, providing accurate and overshoot-free responses. In the steering position control, The implementation of PID control can significantly enhance the performance of robots in various settings, including ABUROBOCON and RoboCup competitions, as well as real-life scenarios. Compared to P or PI control, it offers greater responsiveness and efficiency.

**Keywords:** Swerve drive, Independent steering, Independent driving, Holonomic

**Abstrak.** Penelitian ini mengkaji dan menerapkan Swerve drive sebagai aktuator pada robot mobile dengan sistem modular independent drive dan independent steering. Inovasi ini memungkinkan setiap roda pada robot untuk bergerak secara bebas dan menghadap ke arah manapun, dapat dikendalikan secara terpisah dan mandiri baik untuk steering maupun driving, guna mencapai kinerja optimal. Swerve drive memungkinkan rotasi roda yang independen dan pergerakan yang cepat, meningkatkan manuverabilitas dan kinerja robot dalam kompetisi robotika. Namun, kompleksitas dan biaya yang lebih tinggi menimbulkan tantangan dalam menerapkan Swerve drive pada robot mobile. Tantangan kontrol Swerve drive adalah mengendalikan dua motor untuk mengatur rotasi sesuai target. Solusi yang diusulkan adalah implementasi kontrol PID yang sangat efektif, dengan tingkat kesalahan rata-rata rendah, khususnya sebesar 2.54% untuk kecepatan penggerak dan 0.91% untuk posisi kemudi. Jenis kontrol PID dipilih untuk mengatur kecepatan driving, memberikan respons yang akurat dan bebas overshoot. Dalam kendali posisi steering, Implementasi kontrol PID meningkatkan kinerja robot di berbagai pengaturan, termasuk kompetisi ABUROBOCON dan RoboCup, serta dalam kehidupan nyata, dengan responsivitas dan efisiensi lebih baik dibandingkan dengan kontrol P atau PI.

**Kata Kunci:** Swerve drive, Independent steering, Independent driving, Holonomic

## I. PENDAHULUAN

Penelitian pada bidang robotika khususnya mobile robot atau wheeled mobile robot Terus mengalami peningkatan yang signifikan seiring berjalannya waktu.. Ada berbagai alasannya yaitu antara lain meluasnya penggunaan robot di berbagai bidang untuk memudahkan pekerjaan dan mengurangi risiko kecelakaan yang melukai orang. Oleh karena itu, penelitian di bidang robotika menjadi daya tarik tersendiri bagi para peneliti [1]. Kemajuan terbaru dalam robot beroda telah

membuka banyak aplikasi potensial di dunia industri, medis, Rumah sakit, dan gudang [2].

Berdasarkan pergerakannya robot dibagi menjadi 2 yaitu *holonomic* dan *non-holonomic* [3]. Wheeled Mobile Robot (WMR) yang konvensional, tidak dapat berjalan ke samping tanpa manuver awal yang menjadi batasan dalam pergerakannya. Meskipun ada kemajuan dalam kemampuan manuver WMR, mereka masih tidak bisa menandingi *holonomic* robot. Misalnya, ada beberapa motor yang dipasang secara statis posisi di sisi kiri dan kanan robot dalam diferensial desain. Robot ini disebut *non-holonomic*

hal ini dikarenakan robot tidak bisa bergerak ke segala arah yang memungkinkan. Sebaliknya, robot yang menggunakan konfigurasi omni-directional disebut robot *holonomic* karena mampu bergerak di semua arah pada setiap titik waktu [4].

Banyak tipe dari *drive* omnidirectional salah satunya adalah *Swerve drive* [5]. Sistem swerve adalah sebuah konsep sederhana dalam modul penggerak yang dikendalikan secara independen. Di dalam 1 Modul *Swerve drive* terdiri 2 motor, satu gearbox, 2 encoder, dan roda. Dimana satu dari kedua motor digunakan sebagai *steering* dan satu lagi untuk *driving* [6]. Mekanisme *Swerve drive* secara konvensional membutuhkan delapan motor untuk penggerak independen dalam robot roda empat [7].

