Improved Navigation of Wheeled Soccer Robots Using Artificial Potential Field with Escape Force

Masdika Aliman[1], Anugerah Wibisana[1], Mochamad Rizal Fauzi[1], Danni Syahputra[1], and Kevin Indrawinata[1]

[1] Politeknik Negeri Batam, Kepulauan Riau 29461, Indonesia
masdikaaliman@gmail.com

Abstract. The Barelang63 team developed an autonomous wheeled soccer robot to participate in the Indonesian Robot Contest (KRI) in the KRSBI-Wheeled category, which independently requires object detection, motion planning, strategy, and communication capabilities. This research aims to apply the Artificial Potential Field (APF) method to the robot's navigation system, focusing on obstacle avoidance efficiency and local minima problem-solving. The approach includes an omnidirectional camera for image capture, the YOLOv11 model for object detection, and odometry, integrated with APF modified through escape force. Testing was performed on static and dynamic obstacle scenes. Results indicated a distance detection accuracy of 0.053 m according to Root Mean Square Error (RMSE). The system achieved a 100% success rate with static obstacles, averaging 7.5 seconds, outperforming the standard APF's 65%. In dynamic scenarios, success rates were 100% and 60% for two cases, whereas the standard APF only reached 40% and 10%. These findings demonstrate that escape force notably improves navigation, although further optimization is necessary.

Keywords: Soccer Robot, Barelang63, Artificial Potential Field, Escape Force.

1 Introduction

The Barelang63 team is dedicated to creating autonomous wheeled soccer robots that compete annually in the Indonesian Robot Contest (Kontes Robot Indonesia, KRI), organized by PUSPERNAS. [1]. One of the competition categories, KRSBI – Beroda (Indonesian Wheeled Soccer Robot Contest), tasked participants with creating robots that could play soccer autonomously, involving object detection, motion planning, strategy, and communication [2]. Localization is an important component in autonomous navigation [3][4]. Omnidirectional cameras are crucial in this situation because they can provide a 360° field of view without the need to move the camera, allowing the robot to identify objects in its surroundings thoroughly [5]. One of the main challenges in developing autonomous soccer robots is motion planning, which involves determining the best path for the robot to reach its goal while avoiding obstacles [6].

Several path planning techniques have been created to tackle this issue, such as the artificial potential field (APF). This method models the target as a source of attraction

H. Widiastuti et al. (eds.), *Proceedings of the 8th International Conference on Applied Engineering (ICAE 2025)*,
Advances in Engineering Research 290,
https://doi.org/10.2991/978-94-6463-982-7_33

force and obstacles as a source of repulsion force, which simultaneously forms an artificial potential field to guide the robot's movement [7][8]. The advantages of APF are its ability to perform real–time computation and its simplicity of implementation. However, this method has a weakness in that it tends to fall into local minima, so that the robot can get stuck in a specific position [9]. Several approaches have been taken to overcome this weakness, ranging from modifying the repulsive function, adding virtual target points, to applying an escape force function to avoid local minima [9][10][11].

This research aims to implement the artificial potential field method within a wheeled soccer robot navigation system, with an emphasis on enhancing obstacle avoidance efficiency and addressing local minima issues.

2 Method

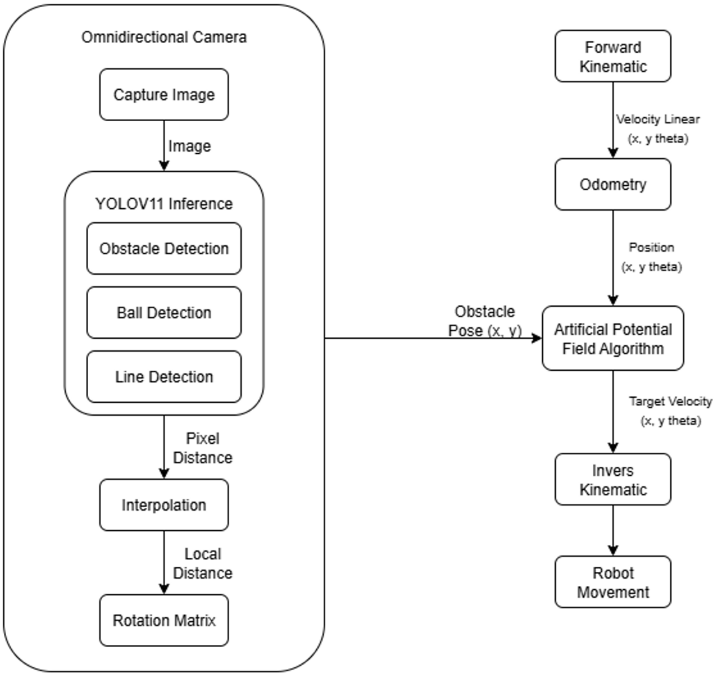


Fig. 1. System Diagram.

Figure 1 illustrates the software design block diagram used in the robot, which enables the artificial potential field algorithm to function. First, the omnidirectional camera captures images and processes visual data through object detection to detect obstacles, the ball, and field lines. The data pixel distance from the detection results is converted into local distance through interpolation and a rotation matrix. Second, forward kinematic processes take data from encoder sensors to determine the robot's movement and translate it into its position, known as odometry. Third, the artificial potential field algorithm uses pose odometry, obstacle poses, and target position as inputs to generate an optimal velocity direction. The output of APF is robot movement that enables the robot to navigate toward a target while independently avoiding obstacles. This software design will be explained in more detail as follows.

2.1 Obstacle Detection

Obstacle detection on omnidirectional cameras is performed using an object detection approach. In this study, the model employed is YOLOv11, which was developed by Ultralytics [12]. Each detection branch in the model ends with a convolutional layer connected to a detection layer, enabling it to generate predictions in the form of bounding boxes, object confidence scores, and class labels [13]. From these predictions, this study only utilizes the bounding boxes indicating obstacles. Next, it compares the bounding box's center point with the frame's center point to calculate the distance. This distance then serves as the basis for interpolation with the actual distance of the obstacle. The implementation of YOLOv11 can be seen in Figure 2.

