

AUGMENTED MONTE CARLO LOCALIZATION FOR WHEELED SOCCER ROBOT

Mochamad Rizal Fauzi, Hendawan Soebhakti

Department of Electrical Engineering, Study Program Robotics Engineering, Politeknik Negeri Batam

Article Info	ABSTRACT
<p>Article history:</p> <hr/> <p>Keywords:</p> <p>Robot sepak bola beroda Lokalisasi Estimasi posisi Robocup MSL Omnivision</p>	<p>Jurnal ini menyajikan penerapan teknik lokalisasi pada robot sepak bola beroda dengan Monte Carlo Localization untuk estimasi posisi secara global. Dengan tiga komponen utama; <i>motion model</i> menggunakan odometry, <i>measurement model</i> dengan <i>likelihood field</i> dan sampling dengan <i>low variance resampling</i>. Untuk mengatasi masalah ketika robot terdorong oleh robot lainnya, ditambahkan metode untuk memberikan posisi sembarang terhadap partikel. Percobaan dilakukan dengan menggerakkan robot ke sembarang posisi, kemudian kembali ke posisi awal. Data hasil estimasi dan odometry kemudian dibandingkan dengan posisi sebenarnya. Jumlah partikel yang digunakan adalah 100, 200, 600 dan 1000. Hasil percobaan menunjukkan dengan 1000 partikel menghasilkan rata-rata error yang sangat kecil, yaitu 1.9 cm pada sumbu x, 5,8 cm pada sumbu y dan 0,05 radian.</p>
<p>Corresponding Author:</p> <p>Mochamad Rizal Fauzi Department of Electrical Engineering, Study Program Robotics Engineering Politeknik Negeri Batam 29461 Batam Kota, Kota Batam, Indonesia Email: fauzimochamadrizal2@gmail.com</p>	

1. INTRODUCTION

Lokalisasi menjadi syarat agar robot dapat melakukan navigasi secara otonom. Seperti pada Robocup Middle Size League (MSL), dimana banyak tim yang berkompetisi pada robot sepak bola beroda, merupakan contoh dari penerapan lokalisasi. Semua robot harus bermain dengan hanya menggunakan sensor yang terdapat pada robot saja. Hal ini menjadi tantangan bagi lokalisasi mengingat kecepatan robot yang tinggi dan akselerasi yang agresif dapat menyebabkan slip.

Barelang63 merupakan salah satu tim yang ikut dalam kompetisi pada MSL, dengan gambaran umum pada [1]. Untuk lokalisasi, sebelumnya menggunakan algoritma Monte Carlo Localization dengan mendeteksi landmark dari beberapa titik yang telah ditentukan, diantaranya sudut kiri-kanan, kotak penalti kiri-kanan dan tiang gawang kiri-kanan [2]. Landmark tersebut dideteksi menggunakan kamera depth dengan posisi pada bagian depan robot YOLO v3 object detection system [3]. Dari sistem tersebut menghasilkan rata-rata error 13,64 cm pada sumbu x dan 12,44 cm pada sumbu y [4]. Oleh karena itu, Untuk meningkatkan performa robot ketika bermain, maka diperlukan estimasi posisi dengan ketentuan

- i) Area deteksi sensor lebih menyeluruh, tidak hanya bagian depan robot, dan
- ii) Mampu memulihkan posisi yang disebabkan kegagalan estimasi.