Motor BLDC digunakan dalam penelitian ini karena keunggulannya dibandingkan motor lain karena torsi tinggi, kecepatan tinggi, dan ukurannya yang kecil [8]. BLDC melibatkan proses yang kompleks seperti pemodelan, pemilihan pengontrol, simulasi, dan parameterisasi. Di dalam sebuah pergerakan robot tentu saja tidak lepas dari performa dari aktuatornya, untuk mengontrol aktuatornya agar berputar sesuai dengan set point sehingga membuat robot dapat menuju titik yang telah ditentukan dengan tepat dan akurat, maka berbagai solusi kontrol canggih telah diusulkan untuk desain pengontrol kecepatan motor BLDC. Kontroler PID atau kontroler *fuzzy* dapat digunakan sebagai unit kontrol untuk mengontrol kecepatan motor BLDC [9].

Kontroler PID dipilih karena karakteristiknya yang telah terbukti dan banyak digunakan dalam kontrol motor BLDC. Pengontrol menggunakan tiga parameter, proporsional, integral, dan turunan, untuk mengontrol output berdasarkan perbedaan antara set point dan kondisi sebenarnya dari motor. Metode ini memberikan kontrol motor BLDC yang stabil dan akurat, sehingga seringkali menjadi pilihan pertama untuk aplikasi yang membutuhkan kecepatan dan posisi motor yang presisi. Selain itu, pengontrol PID relatif mudah dikonfigurasi dan diimplementasikan di berbagai jenis sistem kontrol termasuk motor BLDC. Pengaturan kecepatan adalah aspek penting

dalam pengembangan algoritma kontrol kinematik untuk mengendalikan gerakan robot. Pengaturan kecepatan ini dilakukan dengan menggunakan kendali PID, yang merupakan metode yang digunakan untuk mengontrol kecepatan motor DC yang menggerakkan robot [10][11].

Diskusi dalam penelitian ini, terdapat beberapa bagian, antara lain sebagai berikut : bab 2 bersis tentang metode penelitian yang terdiri dari teori motor BLDC dan control PID, bab 3 berisikan tentang hasil dan pembahasan dari penggunaan kontrol PID untuk mengatur kecepatan dan posisi motor BLDC, dan bab 4 berisikan tentang kesimpulan yang dapat disimpulkan serta saran untuk penelitian selanjutnya.

## II. METODOLOGI

BAB ini berisi paparan tentang tahapan-tahapan atau langkah-langkah yang digunakan untuk menyelesaikan masalah atau menggambarkan solusi yang ditawarkan untuk mengatasi permasalahan yang ada termasuk alat, bahan dan atau turunan rumus yang digunakan.

### A. *Swerve drive*

*Swerve drive* adalah jenis *drivetrain* yang dirancang khusus agar robot dapat berputar saat berjalan di sepanjang jalur yang dilintasinya. Pada 1 modul Normal *Swerve drive* menggunakan 2 buah motor yang dapat mengontrol arah hadap dan kecepatan roda secara independen dan tidak menggunakan kombinasi kedua motornya. *Swerve drive* lebih kompleks dan lebih mahal dari pada *meccanum driver*. *Swerve drive* memerlukan 4 motor untuk *driving*, 4 motor untuk *steering*, dan 8 buah *encoder* untuk mengukur jarak yang ditempuh roda. *Swerve drive* memiliki kelebihan yaitu konsep yang simpel, rodayang simpel, Modul *drive* dan *steer* yang dikendalikan secara independent, dan sedikitnya tingkat selip.