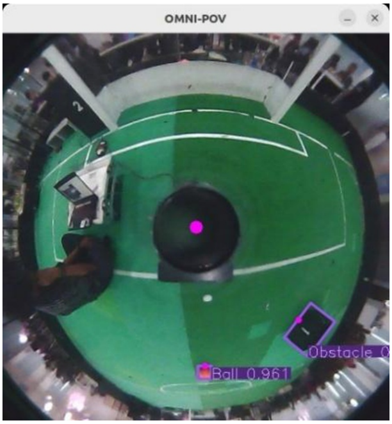


Fig. 2. Implementation of YOLOv11 for Obstacle Detection.

2.2 Robot Movement

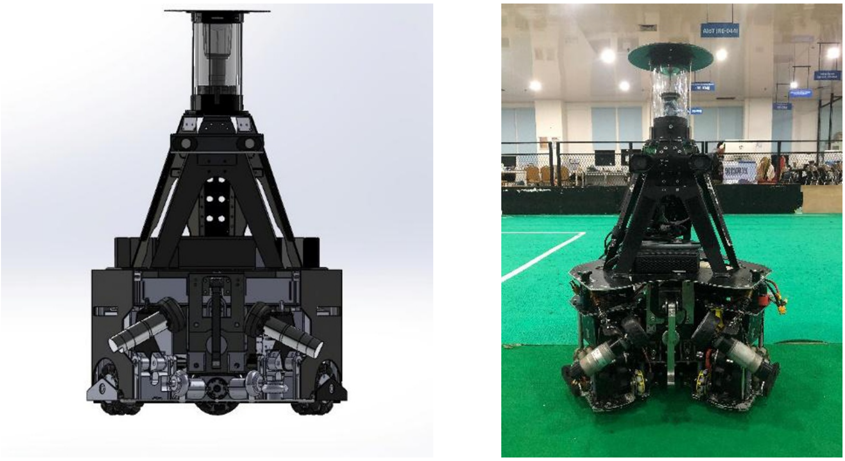

(a) (b)

Fig. 3. Barelang63 Robot Soccer.

The robot used in this study is a wheeled soccer robot designed with an omnidirectional drive mechanism. It includes a kicking mechanism powered by a solenoid and a ball control system driven by two PG36 DC motors. The primary drive system consists of three Maxon DC motors. Furthermore, the robot is equipped with both an omnidirectional camera and a stereo camera to enhance its visual perception.

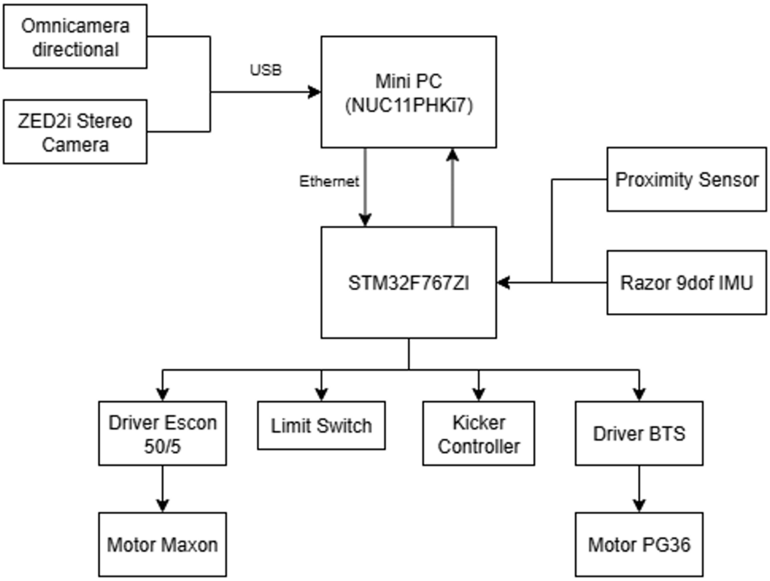


Fig. 4. Hardware block diagram.

To enhance our robot's movement, we applied kinematic analysis to guarantee accurate control and positioning. Kinematics is a discipline that studies the motion of robots, emphasizing the geometric configuration of moving reference frames, without considering the influence of forces or mass acting on the robotic system. Robot kinematics is used to analyze changes in the robot's position relative to global and local reference frames [14]. In robot kinematics, there are two primary methods: inverse kinematics and forward kinematics. Inverse kinematics calculates the necessary wheel speeds to achieve a set target position. On the other hand, forward kinematics calculates the robot's overall position from its linear and angular velocities [15].

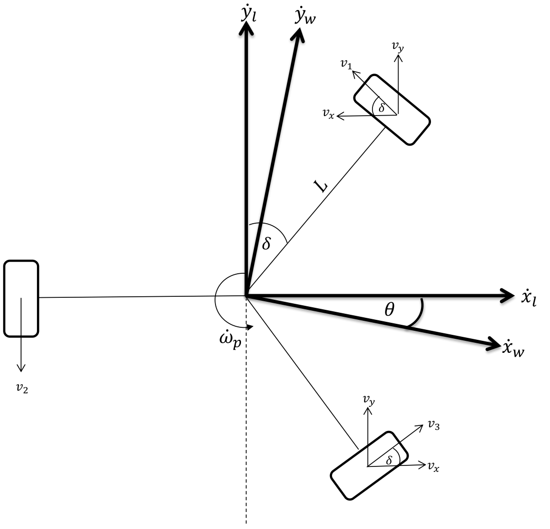


Fig. 5. Barelang63 Robot Wheel Configuration.

