Implementation of Intercept Ball Algorithm for Wheeled Soccer Robot Barelang63

Fahri Rahmat[1] , Anugerah Wibisana[2] , Mochamad Rizal Fauzi[3] , Masdika Aliman[4] and Nur Firmansyah[5]

[12345] Politeknik Negeri Batam, Kepulauan Riau 29461, Indonesia
fahri.rahmat46@gmail.com

Abstract. The Barelang63 team competes in the Indonesian Robot Contest, specifically in the Wheeled Indonesian Soccer Robot Contest (KRSBI-Beroda), where ball interception is critical for both offensive and defensive strategies. This research aims to improve the robot's ball interception capabilities by developing an algorithm that leverages a Kalman filter for precise estimation of the ball's position and velocity. The algorithm integrates data from both omnidirectional and stereo cameras, using color segmentation for ball detection. Experimental results demonstrate that the Kalman filter significantly enhances the accuracy of ball tracking, resulting in a high success rate in interceptions at lower ball speeds (0.8 - 1 m/s). However, at higher speeds (1.5 - 2 m/s), the robot's interception success decreases due to its speed limitations. Future work will focus on increasing the robot's maximum speed to enhance the effectiveness of the interception algorithm in high-speed scenarios.

Keywords: Soccer Robot, Barelang63, Ball Interception.

1 Introduction

The Barelang63 team focuses on developing autonomous wheeled soccer robots that compete annually in the Indonesian Robot Contest (Kontes Robot Indonesia, KRI), organized by PUSPRESNAS [1]. One of the categories contested in this competition is the Wheeled Indonesian Soccer Robot Contest (KRSBI-Beroda), which challenges teams to design robots capable of communication, coordination, and autonomous soccer play [2]. Ball possession is a critical factor in robot soccer, with the ability to intercept the ball playing a key role in both offensive and defensive strategies [3][4].

In previous implementations, robot passes were guided primarily by the receiving robot's position data obtained through odometry. However, odometry is prone to inaccuracies due to wheel slip [5], leading to mismatches between the estimated and actual positions of the receiving robot, often resulting in missed passes [6]. To address this issue, this research focuses on developing a ball interception algorithm aimed at improving the robot's ability to accurately control the ball, thereby enhancing overall game performance.

A key aspect of successful ball interception lies in the accurate estimation of the ball's position and velocity. The Kalman filter has demonstrated excellent performance

L. Lumombo et al. (eds.), *Proceedings of the 7th International Conference on Applied Engineering (ICAE 2024)*, Advances in Engineering Research 251,
https://doi.org/10.2991/978-94-6463-620-8_15

in this regard [7], providing reliable estimates based on noisy sensor data. Additionally, research [8] has utilized the Kalman filter to estimate the ball's velocity and employed geometric approaches for interception. However, there are notable differences in the types of robots used; in that study, the robots belong to the Small Size League (SSL), where the camera is positioned overhead and remains static. In contrast, the Barelang63 robot integrates a camera that moves with the robot, introducing unique challenges and opportunities for real-time processing and decision-making. In this work, the Kalman filter is utilized to predict the ball's trajectory based on camera inputs. Additionally, trigonometric calculations, slope analysis, and time prediction are employed to determine the optimal interception point for the robot. The proposed algorithm is expected to reduce missed passes and improve the team's chances of success in competitive matches.

2 Methods

Figure 1 illustrates the block diagram of the software design used in the robot, enabling the ball interception algorithm to function. First, the ball is detected using two types of cameras, and the system acquires data regarding the ball's position. This position data is then processed by the Kalman filter, which provides an estimated and filtered position, along with the ball's velocity. These estimates of position and velocity serve as inputs to the interception algorithm, which calculates the ideal target position for the robot. The robot then moves to this target position to intercept the ball's movement. The software design will be explained in further detail as follows.

