Jurnal Integrasi
Vol. 14 No. 2, Oktober 2022, 81-91
e-ISSN: 2548-9828

Article History
Received September 2022
Accepted October 2022

Real-time Coordinate Estimation for Self-Localization of the Humanoid Robot Soccer BarelangFC

Susanto, Taufiq Tegar Pratama and Riska Analia

Batam Polytechnics
Electrical Engineering Study Program
Parkway Street, Batam Centre, Batam 29461, Indonesia
E-mail: susanto@polibatam.ac.id, riskaanalia@polibatam.ac.id

Abstrak

Dalam implementasinya, robot humanoid soccer terdiri lebih dari tiga robot di lapangan ketika sedang bermain bola. Semua robot diharapkan dapat memainkan sepak bola seperti manusia seperti mencari, mengejar, menggiring bola dan menendang bola. Untuk melakukan semua perintah tersebut, diperlukan sistem lokalisasi real-time sehingga setiap robot tidak hanya memahami posisi robotnya sendiri tetapi juga robot-robot lain bahkan objek yang berada di sekitar lapangan. Namun dalam implementasi real-time dan karena keterbatasan kemampuan komputasi robot, diperlukan suatu metode komputasi yang cepat dan mampu menghemat banyak memori. Oleh karena itu, dalam makalah ini menyajikan metode implementasi lokalisasi real-time dengan menggunakan metode odometry and Monte Carlo Localization (MCL). Untuk memverifikasi kinerja metode ini, beberapa percobaan telah dilakukan dalam aplikasi *real-time*. Dari hasil percobaan, metode yang diusulkan mampu mengestimasi koordinat posisi robot pada posisi X dan Y di lapangan ketika sedang bermain bola.

Kata kunci: Humanoid robot soccer, Localization, Odometry, Monte Carlo Localization (MCL)

Abstract

In implementation, of the humanoid robot soccer consists of more than three robots when played soccer on the field. All the robots needed to be played the soccer as human done such as seeking, chasing, dribbling and kicking the ball. To do all of these commands, it is required a real-time localization system so that each robot will understand not only the robot position itself but also the other robots and even the object on the field's environment. However, in real-time implementation and due to the limited ability of the robot computation, it is necessary to determine a method which has fast computation and able to save much memory. Therefore, in this paper we presented a real-time localization implementation method using the odometry and Monte Carlo Localization (MCL) method. In order to verify the performance of this method, some experiment has been carried out in real-time application. From the experimental result, the proposed method able to estimate the coordinate of each robot position in X and Y position on the field.

Keywords: Humanoid robot soccer, Localization, Odometry, Monte Carlo Localization (MCL)

1. Introduction

Humanoid robot soccer is designed to have ability on playing soccer like human being. In maneuvering on the field, the robot should be agile to track and dribble the ball, distinguish opponent and ally goal position, pass the ball to friends, and then score a goal. All of these movements are absolutely supported by the excellent robot navigation and vision system. The navigation system is hopped to make the robot able to maneuver it-self and do self-localization when the robot is playing football on the field. In many years, a lot of researchers designed and implemented the method to establish the navigation system of the humanoid robot soccer in different approach. As presented in [1], they developed and designed an interactive interface to optimize sound source localization (SSL) by using the microphone array to order the robot do the localization based on SSL command. While in [2], they constructed the framework to solve the trade-off problems of the robot localization and achieve the perception-during-Traversing Model system. This framework will be used as subdividing and re-integrating the localization

and recognition processes based on task-oriented frequency. Beside developing the localization based on sound and constructed the framework, many methods which have been developed by the researches utilize the sensor for developing an accurate robot pose localization. The pose estimation presented by [3] on the other hand, introduced the theoretical contribution strengthening the formulation of the IMU pre-integration to avoid ambiguity. Fallón, et.al [4] introduced the algorithm probabilistic for sensor fusion data from inertial, kinematic and LIDAR to produce a consistent position estimation for robot walking. The LIDAR also been used to assign localization which is able to maintain reliable drift-free using a Gaussian Particle Filter. Moreover, on [5] used the combination of the LIDAR sensor and odometry to estimate the mapping localization based on the kinematics-inertial of the humanoid robot. The SLAM, a Simulated Localization and Mapping method for the humanoid robot localization also can be approached by combining with the RGB-D camera. The RGB-D camera is used to construct the environment in visual localization [6-8].