Figure 5 shows the configuration of a three-wheeled omnidirectional robot with wheels spaced at 120° intervals. This configuration allows the robot to move in any direction while maintaining its orientation. The inverse kinematic equations obtained from this design are presented below.

$$\begin{pmatrix}\omega_1\\ \omega_2\\ \omega_3\end{pmatrix} = \frac{1}{r}\begin{pmatrix}-\cos\vartheta & \sin\vartheta & L\\ 0 & -1 & L\\ \cos\vartheta & \sin\vartheta & L\end{pmatrix}\begin{pmatrix}\dot{x}_l\\ \dot{y}_l\\ \dot{\omega}_p\end{pmatrix} \tag{1}$$

Based on equation 1, ω represents the angular velocity of each wheel in radians per second (rad/s). L is the distance from the center of the robot to the center of the wheel in centimeters. $\dot{x}_l$ represents the local velocity of the robot along the x-axis in centimeters per second (cm/s), while $\dot{y}_l$ represents the local velocity of the robot along the y-axis in centimeters per second (cm/s). For forward kinematics, the robot's linear motion is calculated based on each wheel's linear velocity, which is derived from odometry using the following equation:

$$v_{\mathrm{i}} = \frac{(pulse\ enc - prev\ pulse\ enc)}{PPR}\, 2\,\pi\,\mathrm{r} \qquad (2)$$

Odometry is a method for estimating a robot's position by utilizing data from sensors on the wheels. The information generally includes wheel speed, orientation changes, and position displacement. The robot's position can be calculated using specific equations after obtaining the linear velocity values for each wheel from equation 2.

$$\begin{pmatrix} \dot{\mathrm{x}}_w \\ \dot{\mathrm{y}}_w \\ \dot{\omega}_p \end{pmatrix} = \left(\frac{1}{3}\right) \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & L & 1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} \qquad (3)$$

Based on equation 3, $\dot{\mathrm{x}}_w$ represents the global velocity of the robot along the x-axis in centimeters per second (cm/s), while $\dot{\mathrm{y}}_w$ represents the global velocity of the robot along the y-axis in centimeters per second (cm/s), and $\dot{\omega}_p$ represents the angular velocity of the robot in radians per second (rad/s). Once the robot's overall velocity is determined, its position can be found by integrating this velocity, as shown in the equation below.

$$\begin{aligned} X_{pos} &= X_{pos-1} + \dot{\mathrm{x}}_w \cdot \mathrm{t} \\ Y_{pos} &= Y_{pos-1} + \dot{\mathrm{y}}_w \cdot \mathrm{t} \\ \theta_n &= \theta_{n-1} + \dot{\omega}_p \cdot \mathrm{t} \end{aligned} \qquad (4)$$

2.3 Artificial Potential Field

Implementing artificial potential fields has become the primary approach in robot navigation. This concept assumes that the target and obstacle points generate two virtual forces. A visualization of an artificial potential field can be seen in Figure 6.

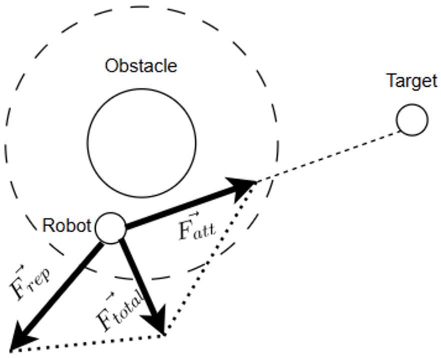


Fig. 6. Artificial Potential Field.

The attractive force pulls the robot toward the target destination. Let's say the robot's position is indicated as $p_r = (x_r\,,\ y_r)$ and $p_g = (x_g\,,\ y_g)$ as the target position to be reached. The direction vector to the destination is $\vec{d}_g = p_g - p_r$, with a distance of $d_g = \| \vec{d}_g \|$. Therefore, the attractive force equation $(\vec{F}_{att})$ is defined based on the distance threshold δ as follows:

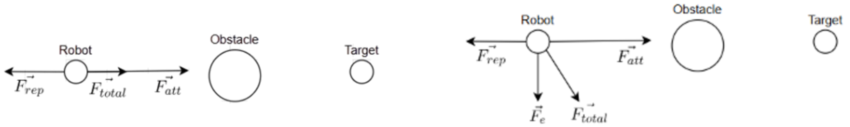
$$\vec{F}_{att} = \begin{cases} 0, & d_g < \delta \\ k_a \cdot \dfrac{\vec{d}_g}{\| \vec{d}_g \|}, & d_g \geq \delta \end{cases} \tag{5}$$

Where k_a is the attraction gain, and $\frac{\vec{d}_g}{\| \vec{d}_g \|}$ is the normalized direction vector. The value of the attractive force will decrease as the distance between the robot and the target point decreases.

Repulsive force is the force that pushes the robot away when it gets close to an obstacle. The closer the robot is to the obstacle, the stronger the repulsive force. However, when the distance between the robot and the obstacle exceeds the specified range threshold, the repulsive force can be expressed as equation 5:

$$\vec{F}_{rep} = \begin{cases} \dfrac{1}{2} \cdot k_o \cdot \left(\dfrac{1}{d_{obs}} - \dfrac{1}{r_{obs}} \right)^2, & d_{obs} < r_{obs} \\ 0, & d_{obs} > r_{obs} \end{cases} \tag{6}$$

Where k_o is the repulsive force gain d_{obs} is the distance from the obstacle to the robot, and r_{obs} is the obstacle threshold distance.



(a) (b)

Fig. 7. The effect of escape force: Local minima (a) and Adding Escape Force (b).