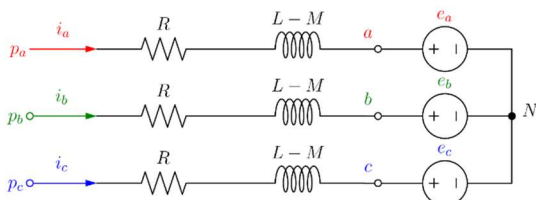


**Gambar 1.** Desain 3D modular *Swerve drive*

### B. Motor BLDC

Jenis motor yang digunakan adalah Motor BLDC (Brushless DC), menggunakan sumber listrik DC tanpa adanya sebuah sikat atau *brushes* [12]. Motor BLDC sendiri memiliki beberapa kelebihan, di antaranya adalah tingkat efisiensi yang tinggi dikarenakan tidak memiliki sikat, motor BLDC memiliki kerusakan mekanik yang lebih rendah dibandingkan dengan motor DC [13]. Motor BLDC merupakan motor 3 fasa sehingga belitannya menggunakan konfigurasi *wye*(Y). Motor BLDC bergerak dengan memanfaatkan gaya magnet yang saling tarik menarik dan tolak menolak. Ketika arus mengalir ke salah satu fasa maka hal itu akan menyebabkan adanya medan magnet yang menyebabkan gaya tarik pada rotor yang memiliki kutub magnet yang bereda dan menolak kutub magnet yang sama.

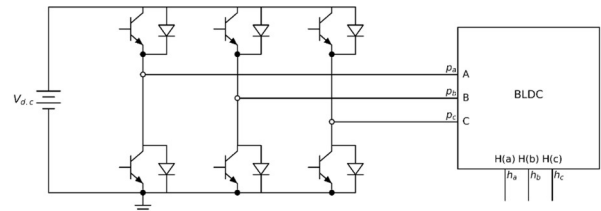
Pada motor BLDC terdapat sebuah back-EMF. Back-EMF adalah tegangan balik yang berasal dari belitan motor saat motor BLDC berputar.



**Gambar 2.** Rangkaian *equivalen* motor BLDC

Motor BLDC menggunakan sistem komutasi elektronik sehingga memanfaatkan hall sensor untuk mendeteksi posisi rotor. Perbedaan fasa antar *coil* adalah sebesar 120 derajat. Pembacaan posisi

rotor berguna untuk proses *switching* pada inverter dan berguna untuk menyalurkan arus pada *coil* [14]. Motor BLDC mempunyai sistem komutasi yang sering disebut *six step commutation*. Hal itu terjadi karena adanya pergantian logika pada masing-masing IGBT pada inverter

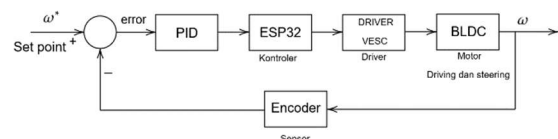


**Gambar 3.** Rangkaian inverter BLDC

### C. PID Control

PID (*Proportional-Integral-Derivative*) *controller* adalah metode kontrol yang efektif dengan tiga komponen utama: *Proporsional* (P), *Integral* (I), dan *Derivatif* (D). Komponen-komponen ini memberikan respons terhadap kesalahan saat ini, mengatasi akumulasi kesalahan seiring waktu, dan menghasilkan respons yang stabil terhadap perubahan kesalahan. Serta penggunaan kontrol PID untuk mengatasi masalah non-linearitas, variasi parameter, dan perubahan beban dalam pengendalian motor BLDC[15]. Kombinasi ketiga komponen ini membantu menghasilkan sinyal kontrol presisi dan stabil untuk mengatur kecepatan motor BLDC dengan *drive* VESC dan sensor encoder, dengan input kontrol berdasarkan kesalahan antara set poin kecepatan dan posisi aktual motor. Dengan mengatur konstanta P, I, dan D yang sesuai, kontrol PID memungkinkan motor mencapai kecepatan target dengan efisiensi dan presisi.