In contrast with [9-10], they developed the visual memory based on topological map for autonomous navigation scheme for path planning and path following. In this work, they also used the RGB-D camera which is mounted on the NAO robot. The other work used the vision system for the self-localization by making the landmark field to determine the coordinate position of the robot. The landmark which is used as the point reference lays on the corner pole and goal [11-13]. The vision based path planning also developed by Mahdi, et.al [14], which is obtained the unknown environment from sensory and vision information and then approach the decision by using the Fuzzy Markov Decision Processes (FMDP). The indoor hybrid map based localization by combining a camera and a laser range finder, was introduced by Xu, et.al [15]. In this work, they constructed the topological map by natural landmarks, and local metrical maps which determined by using the improved Rao-Blackwellized particle filter. While Tian, et.al [16] provide an implementation of the particle filter algorithm for self-localization by employing a fish-eye camera lens. In this work also introduced some series solution in order to oscillation effect caused by the locomotion movement of the robot. And in [17], they used a Kinect sensor to obtain the digital map of the global environment and the position and orientation of the head of the robot. The position and orientation was used to determine the foot placement and calculated based on the spatial geometric relationship between head and the target foot. Moreover, the 3D environment localization model is able to generated by using the depth camera. The depth camera is able to maintain a 3D environment representation, and estimate the robot pose. Which is produced an optimal result when the head angle of the humanoid robot is around 20° and 29° [18-19].

In [20-21], they proposed a vision-based odometric system for robot localization. The position and orientation of the torso are predicted based on the differential kinematic map and apply the prediction-correction structure method of an Extended Kalman Filter. And then, they presented an averaged motion for canceling the sway oscillation of the torso motion. The other visual odometry (VO) method for localization as presented by Minami, et.al [22] included the three different sparse VO algorithm for a function of step length such as a direct, a semi-direct and an indirect algorithm for improving the footstep planning speed. Using the depth camera, stereo camera, or visual odometery are able to improve the precision, however it needs an expensive sensors and request higher computational capacity. Therefore, Carrillo, et.al [23] proposed a gentle algorithm to locate the robot localization by using the odometry information and general regression with Nadayara-Watson kernel. A robust learning method for the robot localization called Lightweight Humanoid robot Odometric Learning method (LHOL) presented by Saeedvand, et.al [24]. In this approached, they used the kinematic computational data, IMU (roll and pitch) data, and robot's actuator load data as the input of the artificial neural network method.

Moreover, the usage of the vision for the localization is presented the other method so called the Monte Carlo Localization (MCL). This localization able to deal with the limited landmarks given by the field. There are many methods of implementing the MCL algorithm on the robot. As presented in [25], they proposed a state-driven MCL which is able to deal with kidnap problem by divided the states of particles into four types: messy, approach, cluster and error. Nagi, et.al [26], on the other hand, used the MCL algorithm to deal with the limited landmarks such as yellow goal post and field marks. In this work they improve the resampling step and then process the estimation localization. In [27], the MCL has been done by virtual robot agents to estimate the global pose of the robot and added the filtering algorithm and stochastic modeling to deal with kidnapping problem. The same as [27], Almeida, et.al [28] also validated the MCL algorithm by simulation. However, in this work they proposed motion and observed models design for the domain to determine the quality of particle needed. In [29], the MCL was integrated with localization information retrieved by 2D RGB image and then combine with simulation data to the pose estimation. The MCL algorithm estimate the position and orientation of the robot used the known map of the environment, range sensor data, and odometry. Therefore, we also implemented the MCL algorithm to our humanoid robot soccer called BarelangFC. In our work, we proposed the MCL algorithm by understanding the heading of the robot from yaw IMU sensor and spread the particle around the field. In

order to implement the real-time localization, we transform the field into grid coordinate to estimate the position of the robot.