Figure 7 illustrates the influence of escape force on the local minima problem. In part (a), local minima occur in regions with zero potential value, indicating that the sum of the attractive and repulsive forces is zero. These minima are recognized when equations 7 and 8 are satisfied, with b and c being arbitrary constants [16]. To overcome this problem, as shown in section (b), an escape force $\vec{F}_e$ is added, placed perpendicular to the repulsive force $\vec{F}_{rep}$. The addition of this force serves to guide the robot away from local minima, as the total force $\vec{F}_{total}$ no longer points toward local minima, but instead pushes the robot to navigate around obstacles.

$$\frac{\|\vec{F}_{total}\|}{\|\vec{F}_{rep}\|} < b \tag{7}$$

$$\cos^{-1}\left(\frac{\vec{F}_{att} \cdot \vec{F}_{rep}}{\|\vec{F}_{att}\| \cdot \|\vec{F}_{rep}\|}\right) < c \tag{8}$$

Once these two criteria are satisfied, the obstacle nearest to the robot is used to determine the escape force. $\vec{F}_e$ using the equation:

$$\vec{F}_e = 10^{\log_{10}(\vec{F}_{rep})} \cdot \frac{1}{\left(d_{obs} + R_r + \left(\frac{{v_R}^2}{v_{max}}\right)\right)^2} \cdot \vec{m}_{rep\perp} \tag{9}$$

Where $\vec{m}_{rep\perp}$ is the unit vector perpendicular to the repulsive force $\vec{F}_{rep}$, the parameter R_r represents the robot's radius, v_R is the robot's velocity, and v_{max} indicates the robot's maximum velocity.

Next, referring to equations 5, 6, and 9, the total force affecting the robot's movement can be written as:

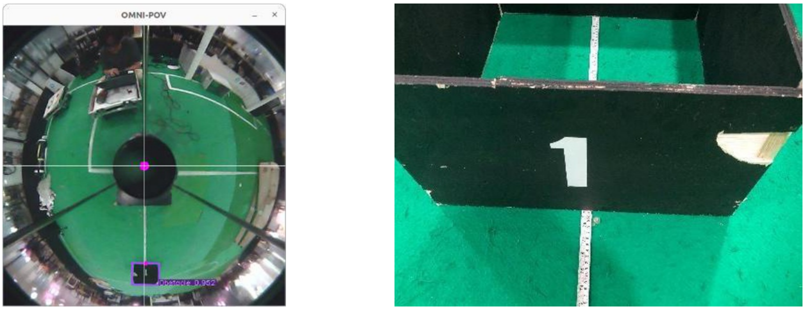
$$\vec{F}_{total} = \vec{F}_{att} + \vec{F}_{rep} + \vec{F}_e \tag{10}$$

Therefore, the total force $\vec{F}_{total}$ not only incorporates the attractive force $\vec{F}_{att}$ toward the goal, but also the repulsive force $\vec{F}_{rep}$ from obstacles, as well as the escape force $\vec{F}_e$, which ensures the robot can avoid local minima traps and move toward a safer and smoother path.

3 Result

3.1 Obstacle Detection Testing

This experiment was conducted by placing obstacles in specific positions, which were then detected by an omnidirectional camera. The test assessed the distance between the robot and obstacles by analyzing the camera's detection results, which were then compared with actual measurements taken using a measuring tape.



(a) (b)

Fig. 8. Comparison of actual distance measurements (b) with detection results (a).

Figure 8 illustrates the distance between the robot and the obstacle. In part (a), the distance detected by the omnidirectional camera is 1.97 meters, while the actual distance in part (b) is 2 meters.

Table 1. Comparison of actual obstacle distance with sensor results.

Obstacle Distance(meters)	
Actual Distance	**Detection Results**
0,5	0,4
1	0,98
1,5	1,492
2	1,97
2,5	2,48
3	2,91
3,5	3,48
4	3,95
RMSE	**0,0533**

Table 1 shows that the camera's distance detection closely matches actual distances, with an average error under 5%. For instance, at a proper distance of 3.0 meters, the camera measured 2.91 meters. The system's accuracy was assessed using the Root Mean Square Error (RMSE), which was found to be 0.053 meters. This demonstrates the high accuracy of the detection system in measuring obstacle distances, especially within the 0.5 to 4.0-meter range, though the error slightly increases for distances greater than 3 meters.

3.2 Parameter Setup

In this section, we determine the optimal values of the artificial potential field (APF) parameters to avoid static and dynamic obstacles. These parameters are sought through iterative experiments to ensure a balance between computational speed and navigation effectiveness. As shown in Table 2, we compare the configurations between the standard APF algorithm and APF with an additional escape force.

Table 2. Parameter Experiment.

Parameter	APF	APF + Escape Force
k_a	50	8
k_o	20	15
$r_o$bs	0.8 m	1 m
v_{max}	0.7 m/s	0.7 m/s
b	-	2
c	-	90

3.3 Avoiding Static Obstacles

In four cases, the experiment compared the standard APF algorithm and APF with an incorporated escape force in a wheeled soccer robot. In these scenarios, the robot operated within a field measuring 12 meters in length and 8 meters in width. It was required to navigate from its starting position to the target while avoiding static obstacles. The quantity and arrangement of these obstacles varied across each scenario, as detailed in Table 3. Each scenario was repeated five times to promote reliability, followed by calculations and analysis of the success rate and execution time for both algorithms. This test directly compared the smoothness of the trajectory, the reliability in avoiding local minima, and the overall efficiency of the two approaches.

Table 3. Static Obstacle Experiment Setup.

Case	Goal	Start	Obstacle Position
1	(-3,8, 0)	(3,0)	(0,0)
2	(-3,8, 0)	(3,0)	(0,1),(1,1),(1,0)
3	(-3,8, 0)	(3,0)	(0,0), (0, 1), (0, -1)
4	(-3,8, 0)	(3,0)	(2.5,0), (1,0.5), (1.5,-0.8).