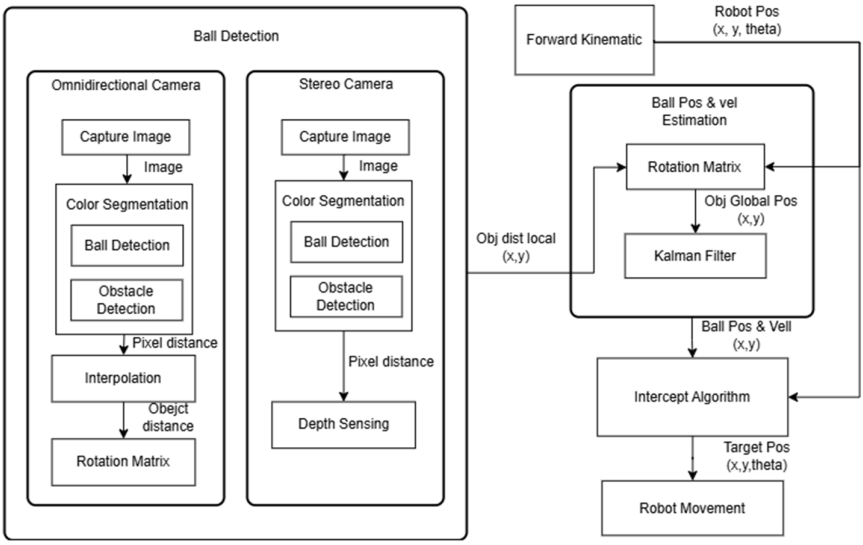


Fig. 1. System diagram.

2.1 Ball Detection

In this research, ball detection is performed using two types of cameras: a stereo camera and an omnidirectional camera.

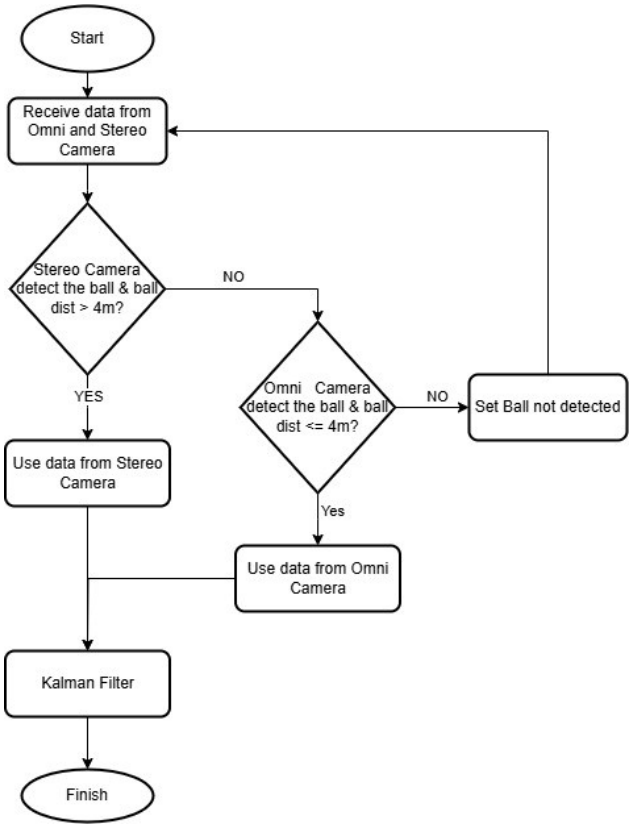


Fig. 2. Coordination between the two cameras.

Figure 2 shows the coordination between the two cameras used, the omnidirectional camera is used when the ball is within four meters of the robot, while the stereo camera is employed when the ball is located at a distance greater than four meters from the robot. This approach is necessary due to the distance limitations of the omnidirectional camera. To detect the ball, the color segmentation method is utilized. This is an image processing technique that separates or identifies objects in an image based on their color information. The technique leverages variations in the color spectrum, typically using the HSV (Hue, Saturation, Value) color model, to segment the image into blobs of color, allowing the computer to detect the objects [9].

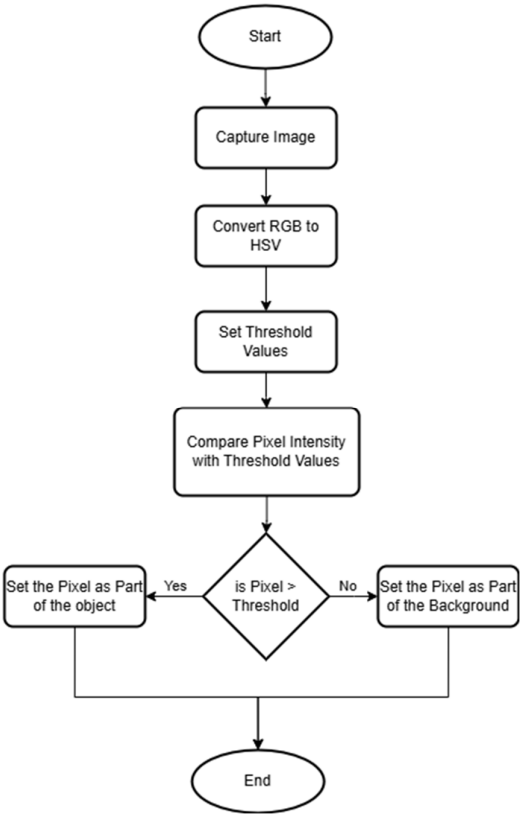


Fig. 3. Flowchart of color segmentation.