2. BarelangFC Design

In order to implement the proposed algorithm, we designed a humanoid robot named as BarelangFC which design presented on **Figure 1** with has dimension about 66 cm x 28 cm. The robot consists of 20 units of servo motor as a robot's actuator. The servo motor placed on each joint of the robot and generated 20 degree of freedom (DOF), where 12 DOF at the leg joint, 3 DOF for each arm joint, and 2 DOF at the neck joint. The robot equipped with webcam camera for the vision system, NVIDIA Jetson TX1 for the vision system generator, in order to command the servo motors the robot also has the servo controllers and also for the main controller we used the intel NUC. All the placement of the hardware which been used in this work presented on **Figure 2**. The number on the robot body which presented on **Figure 2** represented the servo ID for simplify the command send by the servo controller.

The whole block diagram system of the BarelangFC figured on **Figure 3**, at first the webcam camera will capture the object surrounded then the raw image used as the input of vision system on the NVIDA Jetson TX1 to generate the ball and goal detection. Beside detecting the ball and goal, vision system not only generated the coordinate (X, Y) of the ball and goal but also heigh and width of the object. All the coordinates and the heigh or width information will be sent to the main controller which is Intel NUC to process the data and translate the data into a signal to servo controller so that the servo motor able to move the robot according to the coordinate given by the vision system.

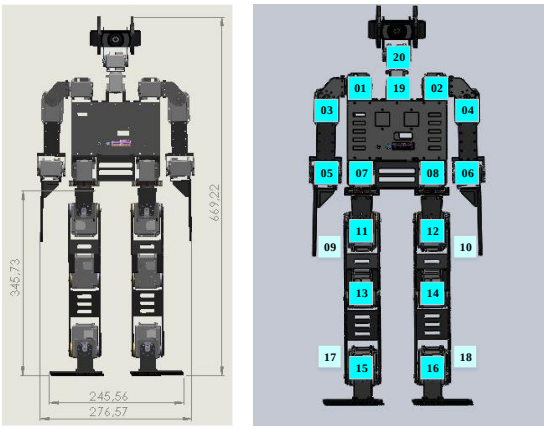

Figure 1: The construction of the Barelang-FC humanoid robot.

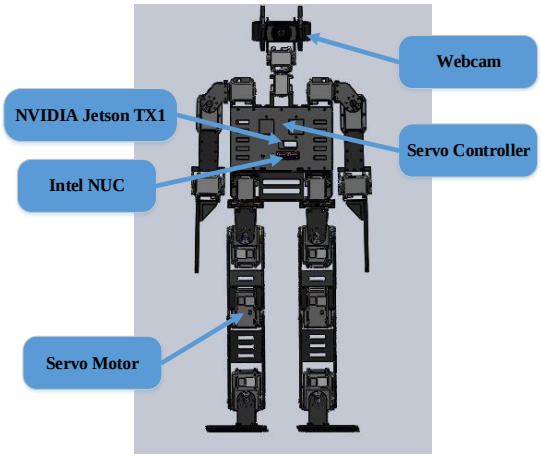

Figure 2: The specification of the Barelang-FC humanoid robot soccer.

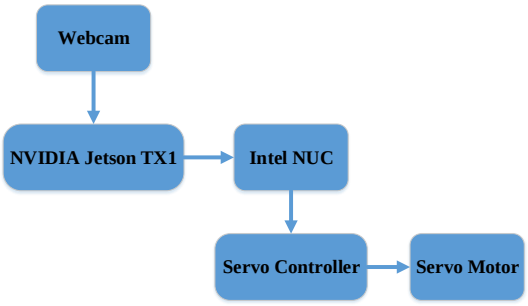

Figure 3: The block diagram system of the BarelangFC.

3. The Self-Localization Prediction

In generating the localization, the MCL which is derived from the Bayes Filter through the Markov Localization and Particle Filter, commonly work with four steps such as: initialization, prediction, update and resample which constructed on **Figure 4**. The MCL algorithm used the random particle or sample from the robot posed where the coordinate generates by MCL is $((x, y, \theta)^T, p)$, the x, y, θ denotes to the robot orientation and position on the field and p represents the weight of the sample.