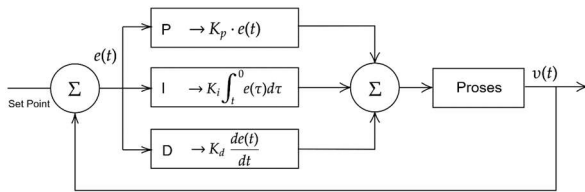
### D. Permodelan Kontrol PID



**Gambar 4.** Diagram blok kontrol PID

Pemodelan kontrol PID bertujuan untuk mengembangkan sistem kontrol yang dapat menyesuaikan variabel sistem dengan cepat, stabil,

dan akurat dengan mengintegrasikan tiga jenis kontrol, yaitu proporsional (P), integral (I), dan turunan (D). Proporsional digunakan untuk menanggapi kesalahan saat ini, Integral digunakan untuk menyeimbangkan kesalahan dari waktu ke waktu, dan Diferensial digunakan untuk menyesuaikan tingkat perubahan kesalahan. Dalam sistem kendali motor BLDC, fungsi kontrol PID adalah mengolah sinyal *error* atau kesalahan antara set point dan nilai terukur, lalu menghasilkan sinyal kontrol yang diteruskan ke *drive* VESC untuk mengontrol kecepatan atau posisi motor sesuai dengan nilai set point yang diinginkan. Kontrol PID memungkinkan peningkatan respons, stabilitas, dan pengurangan kesalahan dalam sistem kendali motor.



**Gambar 5.** Diagram kontrol PID

### III. HASIL DAN PEMBAHASAN

Pada bagian ini dilakukan percobaan untuk mendemonstrasikan akurasi kontrol kecepatan dan kontrol posisi menggunakan kontrol PID, dan eksperimen dengan sensor *encoder* untuk menentukan posisi *steering* motor dan kecepatan motor *driving*. Serta dalam penelitian ini sensor *encoder* yang digunakan ialah *encoder* CUI AMT102 untuk mengetahui kecepatan motor *driving* dan posisi motor *steering* BLDC.

#### A. Pengujian Motor Driving

Pada pengujian ini dilakukan dengan cara memberikan nilai RPM mulai dari 50 sampai 500. Pengujian dilakukan dengan 2 metode yaitu menggunakan kontroller berupa PID. Pengujian ini dilakukan untuk mendapatkan nilai *error* pada saat menggunakan sebuah *controller*.

Pada pengujian yang dilakukan, diperoleh hasil yang direpresentasikan dalam tabel 1. Tabel ini menunjukkan variasi nilai *error* yang terjadi saat menggunakan kontrol PID dalam sistem yang diuji.

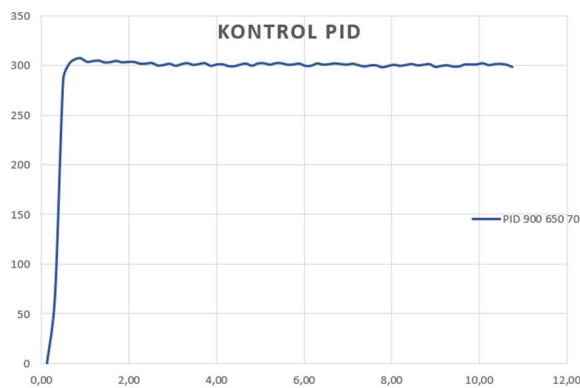
Dalam tabel tersebut, terlihat bahwa rata - rata nilai *error* yang dihasilkan cukup besar, yaitu sebesar 2,54%. Rata-rata ini mengindikasikan adanya kesalahan yang signifikan dalam sistem meskipun telah menggunakan kontrol PID. Nilai *error* yang lebih kecil umumnya diinginkan untuk mencapai performa yang lebih baik dalam mengontrol sistem.