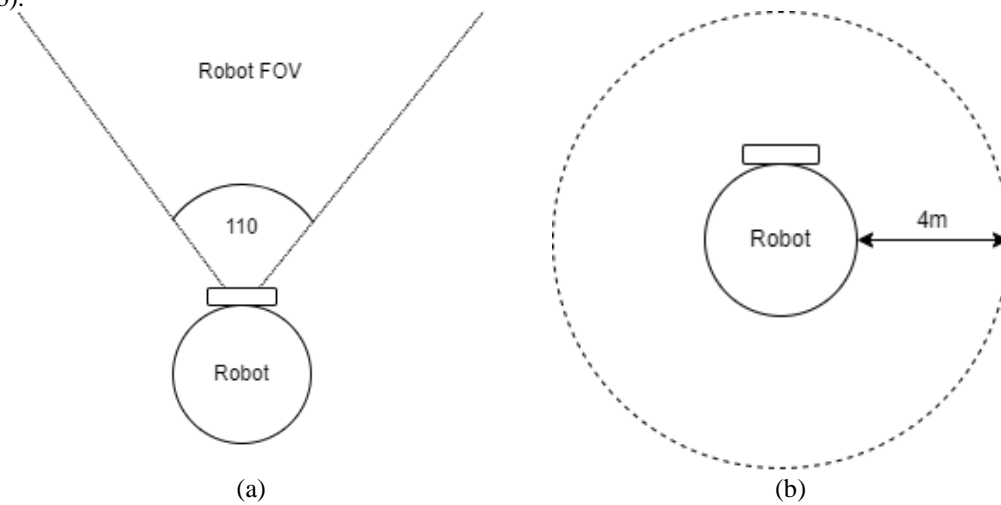
Oleh karena itu, diajukan algoritma Augmented Monte Carlo Localization untuk memulihkan posisi yang disebabkan oleh kegagalan estimasi dan menggunakan kamera omnivision untuk menjangkau area yang lebih menyeluruh.

2. METHOD

2.1. Line Detection

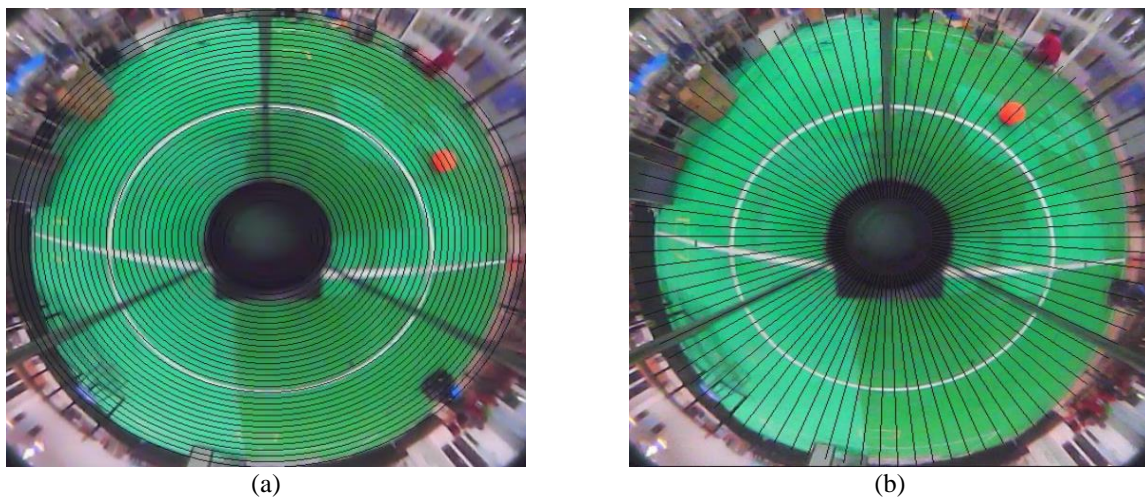
Pendeteksian garis pada robot sepak bola beroda merupakan elemen yang penting dalam sistem lokalisasi. Dengan menggunakan kamera vision, robot dapat mengidentifikasi garis-garis putih pada lapangan untuk menentukan

posisi dan orientasi relatif terhadap robot. Kamera yang digunakan merupakan kamera omnidirectional daripada menggunakan kamera depth. Dengan menggunakan kamera depth, meskipun robot dapat menjangkau jarak yang cukup jauh, tetapi area pendeteksiannya tidak cukup luas seperti pada gambar 1 (a). Sedangkan, dengan menggunakan kamera omnidirection, area pendeteksiannya meliputi 360 derajat dari robot dengan jarak maksimum 4meter seperti pada gambar 1 (b).