The success rate in this experiment was determined with equation 11, which calculates the percentage of successful trials out of the total. A trial is deemed successful if the robot reaches the target position without collisions or getting stuck in local minima.

$$Success\ Rate\ (\%) = \frac{number\ of\ success}{total\ trials} \times 100 \tag{11}$$

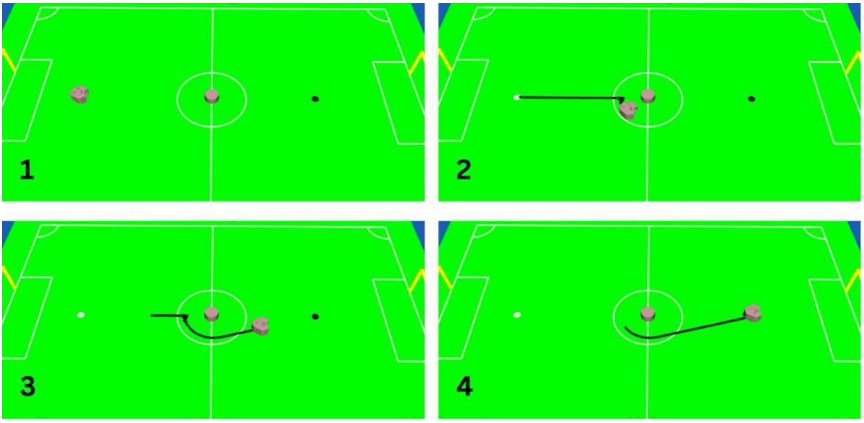


Fig. 9. Case 1 APF Standard.

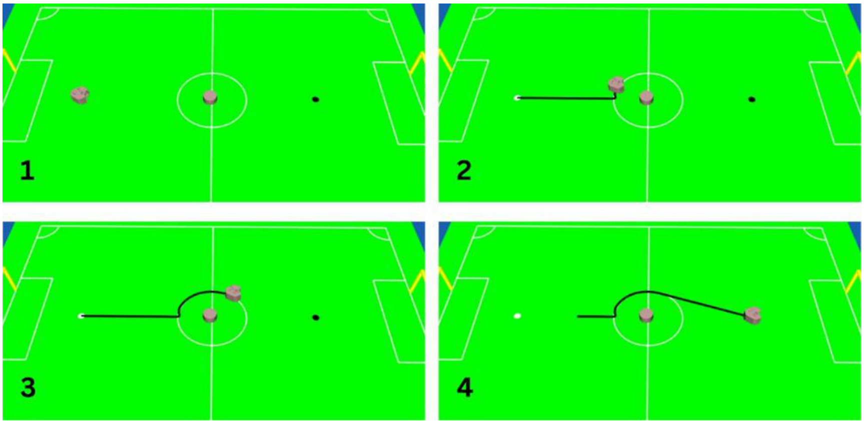


Fig. 10. Case 1 APF + Escape Force.

Figures 9 and 10 show Case 1, where the robot moves from the white point (start) to the black point (target) with one obstacle. The standard APF trajectory follows a relatively straight path with 100% success and takes about 8.118 seconds on average. However, APF with escape force creates a smoother trajectory, reaching the target in 6.54 seconds with similar success.

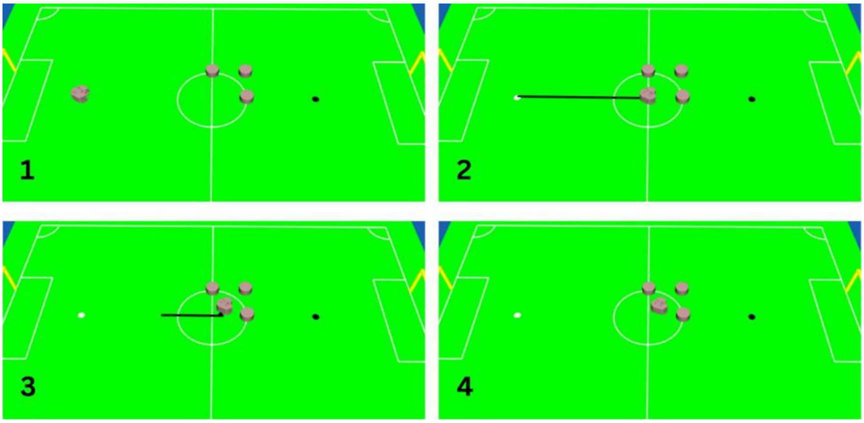


Fig. 11. Case 2 APF Standard.

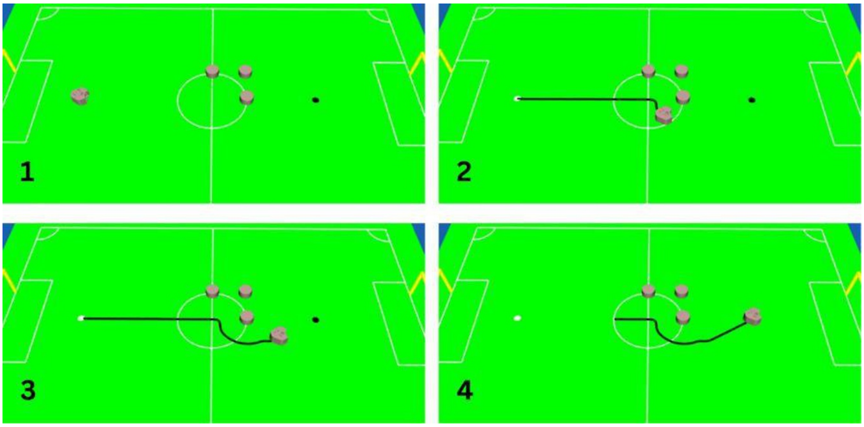


Fig. 12. Case 2 APF + Escape Force.

Figures 11 and 12 illustrate Case 2 with three static obstacles. The standard APF only succeeded 60% of the time, taking an average of 8.63 seconds, as it often got stuck in local minima. The APF with escape force reached 100% success in an average of 7.25 seconds, showing improved efficiency and stability of the trajectory.