In order to estimate the robot position on the field, the first process is to set the robot at initial coordinate position. In this position, robot should be able to initiate 100 particles which are spread on the field, where the 90% of the particles centralized from the robot initial position and the 10% of the particles was spread randomly on the field. The illustration of the initialization process can be seen on **Figure 5**. The blue dotted on **Figure 5**, represented as the robot estimation position while the red dotted to the particle which spread on the field.

In addition, when the robot moves on the field, each particle will move to follow the robot's movement as well. Therefore, in order to estimate the robot movement position (x,y), at first, the heading of the robot is needed to be understood by the robot. In previous work [30], we used the vision and apply the trigonometry algorithm to predict the heading of the

robot keeper. In spite of using the vision, in this work, the heading of the robot was generalized by the yaw data from the IMU sensor. And the robot movement estimation position is calculated by using the angle of the servo leg joint odometry system which produced the distance per step of the robot. The fusion of robot movement estimation and the robot heading, generated the predict step which can be seen on **Figure 4**. As for the movement of the particle samples, it which is taken from the odometry data of the robot where the position and orientation are obtained from equation (1).

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} dx \\ dy \\ d\theta \end{pmatrix} + \begin{pmatrix} rndGx \\ rndGy \\ rndG\theta \end{pmatrix} \quad (1)$$

The $(rndGx, rndGy, rndG\theta)^T$, is the error random zero-mean Gaussian value. This value will simulate the error which is happened from the odometry data measurement. The p-value uses the same sampling data when the robot walked on the field without scanning the filed landmark.

As presented on **Figure 4**, after the robot step or motion prediction is done, then the next is to update the step. The update step will be proceeded after the heading and the robot distance per step is added to the particle filter. Because these two data are needed to update the weight of the particle sample to calculate the distance between robot position and the landmark which is detected by the robot through vision system, which is implemented the XNOR-YOLO [32]. The update step in this work, has three main procedures as follows:

1. Transformation, which is done to translate the prediction step ($\boldsymbol{X_{Local}}, \boldsymbol{Y_{Local}}$) become the coordinate ($\boldsymbol{X_{Global}}, \boldsymbol{Y_{Global}}$) on the field by using these two matrix rotation equation.

$$\boldsymbol{X_{Global}} = (\boldsymbol{cos\,\theta} \times \boldsymbol{X_{Local}}) - (\boldsymbol{sin\,\theta} \times \boldsymbol{Y_{Local}}) \quad (2)$$

$$\boldsymbol{Y_{Global}} = (\boldsymbol{sin\,\theta} \times \boldsymbol{X_{Local}}) + (\boldsymbol{cos\,\theta} \times \boldsymbol{Y_{Local}}) \quad (3)$$

2. Association, this procedure is needed to be done because when the robot detected the landmark through the vision system, sometimes the result did not match to the real distance robot navigation. Therefore, the system will make the distance of the closest landmark to the robot for updating the particle weight as the priority.
3. Update weight, is the process of calculating the final particle weight by using the Multivariate-Gaussian probabilistic density. This probabilistic density generates two dimensions coordinate which are x and y. The mean value is obtained by the landmark position and the deviation of the Multivariate-Gaussian. To evaluate this probability distribution, it can be used the equation as follow:

$$\text{Output} = \frac{1}{2\pi\sigma_x\sigma_y}\, e^{-\left(\frac{(x-\mu_x)^2}{2\sigma_x^2}+\frac{(y-\mu_y)^2}{2\sigma_y^2}\right)} \quad (4)$$

Where, **x** and **y** denoted to the global landmark coordinate. The **μx** and **μy** represented the global landmark measurement. And the **σx** and **σy** produced the deviation of the **x** and **y**.

The last process before obtaining the estimate position is the resample process. In this process, all the previous particle will be replaced by the new one with consider to the weight value of each particle. The higher particle remained on the landmark field while the lower one will be erased. The result of the robot estimation position presented on **Figure 5**, where the global and estimation position of the robot depicted on this figure.