Gambar 1. Representasi Field of View (FoV) dari (a) kamera Zed2 dengan 110° dan (b) kamera omnivision dengan 360° dan radius 4m

Untuk mendeteksi garis putih pada lapangan, diterapkan dua jenis tipe pendeteksian garis yang disebut *scanlines*: radial dan circular. *Scanlines* adalah daftar dari posisi pixel pada gambar yang berbentuk sesuai dengan tipenya [5]. Gambar 2 (a) memperlihatkan *circular scanlines* dengan bentuk garis melingkar dari titik tengah dengan radius paling kecil (sesuai ukuran radius robot yang terdeteksi) sampai radius lingkaran sama dengan setengah dari ukuran frame gambar. Gambar 2 (b) memperlihatkan *radial scanlines* dengan bentuk garis memanjang dari tengah. Setelah jarak pixel dari garis putih didapatkan, dilakukan teknik interpolasi linear agar menghasilkan jarak yang sesuai dengan jarak sesungguhnya.



Gambar 2. (a) circular scanline dan (b) radial scanline

2.2. Odometry Motion Model Sampling

Odometry motion model menggunakan pengukuran odometry sebagai kontrol. Odometry biasanya didapat dari informasi encoder pada roda [6]–[9]. Akan tetapi, karena kemungkinan terjadinya slip, sehingga tidak ada koordinat transformasi yang tetap antara koordinat odometry dengan koordinat global.

Gambar 3 menunjukkan algoritma sample motion model odometry dengan informasi odometry u_t dan initial pose x_{t-1} . Pada baris 2 dan 3 mengkonversi relative odometry x, y menjadi koordinat global particle dengan menggunakan matriks rotasi persamaan (1). Probabilistic motion model mengasumsikan informasi odometry terganggu oleh *noise*. Sehingga, pada baris 4-6 ditambahkan probabilitas dari random zero mean gaussian model *prob()* sebagai bentuk noise. Baris 10 mengembalikan nilai pose x_t yang diperoleh dari pose partikel ditambah motion parameter.

```

1:  Algorithm sample_motion_model_odometry( $u_t, x_{t-1}$ ):
2:       $\delta x_{particle} = dx \cos(\theta_{particle}) - dy \sin(\theta_{particle})$ 
3:       $\delta y_{particle} = dx \sin(\theta_{particle}) + dy \cos(\theta_{particle})$ 
4:       $\hat{\delta} x_{particle} = \delta x_{particle} + \mathbf{prob}(0.0, \sigma_x)$ 
5:       $\hat{\delta} y_{particle} = \delta y_{particle} + \mathbf{prob}(0.0, \sigma_y)$ 
6:       $\hat{\delta} \theta_{particle} = \delta \theta_{particle} + \mathbf{prob}(0.0, \sigma_\theta)$ 
7:       $x_{particle} = x_{particle} + \hat{\delta} x_{particle}$ 
8:       $y_{particle} = y_{particle} + \hat{\delta} y_{particle}$ 
9:       $\theta_{particle} = \theta_{particle} + \hat{\delta} \theta_{particle}$ 
10:  return  $x_t(x_{particle}, y_{particle}, \theta_{particle})$ 

```

Gambar 3. Algoritma untuk sampling $p(x_t | u_t, x_{t-1})$ berdasarkan informasi odometry. Perubahan pose relatif pada waktu t adalah \mathbf{dx}, \mathbf{dy} dipetakan menjadi pose global $\delta x = (\delta x_{particle}, \delta y_{particle})$. Perubahan pose dengan sampling zero mean gaussian $\hat{\delta} x = (\hat{\delta} x_{particle}, \hat{\delta} y_{particle}, \hat{\delta} \theta_{particle})$.