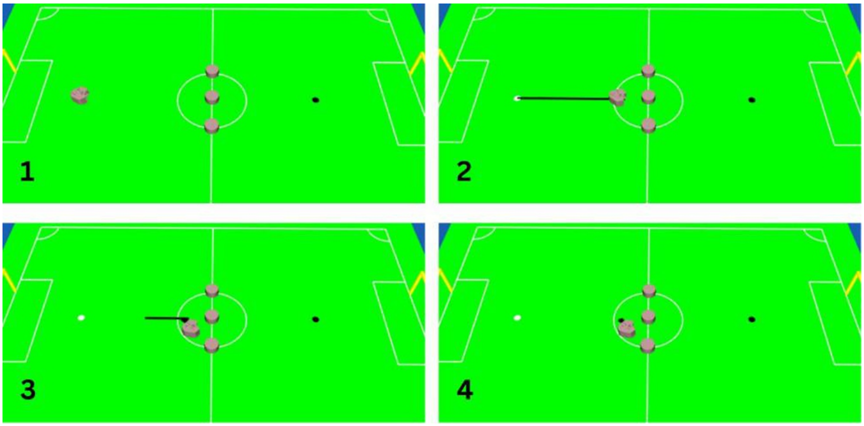


Fig. 13. Case 3 APF Standard.

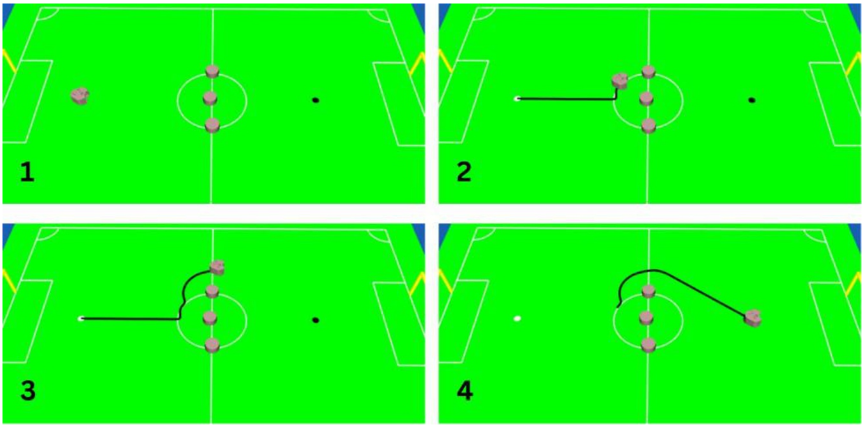


Fig. 14. Case 3 APF + Escape Force.

Figures 13 and 14 illustrate Case 3. The standard APF failed as the robot got stuck in a local minimum. Conversely, the APF with escape force always reached the target, with an average time of 8.044 seconds, following a circular path that successfully avoided traps.

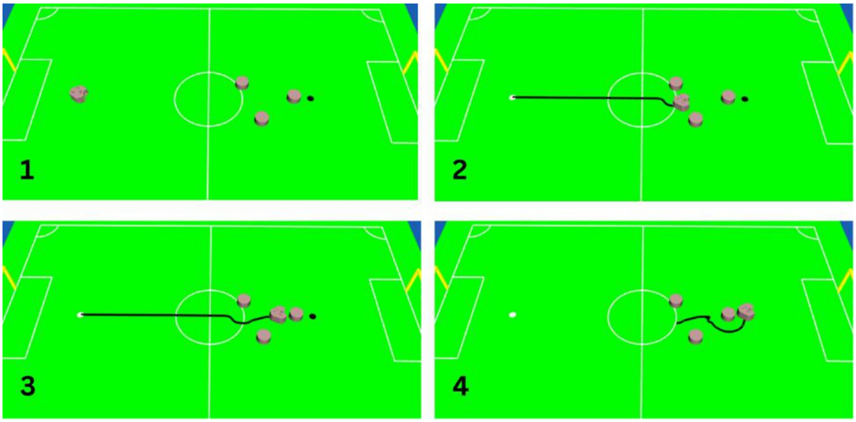


Fig. 15. Case 4 APF Standard.

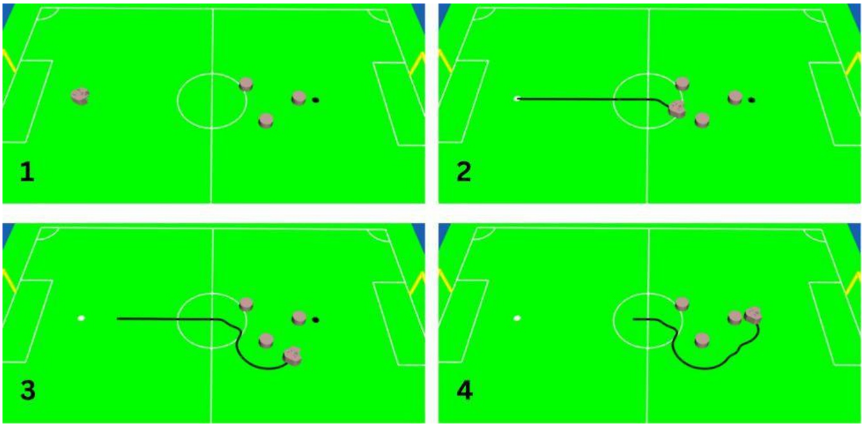


Fig. 16. Case 4 APF Standard + Escape Force.

Figures 15 and 16 illustrate Case 4. Both algorithms achieved 100% success, but the standard APF required an average time of 7.994 seconds with a slightly unstable trajectory. The APF with escape force was more stable, despite an average time of 8.17 seconds.

Table 4. Success Rate and Execution Time Comparison in Static Obstacle Avoidance.

Case	APF		APF + Escape Force	
	Success Rate (%)	Average Time (s)	Success Rate (%)	Average Time (s)
1	100	8.118	100	6.54
2	60	8.63	100	7.25
3	0	0	100	8.044
4	100	7.994	100	8.17

Based on Table 4, APF with escape force achieved a success rate of 100% in all cases, with an average time of 7.5 seconds, while standard APF was only successful in 65% of cases overall. This shows that adding escape force significantly improves the ability to overcome local minima without sacrificing time efficiency.

3.4 Avoiding Dynamic Obstacles

This section describes testing the artificial potential field algorithm to navigate a robot from its starting point to a target, avoiding dynamic obstacles in two cases, each repeated five times. In the first case, a single obstacle moved steadily towards the robot at 1.3 m/s. The second scenario involved two obstacles moving in random directions at speeds similar to the first case.