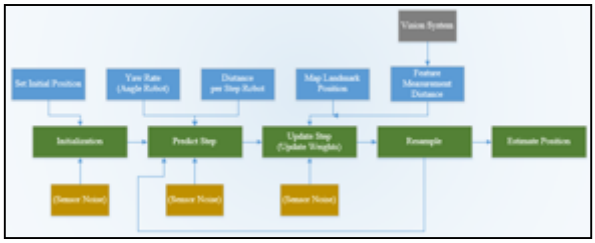

Figure 4: The system architecture of robot localization.

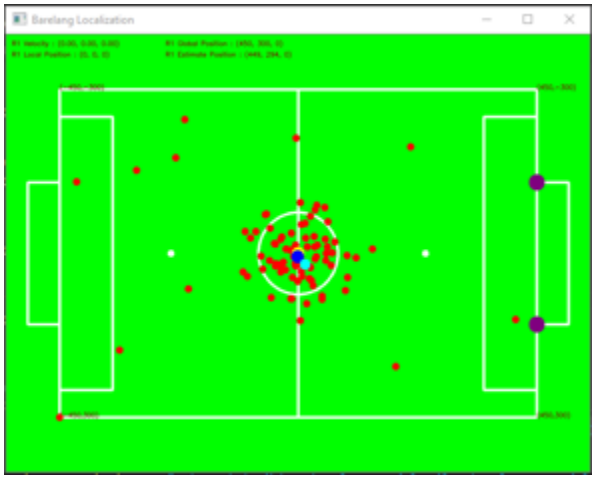

Figure 5: The MCL localization prediction result.

4. Experimental Results

In this experiment, we set the robot on the field based on the Robocop 2019 [33] rules which can be seen on **Figure 6**, while the description of the field described on TABLE I. The field has 9m x 6m for the dimension and each of area will be notated as a landmark for the localization. The notated area list on TABLE I consist of area A to I, where all the area has own dimension and location. The robot will be played in this field to seek and kick the ball towards the goal. To ensure that the robot able to recognize the ball, we put some balls on the field and monitored it through our monitoring system which can be seen on **Figure 7.** The information of the robot monitoring system on **Figure 7**, consists of the last data receive which is used to determine which robot already connected to the monitoring system. Then it also has the data of each

robot and the ball coordinate that send by the robot which already connected to the monitoring system. The monitoring system also present the side position of the robot whether it was on the magenta side or blue side.

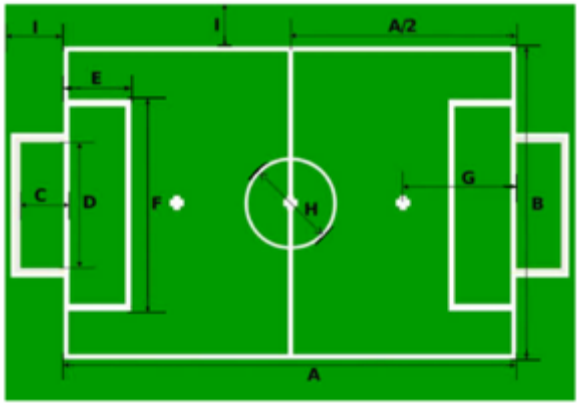

Figure 6: The soccer field area for verifying the proposed method.

TABLE I

THE FOOTBALL FIELD DESCRIPTION

Notation	Description	Dimension (m)
A	Field length	9
B	Field width	6
C	Goal depth	0.6
D	Goal width	2.6
	Goal height	1.8
E	Goal area length	1
F	Goal area width	5
G	Penalty mark distance	2.1
H	Center circle diameter	1.5
I	Border strip width (min.)	0.7

Figure 7: The robot monitoring system.