2.3. Measurement Model

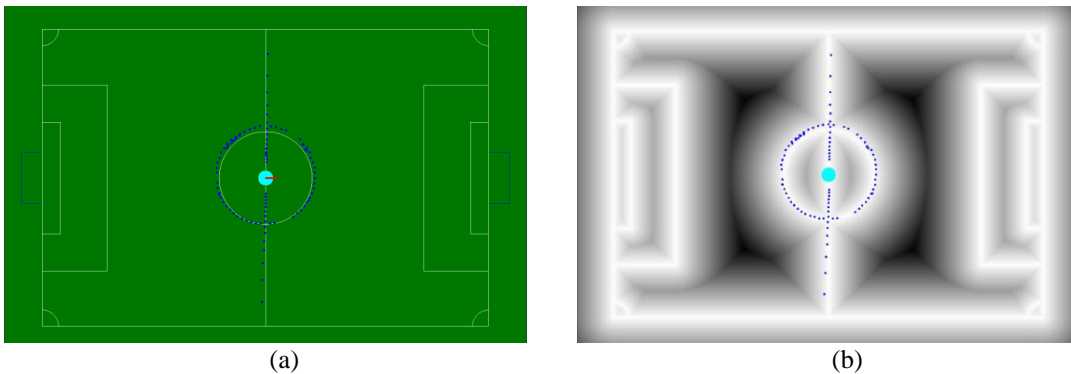
Likelihood fields merupakan salah satu algoritma yang digunakan untuk menentukan pose relatif pada lapangan dengan menghitung importance weight w dari particle [10], [11]. End points dari sensor z_t diproyeksikan menjadi koordinat global dari map. Oleh karena itu, diperlukan data koordinat global robot dan point sensor. $x_t(x \ y \ \theta)^T$ menyatakan pose robot pada waktu t . Koordinat lokal dari sensor terhadap robot dinotasikan dengan $(x_{k,sens} \ y_{k,sens})^T$. Data tersebut kemudian dipetakan menjadi koordinat global dengan matriks rotasi seperti pada persamaan (1) menghasilkan persamaan (2). Hasil perkalian matriks tersebut ditunjukkan pada persamaan (3).

$$\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \quad (1)$$

$$\begin{pmatrix} x_{z_t^k} \\ y_{z_t^k} \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x_{k,sens} \\ y_{k,sens} \end{pmatrix} \quad (2)$$

$$\begin{pmatrix} x_{z_t^k} \\ y_{z_t^k} \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} x_{k,sens}\cos\theta - y_{k,sens}\sin\theta \\ x_{k,sens}\sin\theta + y_{k,sens}\cos\theta \end{pmatrix} \quad (3)$$

Gambar 4 (a) menunjukkan visualisasi dari robot berwarna cyan yang berada di tengah-tengah lapangan dengan $measurement \ z_t^k$ berupa point-point berwarna biru yang telah dipetakan menjadi koordinat global. Sedangkan pada gambar 4 (b) merupakan visual dari array yang berisi $error-distribution$ yang digunakan untuk likelihood field dengan konfigurasi: lokasi semakin gelap, maka semakin kecil kemungkinan terdapat garis pada lokasi tersebut.



Gambar 4. (a) environment lapangan. (b) Likelihood field.

Gambar 5 merupakan algoritma untuk menghitung importance weight w dengan menggunakan likelihood field. Point-point sensor diproyeksikan menjadi posisi global pada baris 5-6 dengan persamaan (3). Kemudian, pada baris 7 mencari nilai minimum jarak dari point sensor (x, y) terhadap posisi (x', y') line point pada map. Pada baris 9 memperbarui nilai weight berdasarkan jumlah point dan nilai error yang diperoleh pada baris 8.

```

1: Algorithm likelihood_field_scan_line_point_model( $z_t, x_t, m$ ):
2:    $error = 0$ 
3:    $n = size(all\ k)$ 
4:   for all k do
5:      $x_{z_t^k} = x + x_{k,sens} \cos\theta - y_{k,sens} \sin\theta$ 
6:      $y_{z_t^k} = y + x_{k,sens} \sin\theta + y_{k,sens} \cos\theta$ 
7:      $dist = \min \left\{ \sqrt{(x_{z_t^k} - x')^2 + (y_{z_t^k} - y')^2} \mid (x', y') \text{ occupied in } m \right\}$ 
8:      $error = error + dist$ 
9:    $w = 1.0\ error^{-1} n^{-1}$ 
10:  return  $w$ 