**Tabel 1.** Hasil pengujian kecepatan *driving* menggunakan kontrol PID

| RPM              | ERPM | Hasil | Error (%) |
|------------------|------|-------|-----------|
| 50               | 350  | 51,1  | 2,20      |
| 75               | 525  | 73,3  | 2,27      |
| 100              | 700  | 97,6  | 2,40      |
| 125              | 875  | 121,6 | 2,72      |
| 150              | 1050 | 145,5 | 3,00      |
| 175              | 1225 | 170,4 | 2,63      |
| 200              | 1400 | 194,5 | 2,75      |
| 225              | 1575 | 219,9 | 2,27      |
| 250              | 1750 | 243,5 | 2,60      |
| 275              | 1925 | 268,3 | 2,44      |
| 300              | 2100 | 292,8 | 2,40      |
| 325              | 2275 | 316,6 | 2,58      |
| 350              | 2450 | 341,5 | 2,43      |
| 375              | 2625 | 364,5 | 2,80      |
| 400              | 2800 | 389,3 | 2,68      |
| 425              | 2975 | 414,1 | 2,56      |
| 450              | 3150 | 439   | 2,44      |
| 475              | 3325 | 463,4 | 2,44      |
| 500              | 3500 | 487,2 | 2,56      |
| Rata -Rata Error |      |       | 2,54      |

#### B. Respon Motor Driving

Pada bagian ini menunjukkan respon kecepatan *driving* dari kontrol PID. Kontrol PID pada *driving* memiliki performa baik dengan overshoot 2.87%, rise time 0.54s, settling time 2.79s, dan steady-state *error* yang kecil. Kontrol PID dapat meningkatkan respon kecepatan *driving* dengan mempercepat rise time, mengurangi overshoot, dan mempercepat waktu untuk mencapai steady state. Meskipun kontrol PID pada *driving* merupakan contoh kontrol PID yang baik untuk respon kecepatan *driving*. Ditunjukkan pada Gambar 6.



**Gambar 6.** Grafik respon PID pada *driving*

### C. Pengujian Motor Steering

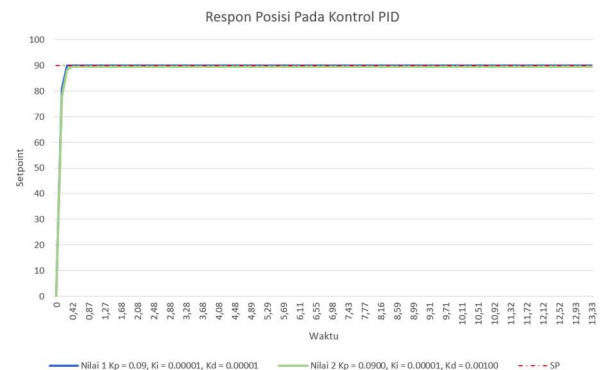
Pada pengujian ini dilakukan dengan cara memberikan nilai posisi mulai dari 0 sampai 360. Pengujian dilakukan dengan metode kontroller berupa PID. Pengujian ini dilakukan untuk mendapatkan *error* pada sistem ini.

Tabel 2 menunjukkan hasil dari penggunaan kontrol PID untuk kontrol posisi. Penggunaan kontrol PID menghasilkan rata-rata nilai *error* sebesar 0.91%. Namun, ketika posisi berada pada 90 derajat dan 330 derajat, terdapat *error* yang besar 5% dan tidak melebihi dari nilai tersebut.

**Tabel 2.** Hasil pengujian posisi *steering* menggunakan control PID

| Posisi                  | Hasil | Error |
|-------------------------|-------|-------|
| 0                       | 0     | 0,00  |
| 30                      | 30    | 0,00  |
| 60                      | 63    | 5,00  |
| 90                      | 93    | 3,33  |
| 120                     | 120   | 0,00  |
| 150                     | 149   | 0,67  |
| 180                     | 180   | 0,00  |
| 210                     | 210   | 0,00  |
| 240                     | 240   | 0,00  |
| 270                     | 271   | 0,37  |
| 300                     | 299   | 0,33  |
| 330                     | 335   | 1,52  |
| 360                     | 358   | 0,56  |
| <b>Rata -Rata Error</b> |       | 0,91  |