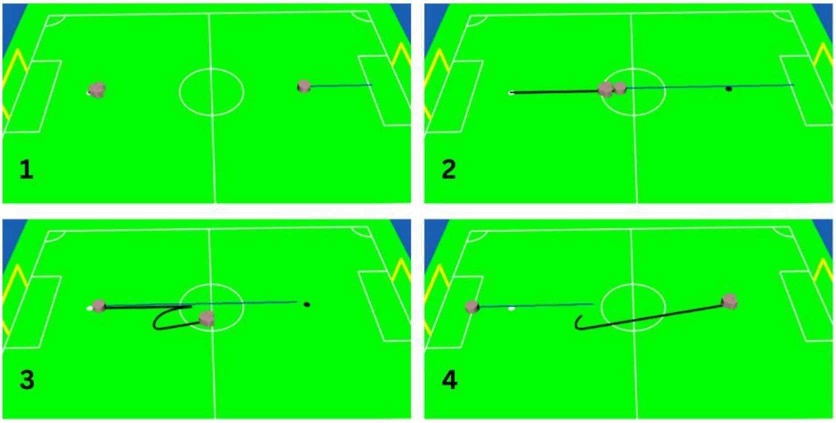


Fig. 17. Case 1 APF Standard Dynamic.

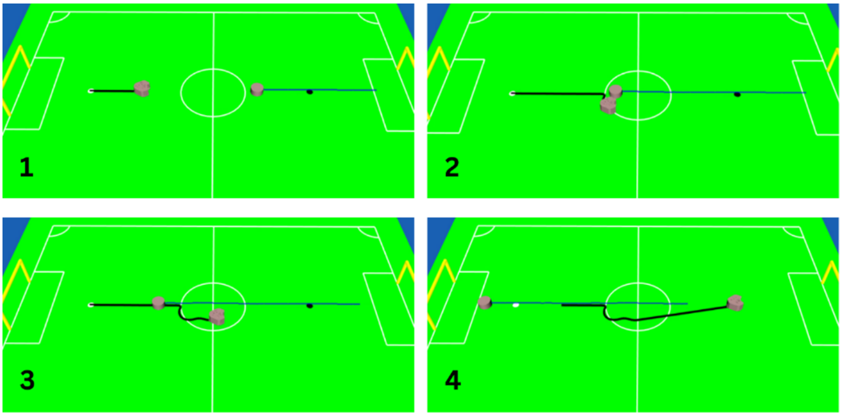


Fig. 18. Case 1 APF + Escape Force Dynamic.

Figures 17 and 18 illustrate Case 1, where a single obstacle moves linearly toward the robot. The standard APF trajectory shows three collisions out of 5 trials with an average time of 9.8 seconds. In contrast, APF with escape force achieves zero collisions out of

5 trials in 7.43 seconds, with a more adaptive trajectory that adjusts the repulsive force based on the obstacle's speed.

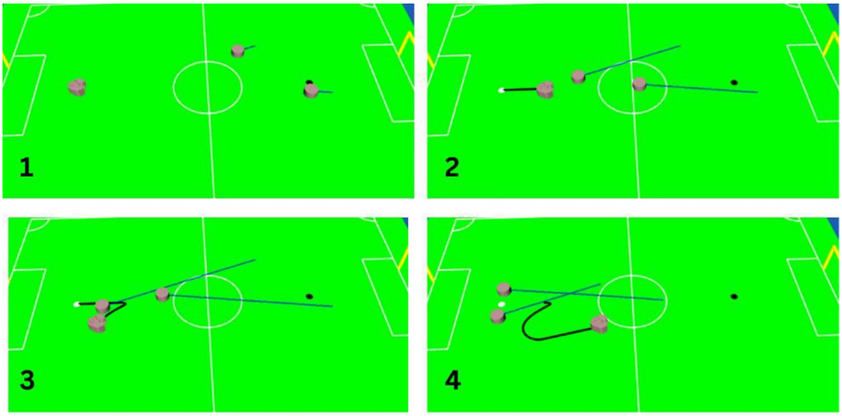


Fig. 19. Case 2 APF Standard Dynamic.

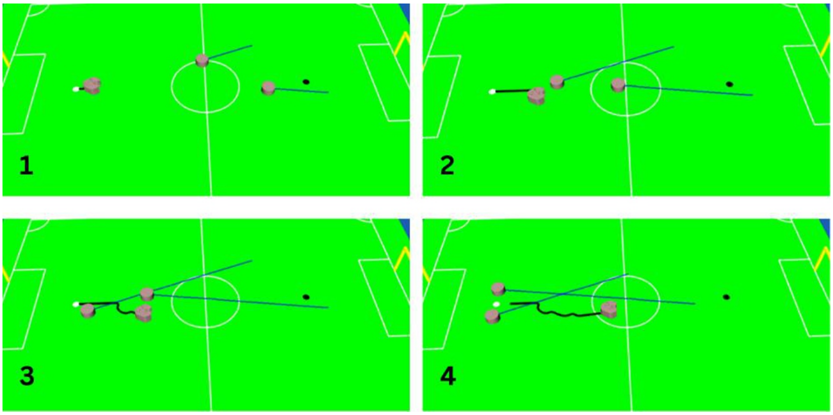


Fig. 20. Case 2 APF + Escape Force Dynamic.

Figures 19 and 20 illustrate Case 2, which involves two obstacles with random movement directions. In the standard APF, the robot experienced four collisions out of five trials with an average time of 15.87 seconds, due to limitations in responding to dynamic changes. Meanwhile, the APF with escape force recorded two collisions out of five trials with an average time of 11.73 seconds, confirming that escape force is effective in anticipating the direction of obstacle movement.

Table 5. Number of Collisions and Execution Time Comparison in Dynamic Obstacle Avoidance.