```

Gambar 5. Algoritma pendeteksian garis dengan model likelihood field

2.4. Augmented Monte Carlo Localization

Augmented Monte Carlo Localization atau AMCL merupakan varian dari algoritma Monte Carlo Localization (MCL) untuk mengatasi masalah estimasi posisi dan orientasi (pose) pada robot otonom [12]–[16]. AMCL diperoleh dengan menggabungkan probabilistic motion dan perceptual models ke dalam algoritma particle filter.

```

1: Algorithm Augmented_MCL( $X_{t-1}, u_t, z_t, m$ ):
2:   static  $w_{slow}, w_{fast}$ 
3:    $X_t = X_t = \emptyset$ 
4:   for  $m = 1$  to  $M$  do
5:      $x_t^{[m]} = \text{sample\_motion\_model}(u_t, x_{t-1}^{[m]})$ 
6:      $w_t^{[m]} = \text{measurement\_model}(z_t, x_t^{[m]}, m)$ 
7:      $\bar{X}_t = \bar{X}_t + \langle x_t^{[m]}, w_t^{[m]} \rangle$ 
8:      $w_{avg} = w_{avg} + \frac{1}{M} w_t$ 
9:   endfor
10:   $w_{slow} = w_{slow} + \alpha_{slow} (w_{avg} - w_{slow})$ 
11:   $w_{fast} = w_{fast} + \alpha_{fast} (w_{avg} - w_{fast})$ 
12:  for  $m = 1$  to  $M$  do
13:    with probability  $\max\{0.0, 1.0 - w_{fast}/w_{slow}\}$  do
14:      add random pose to  $X_t$ 
15:    else
16:      draw  $j \in \{1, \dots, N\}$  with probability  $\propto w_t^{[j]}$ 
17:      add  $x_t^{[j]}$  to  $X_t$ 
18:    endwith
19:  endfor
20:  return  $X_t$ 

```

Gambar 6. Variasi MCL yang menambahkan sample acak. Banyaknya sample acak ditentukan dengan membandingkan w_{slow} dan w_{fast}

Gambar 6 menunjukkan algoritma AMCL yang merepresentasikan *the belief* $bel(x_t)$ dari banyak M partikel seperti pada persamaan (4). Baris 5 menunjukkan sample dari motion model dengan menggunakan partikel dari *present belief* sebagai point start. Kemudian, perhitungan measurement model pada baris 6 untuk menentukan importance weight dari partikel tersebut. Fungsi **motion_model** yang digunakan adalah odometry motion model yang terdapat pada gambar 3 dan **measurement_model** yang digunakan adalah likelihood field model yang terdapat pada gambar 5. **Augmented_MCL** menghitung empirical measurement likelihood pada baris 8, dan mempertahankan short-term dan long-term averages dari likelihood tersebut pada baris 10-11. Algoritma ini membutuhkan α_{slow} dan α_{fast} dengan ketentuan seperti pada persamaan (5). Parameter α_{slow} dan α_{fast} merupakan peluruhan untuk exponential filters yang memperkirakan rata-rata long-term dan short-term. Pada baris 13, ketika proses resampling, random sample ditambahkan dengan probability nilai tertinggi antara 0 dengan 1 dikurangi α_{fast} dibagi α_{slow} seperti pada persamaan (6). Sebaliknya, dilakukan resampling dengan *low variance resampling* seperti pada baris 16-17.