### D. Respon Motor Steering



**Gambar 7.** Respon posisi *steering*

Dari hasil analisis pada gambar 7, terdapat perbedaan karakteristik antara dua data pada kontrol PID. Pada nilai konstanta 1 ( $K_P = 0.0900$ ,  $K_I = 0.00001$ , dan  $K_D = 0.00100$ ), sistem menunjukkan risetime sebesar 0.10 detik (s) dan tidak mengalami overshoot. Sistem mencapai settling time sekitar 0.10 detik (s) dan memiliki steady state *error* sebesar 0.10°. Sementara itu, pada nilai konstanta 2 ( $K_P = 0.0900$ ,  $K_I = 0.00001$ , dan  $K_D = 0.00001$ ), respon sistem juga menunjukkan risetime yang sama, tetapi sistem memiliki steady state *error* yang lebih tinggi, yaitu sebesar 0.62°. Hal ini menunjukkan adanya kesalahan dalam mencapai nilai setpoint secara akurat pada nilai konstanta 2. Meskipun nilai konstanta 2 memiliki risetime dan settling time yang relatif cepat, nilai konstanta 1 menunjukkan performa yang lebih baik dalam mengontrol sistem dan mencapai nilai setpoint yang diinginkan. Karakteristik respon yang lebih akurat, stabil, dan tanpa overshoot pada nilai konstanta 1 menjadikannya pilihan yang lebih optimal dalam implementasi kontrol PID untuk sistem yang dianalisis.

## IV. KESIMPULAN DAN SARAN

Pada penelitian pengaturan kecepatan dan posisi motor BLDC pada modular *Swerve drive*, dapat disimpulkan bahwa penggunaan kontrol PID untuk mengatur kecepatan motor *driving* pada modular *Swerve drive* sangat cocok karena kontrol PID mampu meningkatkan respon dan mengurangi nilai rata-rata *error*. Penggunaan kontrol PID juga efektif dalam mengatur posisi motor *steering*, karena mampu meningkatkan respon dan signifikan

mengurangi nilai rata-rata *error* pada kontrol posisi. Dalam pengaturan kecepatan, kontrol PID memiliki keunggulan dalam merespon perubahan dengan cepat dan mengatur kecepatan motor secara stabil tanpa overshoot. Sedangkan dalam pengaturan posisi, kontrol PID dapat secara akurat mencapai nilai setpoint yang diinginkan dengan steady-state yang lebih baik dibandingkan dengan kontrol *fuzzy*. Oleh karena itu, penggunaan kontrol PID pada pengaturan kecepatan dan posisi motor BLDC pada modular *Swerve drive* adalah pilihan yang optimal untuk meningkatkan kinerja dan akurasi sistem.

#### UCAPAN TERIMA KASIH

Alhamdulillah, dengan penuh rasa syukur, penulis ingin mengucapkan terima kasih kepada Allah SWT atas nikmat dan rahmat-Nya yang telah memberikan kesempatan untuk menyelesaikan penyusunan skripsi ini. Shalawat dan salam senantiasa tercurahkan kepada junjungan kita, Nabi Muhammad SAW, yang telah membimbing kita dari zaman kegelapan menuju zaman yang terang benderang ini. Penyusunan skripsi ini merupakan upaya untuk memenuhi sebagian persyaratan guna memperoleh gelar diploma 4 Sarjana Terapan Teknik (S.Tr.T), khususnya di Jurusan Teknik Elektro Politeknik Negeri Malang.