The flowchart on Figure 3 illustrates the color segmentation process to detect an object in the form of an orange ball. The first stage is image capture from the camera, which is then converted into an image in HSV (hue, saturation, value) color format. This conversion is done to simplify the segmentation process and produce more accurate results [10]. The next step is to determine the threshold value of HSV to detect the orange color. The value is then compared with each pixel in the image. If the intensity at a pixel is greater than the predetermined threshold value, the pixel will be considered part of the object. Conversely, if the intensity is lower, the pixel will be considered part of the background. After the camera detects the ball, distance data in the form of pixels is obtained. The data, in the program for the omni camera, will be processed using the interpolation method to produce the actual distance of the object to the robot. Meanwhile, the pixel distance data obtained from the stereo camera will be processed using the depth sensing method, which also produces the distance data of the object to the robot.

2.2 Ball Position and Velocity Estimation

The information about the position and velocity of the ball is very important in this game as the ball is the main object that affects key decisions [11]. Therefore, this information should be as accurate as possible. To achieve this, a Kalman filter is used to improve the estimated ball position obtained from visual information. Kalman filter is an algorithm used to estimate the state of a dynamic system that evolves over time in the presence of uncertainty or noise.

The ball is assumed to move at a constant speed between cycles. Although this is not entirely true due to the short time between cycles and the noisy environment, this assumption is quite reasonable. Therefore, factors such as friction and external control are not considered in the model. Based on the Kalman filter formula (described in [12][13]), the assumed state transition model given by

$$X_{k = FX_{k-1}} \quad (1)$$

Where X_k is the state vector that containing position and velocity of the ball, and F is a transition matrix with the following equation.

$$F = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \quad (2)$$

Where Δt is the difference between the current time and the previous measurement time. Kalman filter consists of two main stages, namely prediction and update, which in the prediction stage Kalman filter uses a mathematical model of the system to predict the next state of the system based on the current state. Then in the update stage the Kalman filter updates the prediction results with the latest measurements obtained from the sensors.

2.3 Intercept Algorithm

This algorithm aims to get the target position of the robot, so that the robot can intercept the movement of the ball.

```
1:  Algorithm intercept_ball(xb, yb, xr, yr, vrobot, vbx, vby):
2:      vrobot = 1.6
        flag = 1
3:      A = ( xb - xr )/( yb - yr )
4:      discriminant = -A²*vby²+2*A*vby*vbx-
                       vbx*vbx+vrobot²+A²*vrobot²
5:      if(discriminant > 0) then
6:          B = √discriminant
7:          alpha = atan2((A²*vby-vbx*A+B)/((1 + A²)*vrobot),
                  (A*B-A*vby+vbx) / ((1+A²)*vrobot))
8:          time = (yb - yr) / (vrobot * sin(alpha) - vby)
9:      else
10:         time = 0
11:         alpha = 0
12:         flag = 0
13:     if(flag = 1) then
14:         target_x = xr + vrobot * cos(alpha) * time
15:         target_y = yr + vrobot * sin(alpha) * time
16:         target_theta = atan2( yb - yr, xb - xr )
17:     else
18:         flag = 1
19:         time = 5
20:         target_x = xb + vbx * time
21:         target_y = yb + vby * time
22:         target_theta = atan2( yb - yr, xb - xr )
```

Fig. 4. Intercept Ball algorithm.

Figure 4 shows the ball interception algorithm (adapted from [14]) with information about the ball position (xb, yb), robot position (xr, yr), maximum robot speed (vrobot), and ball speed (vbx, vby). In line 2, the robot's maximum speed is initialized, along with the flag, which is set to 1. Line 3 contains the equation used to calculate the slope between the ball and the robot, which will be used to find the solution for the ball interception, as represented by the equation in line 4. If a solution to intercept the ball is found, the angle alpha, which indicates the direction the robot needs to move, and the time required for the robot to intercept the ball are computed using the equations in lines 6 through 8. However, if no solution for intercepting the ball is found, the values for time, alpha, and the flag are set to 0. If the flag remains 1 (indicating that a ball interception solution is found), the target position of the robot (x, y, theta) is calculated using the equations in lines 14 to 16. On the other hand, if the flag is 0, the robot's target position is calculated using the equations in lines 18 to 22.