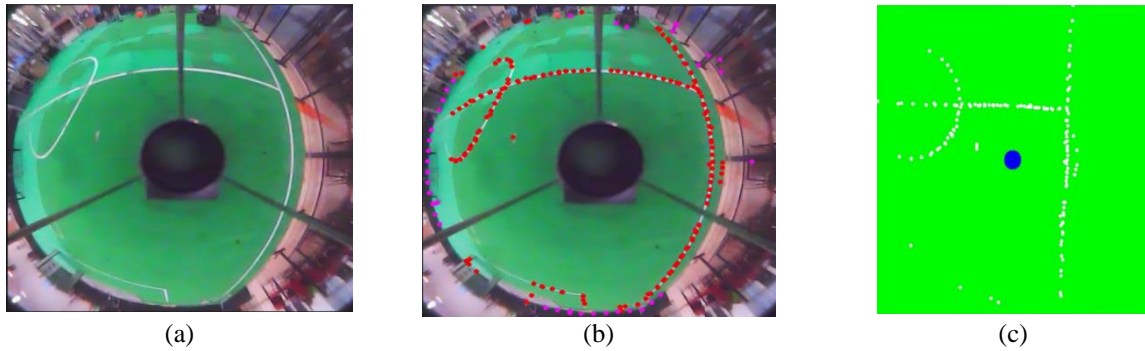
$$X_t = \{x_t^{[1]}, x_t^{[2]}, \dots, x_t^{[M]}\} \quad (4)$$

$$0 \leq \alpha_{slow} \ll \alpha_{fast} \quad (5)$$

$$\max\left\{0.0, 1.0 - \frac{w_{fast}}{w_{slow}}\right\} \quad (6)$$

3. RESULT AND DISCUSSION

3.1 Hasil Pendeteksian Garis



Gambar 7. hasil pendeteksian garis (a) raw image, (b) point-point dari garis dan (c) hasil regresi

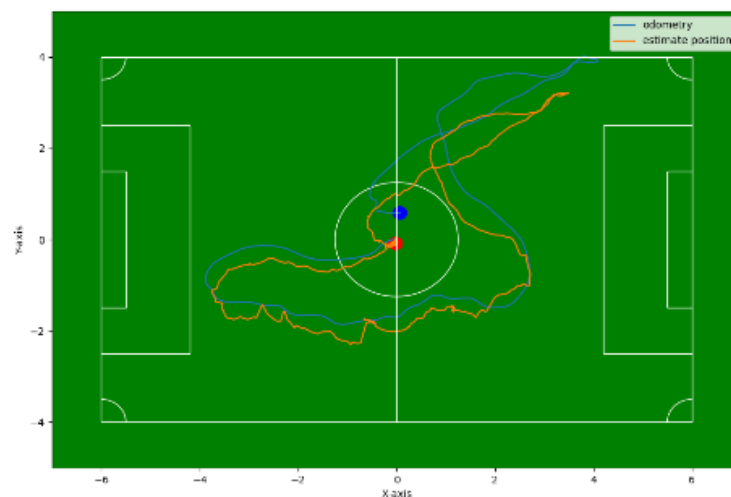
Pada gambar 7 (a) menunjukkan gambar yang dihasilkan oleh kamera omnivision. Dengan menggunakan *scanlines*, diperoleh informasi titik-titik yang didapatkan dari garis lapangan seperti pada gambar 7 (b). Titik-titik tersebut kemudian diinterpolasi agar sesuai dengan jarak sesungguhnya seperti pada gambar 7 (c).

3.2 Hasil Estimasi Posisi

Pada percobaan ini robot digerakkan secara manual dengan remote dari android. Robot digerakkan secara acak dengan kecepatan maksimum 1,2 m/s, kemudian kembali ke posisi awal. Setelah itu, dibandingkan posisi akhir dari odometry encoder dengan estimasi dari AMCL. Hasil percobaan dapat dilihat pada Tabel 1 dan gambar (8).