	APF			APF + Escape Force		
Case	Percentage of Safe Navigation	number of collisions	Average Time (s)	Percentage of Safe Navigation	number of collisions	Average Time (s)
1	40%	3 of 5	9.8	100%	0 of 5	7.43
2	10%	4 of 5	15.87	60%	2 of 5	11.73

Referring to Table 5, the standard APF algorithm completed only 40% of trials in case 1 and just 10% in case 2, with a high collision rate occurring 3–4 times out of 5 trials and longer travel durations. Conversely, incorporating escape force into APF significantly boosts performance, achieving 100% success without collisions in case 1 and 60% in case 2, along with faster average execution times of 7.43 seconds and 11.73 seconds. These findings demonstrate that adding escape force improves the robot's ability to navigate around dynamic obstacles, although collisions still happen in more complex scenarios.

4 Conclusion

This study effectively integrated the artificial potential field (APF) algorithm into the Barelang63 wheeled soccer robot navigation system. It combined odometry, YOLOv11-based object detection, and escape forces to solve the local minima problem. Obstacle detection tests showed high accuracy, with an RMSE of 0.053 m, confirming the conversion of visual data into local distance measurements. When avoiding static obstacles, the APF paired with escape force achieved a 100% success rate in all trials, with an average response time of 7.5 seconds. This approach significantly outperformed the standard APF, which had a success rate of 65% and often got stuck in local minima. For dynamic obstacle avoidance, the APF method with escape force demonstrates greater adaptability. In the first scenario, it achieved a 100% success rate without collisions, with an average time of 7.43 seconds. In contrast, standard APF succeeded only 40% of the time, with three collisions on average. In the more complex second case, the success rate of standard APF decreased to 10%, with four collisions, whereas APF with escape force improved to 60%, averaging two collisions and completing in 11.73 seconds. These findings indicate that the escape force can reduce collisions and speed up navigation. However, under more dynamic and complex conditions, the success rate remains suboptimal, and further refinement is needed. Therefore, further research needs to be directed at enriching the representation of the environment using virtual obstacles, which can help robots anticipate potentially trap areas, as well as obstacle velocity analysis to improve predictions and responses to the dynamics of opponent movements.

References

1. Soebhakti, H. et al.: BARELANG63 Team Description 2023 (2023)
2. Kusumoputro, B. et al.: Pedoman Kontes Robot Indonesia (KRI) Pendidikan Tinggi Tahun 2024. Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi, Jakarta (2024)
3. Fauzi, M.R., Soebhakti, H., Program, S., Engineering, R., Batam, P.N., Info, A.: Augmented Monte Carlo Localization for Wheeled Robot Soccer (2023)
4. Pratama, Y., Soebhakti, H., Prayoga, S., Budiana, B.: Localization System on Wheel Robot Soccer. In: 5th International Conference on Applied Engineering (ICAE 2022), pp. 1–10. EAI, Batam, Indonesia (2023). https://doi.org/10.4108/eai.5-10-2022.2327751
5. Soebhakti, H., Fakhruddin, H., Singarimbun, R., Musthofa, N.: Object Detection Using Omnidirectional Camera. In: 6th International Conference on Applied Engineering (ICAE 2023), pp. 1–11. EAI, Batam, Indonesia (2023). https://doi.org/10.4108/eai.7-11-2023.2342944
6. Soebhakti, H., Budiana, B., Rahmat, F., Aliman, M., Prebawa, N.: Implementation of Rapidly-Exploring Random Tree (RRT) for Path Planning and Stanley Control Method for Path Tracking on Wheeled Soccer Robots. In: 6th International Conference on Applied Engineering (ICAE 2023), pp. 1–13. EAI, Batam, Indonesia (2023). http://dx.doi.org/10.4108/eai.7-11-2023.2342945
7. Ge, S.S., Cui, Y.J.: Dynamic motion planning for mobile robots using potential field method. Auton. Robots 13(3), 207–222 (2002). https://doi.org/10.1023/a:1020564024509
8. Ananda, D.C., Dikairono, R., Purwanto, D.: Potential Field Path Planning pada Permainan Robot Sepak Bola Beroda. Jurnal Teknik ITS. 13(2) (2024). http://dx.doi.org/10.13140/RG.2.2.29120.96007/1
9. Lu, S.X., Li, E., Guo, R.: An Obstacles Avoidance Algorithm Based on Improved Artificial Potential Field. In: 2020 IEEE International Conference on Mechatronics and Automation (ICMA 2020), pp. 425–430. IEEE, Location (2020). https://doi.org/10.1109/ICMA49215.2020.9233866
10. Zheng, Y., Shao, X., Chen, Z., Zhang, J.: Improvements on the virtual obstacle method. International Journal of Advanced Robotic Systems. 17(2), 1–9 (2020). https://doi.org/10.1177/1729881420911763
11. Romme, J.C.T., Wolma, N.A., Naus, G.J.L., van de Molengraft, R.: Implementation of the Potential Field Method for motion planning at the TURTLEs. https://www.techunited.nl/media/files/MSL/BEP/BEP_Romme&Wolma.pdf, last accessed 2025/09/14
12. Jocher, G., Qiu, J.: Ultralytics YOLO11. https://github.com/ultralytics/ultralytics, last accessed 2025/09/14
13. Khanam, R., Hussain, M.: YOLOv11: An Overview of the Key Architectural Enhancements. http://arxiv.org/abs/2410.17725, last accessed 2025/09/14
14. Li, X., Zell, A.: Motion control of an omnidirectional mobile robot. Lect. Notes Electr. Eng. 24, 181–193 (2009). https://doi.org/10.1007/978-3-540-85640-5_14
15. Utama, K.V., Fatekha, R.A., Prayoga, S., Pamungkas, D.S., Hudhajanto, R.P.: Positioning and Maneuver of an Omnidirectional Robot Soccer. In: Proceedings of 2018 International Conference on Applied Engineering (ICAE 2018), pp. 1–5. IEEE, Location (2018). https://doi.org/10.1109/incae.2018.8579148
16. Wilschut, T.: An Obstacle Avoidance Algorithm for a Mobile Robot Based upon the Potential Field Method (2011)