2.4 Omniwheel Robot

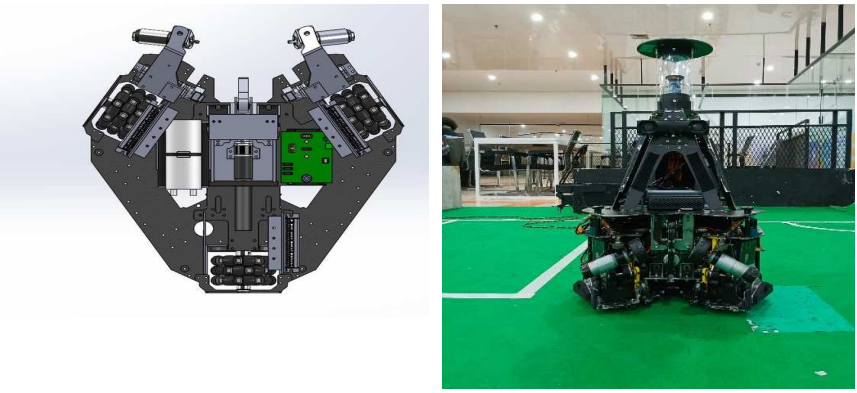


Fig. 5. Omniwheel robot.

The robot used in this research is an Omniwheel soccer robot. This robot is equipped with two types of cameras, namely omnidirectional cameras and stereo cameras. In addition, the robot is also equipped with a kicking system and a ball handling system. The kicker system uses a solenoid mechanism, while the ball handling system utilizes two PG 36 DC motors.

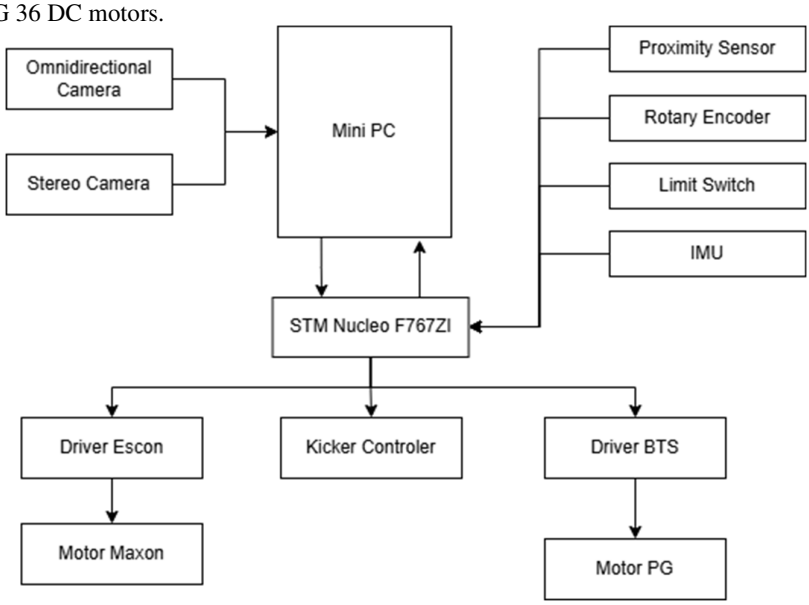


Fig. 6. Hardware block diagram.

For our robot's movement, we apply kinematic calculations to ensure precise control and positioning. Kinematics is the study of robot motion, focusing on the geometric structure of a moving reference frame system without considering the forces and masses acting on the robot. Robot kinematics is used to account for changes in the robot's position relative to both global and local reference frames [15]. In robot kinematics, there are two main models: inverse kinematics and forward kinematics. Inverse kinematics is used to determine the speed of each wheel so that the robot can reach a specified target point. Forward kinematics, on the other hand, is used to determine the robot's global position based on its linear and angular velocities [16].

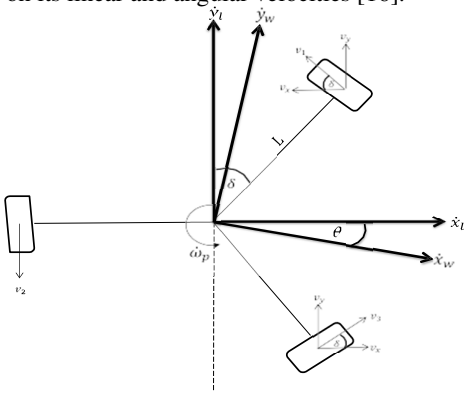


Fig. 7. Wheel configuration of the Barelang63 robot.