Table 1. Rata-rata error posisi robot terhadap posisi berhenti

Jumlah Partikel	Rata-rata Error (m)					
	x_odom	y_odom	theta_odom (rad)	x_est	y_est	theta_est (rad)
100	0.5910	0.6359	0.0465	0.05967	0.13225	0.2621
200	0.4701	0.7705	0.0603	0.06030	0.10035	0.0945
600	0.4159	0.9959	0.0590	0.05417	0.18417	0.0601
1000	0.54061	0.3304	0.0252	0.01922	0.05812	0.0597



Gambar 8. percobaan dengan 1000 partikel. Lingkaran biru: data odometry, lingkaran merah: hasil estimasi

Berdasarkan Tabel 1, dengan jumlah 100 partikel, hasil estimasi menghasilkan rata-rata error 0.059 m pada sumbu x, 0.132 m pada sumbu y dan 0.262 rad pada theta, sementara dengan odometry menghasilkan 0.591 m error pada sumbu x, 0.635 m error pada sumbu y dan 0.046 rad pada theta. Tingkat akurasi dari estimasi AMCL terus meningkat berbanding lurus dengan jumlah partikel yang digunakan. Hal ini terlihat pada percobaan dengan menggunakan 1000 partikel yang menghasilkan rata-rata error 0.0192 m pada sumbu x, 0.0581 m pada sumbu y dan

0.059 rad pada theta meskipun dengan error odometry 0.540 m pada sumbu x, 0.330 pada sumbu y dan 0.025 rad pada theta. Visualisasi dari percobaan dengan 1000 partikel ditunjukkan pada gambar 8.

4. CONCLUSION

Berdasarkan hasil yang didapat, AMCL dapat memberikan posisi yang optimal untuk lokalisasi pada robot sepakbola beroda dengan rentang error 0.01m pada sumbu x, 0,05m pada sumbu y dan 0,05 radian pada orientasi robot. Jumlah partikel optimal yang digunakan dari percobaan ini adalah 1000 partikel. Dengan jumlah partikel tersebut, menghasilkan error yang kecil. Penerapan algoritma ini dapat digabungkan dengan *traction control* daripada dengan kecepatan robot secara konstan se hingga tidak terjadi slip yang terlalu besar.


ACKNOWLEDGEMENTS

Penelitian ini didukung oleh tim BARELANG63 dari Politeknik Negeri Batam (POLIBATAM). BARELANG63 merupakan salah satu tim dari POLIBATAM yang berfokus mengembangkan robot *Middle-Size League* (MSL) beroda sejak 2017.