#### REFERENSI

- [1] I. Siradjuddin, G. Al Azhar, A. Murdani, and M. L. M. Faizin, "Desain dan pemodelan kontrol kinematik pergerakan robot beroda dengan menggunakan 6 roda omni-wheels," *Jurnal Eltek*, vol. 18, no. 1, p. 116, 2020, doi: 10.33795/eltek.v18i1.226.
- [2] A. Sofwan, H. R. Mulyana, H. Afrisal, and A. Goni, "Development of omni-wheeled mobile robot based-on inverse kinematics and odometry," *2019 6th International Conference on Information Technology, Computer and Electrical Engineering, ICITACEE 2019*, pp. 1–6, 2019, doi: 10.1109/ICITACEE.2019.8904418.
- [3] K. V. Vasireddy, M. Rose, Rakesh, and S. Suresh, *OMNI-DIRECTIONAL ROBOT USING SWERVE DRIVE*. 2021.
- [4] M. A. Al Mamun, M. T. Nasir, and A. Khayyat, "Embedded system for motion control of an omnidirectional mobile robot," *IEEE Access*, vol. 6, no. 8, pp. 6722–6739, 2018, doi: 10.1109/ACCESS.2018.2794441.
- [5] J. Carothers, *Design of a triple singularity drive for mobile wheeled robots*. 2014.
- [6] B. Denoma, M. Kendall, and N. Poulos, "4-wheel Independent Steering 'Swerve Drive' Table of Contents," 2022.
- [7] V. K. Pranav, K. S. Kumar, A. R. Nair, and G. Udupa, "Design and Manufacture of Octagonal Rover for Space Exploration," *2020 4th International Conference on Automation, Control and Robots, ICACR 2020*, pp. 48–52, 2020, doi: 10.1109/ICACR51161.2020.9265497.
- [8] A. Shyam and F. D. J. L., "A comparative study on the speed response of BLDC motor using conventional PI controller, anti-windup PI controller and fuzzy controller," in *2013 International Conference on Control Communication and Computing (ICCC)*, Dec. 2013, pp. 68–73. doi: 10.1109/ICCC.2013.6731626.
- [9] K. S. Devi, R. Dhanasekaran, and S. Muthulakshmi, "Improvement of speed control performance in BLDC motor using fuzzy PID controller," *Proceedings of 2016 International Conference on Advanced Communication Control and Computing Technologies, ICACCCT 2016*, no. 978, pp. 380–384, 2017, doi: 10.1109/ICACCCT.2016.7831666.
- [10] I. Siradjuddin, T. Winarno, M. Khairuddin, M. N. Achmadiah, R. P. Wicaksono, and G. Al Azhar, "An Exponential Decreased Kinematic and PID Low Level Control Schemes for an Omni-Wheeled Mobile Robot," in *2021 International Conference on Electrical and Information Technology (IEIT)*, 2021, pp. 274–280. doi: 10.1109/IEIT53149.2021.9587370.
- [11] I. Siradjuddin, L. Kamajaya, S. Wibowo, A. A. Rofiq, G. Al Azhar, and M. Khairuddin, "Near Real Time Simulation of an Independent Steering Independent Driving Mobile Robot," in *2021 3rd International Conference on*

*Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA)*, 2021, pp. 62–68. doi: 10.1109/SUMMA53307.2021.9632085.

- [12] K. S. K. Veni, N. S. Kumar, and J. Gnanavadivel, “Low cost *fuzzy* logic based speed control of BLDC motor drives,” *Proceedings of IEEE International Conference on Advances in Electrical Technology for Green Energy 2017, ICAETGT 2017*, vol. 2018-January, pp. 7–12, 2018, doi: 10.1109/ICAETGT.2017.8341453.
- [13] Y. Chandra Wibowo and S. Riyadi, “Analisa Pembebanan Pada Motor Brushless Dc (Blde),” pp. 277–282, 2019, doi: 10.5614/sniko.2018.33.
- [14] K. Sreeram, “Design of *Fuzzy Logic Controller* for Speed Control of Sensorless BLDC Motor Drive,” *2018 International Conference on Control, Power, Communication and Computing Technologies, ICCPCCT 2018*, pp. 18–24, 2018, doi: 10.1109/ICCPCCT.2018.8574280.
- [15] M. Mahmud, S. M. A. Motakabber, A. H. M. Zahirul Alam, and A. N. Nordin, “Adaptive PID *Controller* Using for Speed Control of the BLDC Motor,” in *2020 IEEE International Conference on Semiconductor Electronics (ICSE)*, 2020, pp. 168–171. doi: 10.1109/ICSE49846.2020.9166883.