Figure 7 shows the wheel configuration of a three-wheeled omniwheel robot. Based on the figure, the inverse kinematics equations that can be used are as follows.

$$\begin{pmatrix} \emptyset_1 \\ \emptyset_2 \\ \emptyset_3 \end{pmatrix} = \frac{1}{r} \begin{pmatrix} -\cos\vartheta & \sin\vartheta & L \\ 0 & -1 & L \\ \cos\vartheta & \sin\vartheta & L \end{pmatrix} \begin{pmatrix} \dot{x}_l \\ \dot{y}_l \\ \omega_p \end{pmatrix} \tag{3}$$

Based on equation 3, $\emptyset$ 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). As for forward kinematics, the linear motion of the robot is determined based on the linear velocity of each wheel vi. Which vi is obtained from odometry with the following equation.

$$v_{\mathrm{i}} = \frac{\text{tick encoder}}{\text{ppr}} \, 2 \, \pi \, \mathrm{r} \tag{4}$$

Odometry is a technique used to determine the position of a robot based on data from sensors on the robot. This data usually includes information about wheel speed, roll angle, and position change. After obtaining the linear velocity value of each wheel, the following equation can be used to obtain the position of the robot.

$$\begin{pmatrix} \dot{x}_w \\ \dot{y}_w \\ \dot{\omega}_p \end{pmatrix} = \begin{pmatrix} \frac{1}{3} \end{pmatrix} \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & L & 1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} \quad (5)$$

Based on equation 5, $\dot{x}_w$ represents the robot's global velocity along the x-axis in centimeters per second (cm/s), while $\dot{y}w$ represents the robot's global velocity along the y-axis in centimeters per second (cm/s), and $\dot{\omega}p$ represents the robot's angular velocity in radians per second (rad/s). Once the robot's global velocity is determined, its global position can be obtained using the following equation.

$$X_{pos} = X_{pos-1} + \dot{x}_w \quad (6)$$

$$Y_{pos} = Y_{pos-1} + \dot{y}_w \quad (7)$$

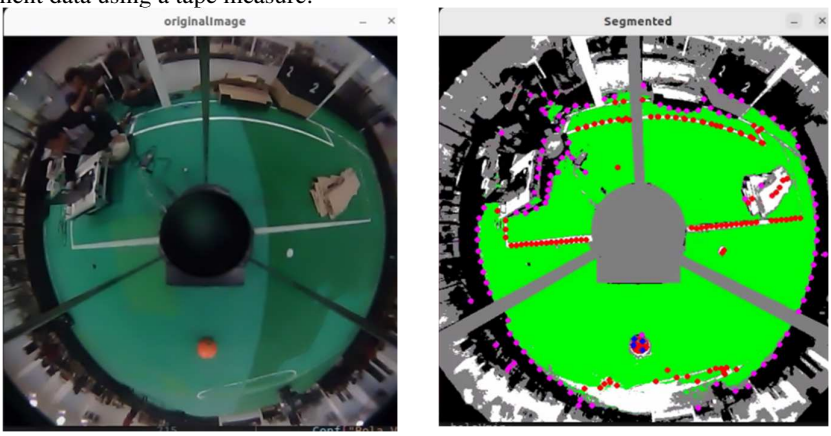
$$\theta_n = \theta_{n-1} + \dot{\omega}_p \quad (8)$$

3 Result

This chapter covers the ball detection test, the results of the Kalman filter, and the success rate of the ball intercept algorithm.

3.1 Ball Detection Testing

This test is carried out by placing the ball in a certain position which is then detected by two types of cameras, namely omnidirectional cameras and stereo cameras. In addition, testing was also carried out on the distance between the robot and the ball which was measured based on the results of camera detection, then compared with the measurement data using a tape measure.



(a) (b)

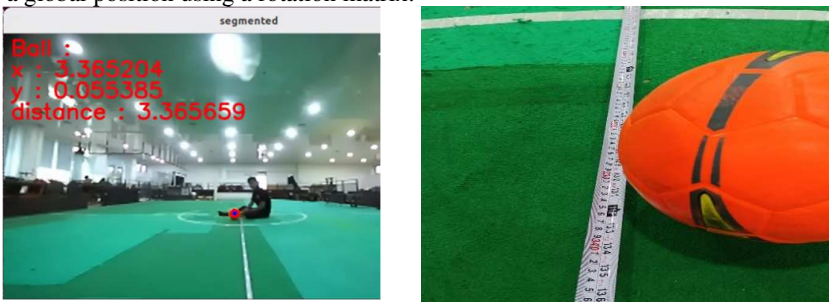
Fig. 8. Segmentation result image on omnidirectional camera.

Figure 8 shows the results of object segmentation from omnidirectional camera. Part (a) shows the original image taken by the omnidirectional camera, while part (b) shows the segmentation result where an orange colored ball was successfully detected. The detection of the ball is indicated by the blue colored spots surrounding the ball object.