REFERENCES

- [1] H. Soebhakti *et al.*, "BARELANG 63 Team Description 2022".
- [2] B. Sugandi, S. Prayoga, I. A. Riandi, and D. G. Tinambunan, "Goal Detection and Opponent Avoidance Algorithm for Wheeled Robot Soccer using Color Filtering and Contour Extraction," in *2018 International Conference on Applied Engineering (ICAE)*, Batam: IEEE, Oct. 2018, pp. 1–5. doi: 10.1109/INCAE.2018.8579403.
- [3] H. Soebhakti, S. Prayoga, R. A. Fatekha, and M. B. Fashla, "The Real-Time Object Detection System on Mobile Soccer Robot using YOLO v3," in *2019 2nd International Conference on Applied Engineering (ICAE)*, Batam, Indonesia: IEEE, Oct. 2019, pp. 1–6. doi: 10.1109/ICAE47758.2019.9221734.
- [4] Y. Pratama, H. Soebhakti, S. Prayoga, and B. Budiana, "Localization System on Wheel Robot Soccer," in *Proceedings of the 5th International Conference on Applied Engineering, ICAE 2022, 5 October 2022, Batam, Indonesia*, Batam, Indonesia: EAI, 2023. doi: 10.4108/eai.5-10-2022.2327751.
- [5] A. J. R. Neves, A. Trifan, and B. Cunha, "UAVision: A Modular Time-Constrained Vision Library for Color-Coded Object Detection," in *Computational Modeling of Objects Presented in Images. Fundamentals, Methods, and Applications*, vol. 8641, Y. J. Zhang and J. M. R. S. Tavares, Eds., in Lecture Notes in Computer Science, vol. 8641, Cham: Springer International Publishing, 2014, pp. 351–362. doi: 10.1007/978-3-319-09994-1_35.
- [6] K. V. Utama, R. A. Fatekha, S. Prayoga, D. S. Pamungkas, and R. P. Hudhajanto, "Positioning and Maneuver of an Omnidirectional Robot Soccer," in *2018 International Conference on Applied Engineering (ICAE)*, Batam: IEEE, Oct. 2018, pp. 1–5. doi: 10.1109/INCAE.2018.8579148.
- [7] J. J. Kon, "Pose and Velocity Estimation for Soccer Robots".
- [8] Kok Seng Chong and L. Kleeman, "Accurate odometry and error modelling for a mobile robot," in *Proceedings of International Conference on Robotics and Automation*, Albuquerque, NM, USA: IEEE, 1997, pp. 2783–2788. doi: 10.1109/ROBOT.1997.606708.
- [9] M. A. Ismail, D. Purwanto, and A. Arifin, "Soccer Robot Localization Based on Sensor Fusion From Odometry and Omnivision," in *2022 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, Surabaya, Indonesia: IEEE, Jul. 2022, pp. 273–278. doi: 10.1109/ISITIA56226.2022.9855313.
- [10] S. Huang, H.-Z. Huang, Q. Zeng, and P. Huang, "A Robust 2D Lidar SLAM Method in Complex Environment," *Photonic Sens.*, vol. 12, no. 4, p. 220416, Dec. 2022, doi: 10.1007/s13320-022-0657-6.
- [11] A. Howard, M. J. Matarik, and G. S. Sukhatme, "Localization for mobile robot teams using maximum likelihood estimation," in *IEEE/RSJ International Conference on Intelligent Robots and System*, Lausanne, Switzerland: IEEE, 2002, pp. 434–439. doi: 10.1109/IRDS.2002.1041428.
- [12] S. Thrun, W. Burgard, and D. Fox, "PROBABILISTIC ROBOTICS".
- [13] F. R. Q. Aini, A. N. Jati, and U. Sunarya, "A study of Monte Carlo localization on robot operating system," in *2016 International Conference on Information Technology Systems and Innovation (ICITSI)*, Bandung - Bali, Indonesia: IEEE, Oct. 2016, pp. 1–6. doi: 10.1109/ICITSI.2016.7858235.
- [14] H. Abualkebash and H. Ocak, "Improved Global Localization and Resampling Techniques for Monte Carlo Localization Algorithm," *Int. J. Appl. Math. Electron. Comput.*, vol. 8, no. 3, pp. 102–108, Oct. 2020, doi: 10.18100/ijamec.800166.
- [15] A. Muzio, L. Aguiar, M. R. O. A. Maximo, and S. C. Pinto, "Monte Carlo Localization with Field Lines Observations for Simulated Humanoid Robotic Soccer," in *2016 XIII Latin American Robotics Symposium and IV Brazilian Robotics Symposium (LARS/SBR)*, Recife, Brazil: IEEE, Oct. 2016, pp. 334–339. doi: 10.1109/LARS-SBR.2016.63.
- [16] B. Sugandi and A. D. Zain, "Localization of Wheeled Soccer Robots Using Particle Filter Algorithm," in *2019 2nd International Conference on Applied Engineering (ICAE)*, Batam, Indonesia: IEEE, Oct. 2019, pp. 1–4. doi: 10.1109/ICAE47758.2019.9221670.

BIOGRAPHIES OF AUTHORS

 A portrait of a young man with dark hair, wearing a blue and white striped polo shirt, against a solid blue background.	<p>Mochamad Rizal Fauzi, lahir di Garut, Jawa Barat pada tanggal 01 September 2002. Sekarang sedang menjalani perkuliahan pada semester 7 di Politeknik Negeri Batam jurusan elektronika, prodi robotika. Sejak tahun 2020 aktif dalam tim robot Politeknik Negeri Batam. Pernah mengikuti kontes robot regional dan nasional pada tahun 2022 - 2023 di divisi KRSBI-beroda.</p>