(a) (b)

Fig. 9. Segmentation result image on stereo camera.

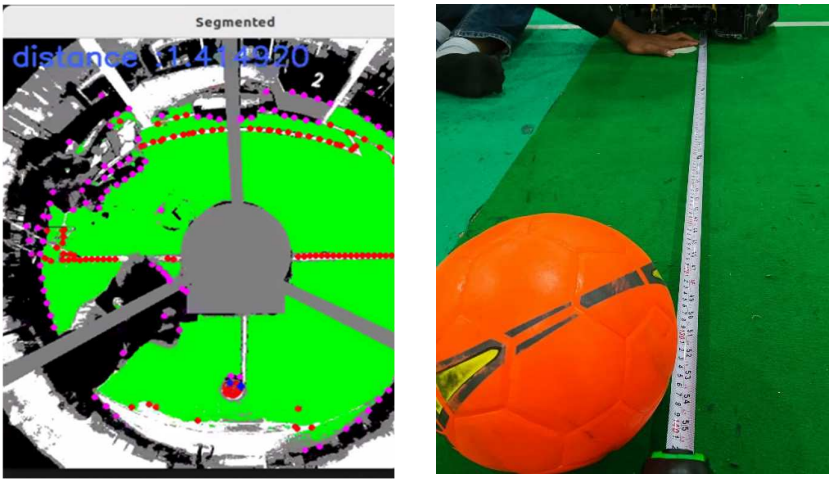
Figure 9 shows the object segmentation results from the stereo camera. Part (a) shows the original image taken by the stereo camera, while part (b) shows the segmentation result where an orange colored ball was successfully detected. The ball detection is indicated by a red circle and a blue dot on the ball object. To obtain the ball position data, two methods are used based on the type of camera used. The stereo camera applies the depth sensing method to measure the distance of the ball object to the robot. Meanwhile, in the omnidirectional camera, the distance of the robot to the ball object is obtained through the interpolation method. The distance data obtained is then converted into a global position using a rotation matrix.



(a) (b)

Fig. 10. Comparison of actual ball distance with ball detection results from stereo cameras.

Figure 10 shows the distance of the ball to the robot. In part (a), the distance detected by the ball using the stereo camera shows 3.36 meters. The actual distance measured using the tape measure is also close to this result, which is about 3.3 meters.



(a) (b)

Fig. 11. Comparison of actual ball distance with ball detection results from omnidirectional camera.

Figure 11 shows the distance of the ball to the robot. In part (a), the distance detected by the omnidirectional camera shows 1.41 meters. The actual distance measured using the tape measure also shows similar results, which is about 1.4 meters.

Table 1. Comparison of actual ball distance with sensor results

Ball Distance (meter)		
Actual	**Omni Camera**	**Stereo Camera**
0,82	0,81	0,8
1,12	1,09	1,1
1,53	1,49	1,5
1,77	1,79	1,8
2,13	2,09	2,1
2,37	2,38	2,4
2,63	2,63	2,6
2,78	2,82	2,8
2,82	2,91	2,9
3,1	2,99	3
RMSE	0.051	0.044

Based on Table 1, the evaluation compares the ball distance measurements obtained from two different cameras: an Omni Camera and a Stereo Camera. These measurements are compared against actual distance data measured using a tape measure, and

the Root Mean Square Error (RMSE) is calculated for both. The RMSE for the Omni Camera is 0.051, while the Stereo Camera achieves an RMSE of 0.044. These results indicate that the Stereo Camera provides a more accurate distance measurement compared to the Omni Camera. Specifically, the lower RMSE of the Stereo Camera suggests that its distance estimations are closer to the actual distances, resulting in less overall error.

3.2 Kalman Filter Testing

Kalman Filter testing is done by throwing a ball at the robot and observing the movement of the ball. The position data obtained is analyzed and compared with three sources: Omni data (from the omnidirectional camera), Stereo data (from the camera stereo sensor), and Kalman Filter data (Kalman Filter prediction results). In Figure 9, it can be seen that the prediction of the ball position by the Kalman Filter has high accuracy, with results that are almost in line with the stereo sensor data. The data from the omnidirectional sensor shows some deviation at some points along the trajectory of the ball. These test results indicate that the Kalman Filter is able to reduce noise and produce a more stable and accurate estimation of the ball position compared to other sensors.

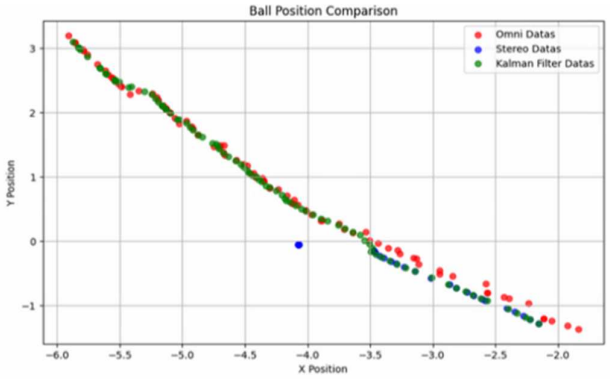


Fig. 12. Plot of Ball Movement.

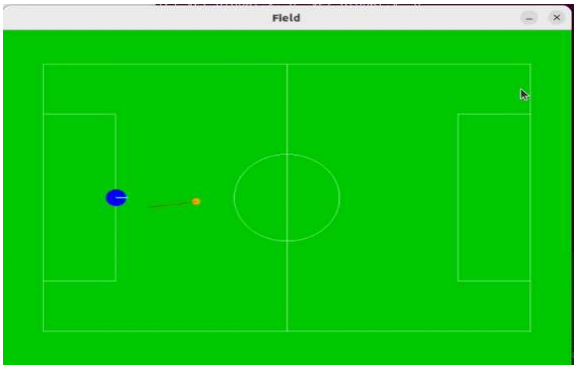


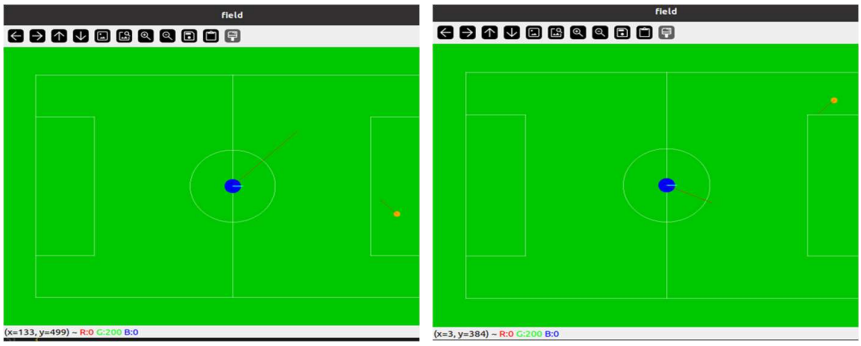
Fig. 13. Ball Movement.

In Figure 13, the blue circle shows the position of the robot, while an orange ball is detected moving towards the robot. The Kalman filter successfully estimates the velocity vector of the ball, which is visualized as a red line.

3.3 Intercept Ball Testing

The test was conducted using two striker robots. One robot takes the ball and functions as a kicker to pass the ball to the other robot. Testing was carried out 20 times, and then the percentage of success will be calculated.

$$Success\ Rate\ (\%) = \frac{Number\ of\ Successes}{Total\ Trials} \times 100 \tag{9}$$



(a) (b)

Fig. 14. Intercept ball algorithm test with simulation, the blue dot is the robot and the orange dot is the ball.

Figure 14 shows that the algorithm works well. In part (a), the robot position is at point (0,0) and the ball position is at point (5, -1) with ball speed (-0.5, 0.5). While in part

(b), the robot position is at point (0,0) and the ball position is at point (5, 3) with ball speed (-0.5, -0.5). It can be seen that the target movement of the robot (shown by the red line on the robot) successfully cuts off the movement of the ball shown by the red line on the ball.

Table 2. The results obtained in the interception tests performed on the robot.

Experiment	Ball Velocity (m/s)	Status		
		Success	Contact	Failed
1	0.8	100%	0%	0%
2	1	100%	0%	0%
3	1.5	80%	20%	0%
4	2	40%	60%	0%

The results of the intercept algorithm test conducted directly on the robot indicate that at ball velocities of 0.8 m/s and 1 m/s, the robot achieved a 100% success rate across five trials for each speed. However, at a velocity of 1.5 m/s, the success rate decreased to 80%, with the ball making contact with the robot's body in 20% of the trials. At the highest tested velocity of 2 m/s, the algorithm's performance further declined, resulting in three contacts, one failure, and only one successful interception.

4 Conclusion

In this paper, a ball interception algorithm was implemented on a robot using color segmentation-based ball detection with two types of cameras: an omni camera and a stereo camera. The omni camera produced an RMSE value of 0.051, while the stereo camera achieved an RMSE value of 0.044 in estimating the ball's distance compared to the actual distance. Subsequently, a Kalman filter was applied to smooth the data and obtain estimates of the ball's velocity. The data obtained were used as input for the ball interception algorithm, demonstrating high effectiveness with a success rate of 100% at speeds of 0.8 – 1 m/s. However, at a speed of 1.5 m/s, the success rate decreased to 80% due to the ball bouncing off the robot's body, and at a speed of 2 m/s, the success rate further dropped to 40% due to the robot's maximum speed limitation of only 1.6 m/s. For future work, it is recommended to enhance the robot's capabilities to achieve a higher maximum speed to improve the effectiveness of the ball interception algorithm.

References

1. Soebhakti, H., Prayoga, S., Amalya Fatekha, R., Pratama, Y., Krisna Prebawa, N., Dzaki Musthofa, N., Eka Kurniawan, W., Yusup, M., Suherryman, N., Revalba Ginggang, M., Sabiqul Himam, M., Hal Ata, T., Rizal Fauzi, M., Bahari, R., Afnan Fakhruddin, H., Rahmat, F., Pratama Singarimbun, R., Geovany Limbong, R., Imron Shodiq, M.: BARELANG63 Team Description 2023.

2. Pengembangan Talenta, B., Pusat, I., Nasional, P., Pendidikan, K., Teknologi, D.: PEDOMAN KONTES ROBOT INDONESIA (KRI) PENDIDIKAN TINGGI TAHUN 2024.
3. Cunha, J., Lau, N., Rodrigues, J.: LNAI 7416 - Ball Interception Behaviour in Robotic Soccer. (2011).
4. Baten, J.H.: Design and implementation of a generic intercept strategy for soccer robots. (2011).
5. Zainudin, Ahmad., Yunant, A.Ampuh.: Proceedings, IES 2019 : IES, International Electronics Symposium : Surabaya, Indonesia, September 27-28, 2019 : the Role of Techno-intelligence in Creating an Open Energy System Towards Energy Democracy. IEEE (2019).
6. Bachtiar, M.M., Hakim Ihsan, F.L., Wibowo, I.K., Wibowo, R.E.: Intercept Algorithm for Predicting the Position of Passing the Ball on Robot Soccer ERSOW. Inform : Jurnal Ilmiah Bidang Teknologi Informasi dan Komunikasi. 6, 35–39 (2021). https://doi.org/10.25139/inform.v6i1.3353.
7. Ferrein, A., Hermanns, L., Lakemeyer, G.: LNAI 4020 - Comparing Sensor Fusion Techniques for Ball Position Estimation. (2005).
8. Makarov, P.A., Yirtici, T., Akkaya, N., Aytac, E., Say, G., Burge, G., Yilmaz, B., Abiyev, R.H.: A Model-Free Algorithm of Moving Ball Interception by Holonomic Robot Using Geometric Approach.
9. Yoga Budi Putranto, B., Hapsari, W., Wijana, K., Kristen Duta Wacana Yogyakarta, U.: SEGMENTASI WARNA CITRA DENGAN DETEKSI WARNA HSV UNTUK MENDETEKSI OBJEK.
10. Hema, D., Kannan, Dr.S.: Interactive Color Image Segmentation using HSV Color Space. Science & Technology Journal. 7, 37–41 (2019). https://doi.org/10.22232/stj.2019.07.01.05.
11. Rocha Silva, A., Rocha Olivi, L., Barros Costa, E., Cavalcanti Alves Vilas Boas, A.S.: Dynamic Modeling and Stochastic Prediction of the Robot's Soccer Ball Trajectory. Presented at the December 27 (2019). https://doi.org/10.17648/sbai-2019-111566.
12. Silva, J., Lau, N., Rodrigues, J., Azevedo, J.L.: Ball sensor fusion and ball interception behaviours for a Robotic Soccer Team.
13. Bachtiar, M.M., Wibowo, I.K., Dikarinata, R., Priambudi, R.A., Anwar, K.: Dynamic Local Ball Tracking in Middle Size League Robot Soccer ERSOW based on Kaiman Filter. In: IES 2020 - International Electronics Symposium: The Role of Autonomous and Intelligent Systems for Human Life and Comfort. pp. 253–259. Institute of Electrical and Electronics Engineers Inc. (2020). https://doi.org/10.1109/IES50839.2020.9231877.
14. Tech United Eindhoven Team: Turtle3 Repository, https://gitlab.tue.nl/tech-united-eindhoven/Turtle3, (last accessed: 25 August 2024).
15. Priambudi, R.A., Wibowo, I.K., Mobed Bachtiar, M.: Penentuan Posisi Menggunakan Odometry Omniwheel. ISRSC 2018.
16. Proceedings of the 2018 International Conference on Applied Engineering (ICAE) : Batam, Indonesia, October 3-4, 2018. IEEE (2018).