The first experiment was set to verify the distance estimation based on the odometry system to determine the amount of robot steps for estimating the walking distance of the robot. In order to determine the amount of robot steps, at first, we collected the angle generated by the servo motor when the robot in stood still, walked, and kicked the ball. As the kinematic generated by the LUA, then it is necessary to transmit all the servo data to the main program based on the servo ID mounted to the robot. Each of servo ID from the robot can be seen on **Figure 8.** On **Figure 8**, the left picture presented each servo ID while the other side described the variable data of the servo ID from LUA. Moreover, the LUA kinematic will be command to transmit the angle data of the robot when the robot in stood position, moved forward and backward, also moved to the left and right side. All the collected data presented from **Figure 9** to **Figure 13**. As seen on each figure from **Figure 9** to **Figure 13**, the data which send by the LUA shown in the red box for the left foot and the green box for the other foot consist of each angle data of the servo motors as presented on **Figure 8**. All these angle data will be used to discover the robot foot step, then we will calculate a robot distance estimation displacement. In order to testify the distance estimation, we collected the data on the distance of each speed of the robot's displacement and controlled it by the angular speed of the servo with variable function (x, y, α). From the data collection, it can be concluded that the characteristic of robot movement was, when the higher variable of an angular velocity was generated then the robot step height will be greater and affected the distance traveled by the robot.

The collected data can be seen on TABLE II which denoted that the robot moved backward in negative sign and positive sign for the other movement for the X position. If we made the relation between robot steps and the angular velocity of the robot the graph can be seen on **Figure 14**. While the Y position denoted to the robot moved to the right and left side can be seen on TABLE III. The same as before, in TABLE III also has a negative and positive sign which mean the right-side movement for the positive sign and the negative sign for the other movement. And the graph relation presented on **Figure 15**. From the **Figure 14** and **Figure 15**, each of graph can be generated the linear regression and assumed that the servo angular velocity become x and the actual distance for the y. The equation presented in (4) and (5). The equation (4) was used to get the distance traveled of the robot from the start position by using x function, and the equation (5) for the y function.

$$Fx(y) = 89.233x + 0.3757 \quad (4)$$

$$Fx(y) = 208.93x + 0.0286 \quad (5)$$

As the example when we command the robot to move forward with 20 footsteps, then our system will provide the information as seen on **Figure 16**. On **Figure 16** it has the label such as "Jarak" which mean distance, "Jumlah Maju" denoted to amount of robot steps and "Robot Bergerak" represented as robot

movement. The experiment which presented on **Figure 16**, command the robot to move about 20 steps, then the system will calculate the distance traveled of the robot. When the robot got command for step forward in 20 steps, the system record the distance traveled about 133.44 cm.

After the distance traveled of the robot able to determine, then the position coordinate of the robot can be estimate by using the MCL algorithm. Before adding the MCL algorithm, we need to update the sample pose of the robot by using the motion model. On the MCL algorithm, we implemented the particle filter in order to identify the motion model from the odometry model. The odometry was used to determine the pole position of the robot. The robot position estimation, therefore generated by using the degree heading of the robot, and the X, Y coordinate and calculate the estimation coordinate by using the simple trigonometry function. Moreover, to simplify the implementation of the localization, we assumed some grid on the field which can be seen on **Figure 17** which has 54 grids and each dimension was 100cmx100cm. To testify the robot moved to each grid, as presented on **Figure 18** and **Figure 19**, where **Figure 18** presented the results on the monitoring system when we command the robot to moved 200 to coordinate X and 154 to Y coordinate from the center of the field with the degree movement was 37 degree. The simulation of the robot movement on the gird can be seen on **Figure 19**, where at first the robot stand in the middle of the field denoted with orange dotted and then move forward to coordinate (200, 154) with angle 37 degree. If we can see on **Figure 19**, the robot moved to grid 41 and the heading of the robot refers to the center of the goal which represented by cyan line. To verify the estimation system by using the grid on the field, we commanded the robot to move according to the coordinate that send to robot and compared the estimate coordinate to the actual coordinate distance which presented on TABLE III. From TABLE III, it has deviation between the estimate coordinate and the actual distance coordinate with the deviation around <100 cm for each coordinate X and Y. It is because of the tolerance range of the sensor heading about ± 5 degree and the tolerance to the destination coordinate about ± 10 cm for tolerance the sensor reading data.

In order to testify further performance of the proposed algorithm, the coordinate estimation also verify by using the field landmark. The landmark of the field generated by the vision system to detect the upper and bellow of the goal's pole which represented on **Figure 20**. The upper pole which detected by the object detection represented by yellow square and the lower pole denoted by the blue square on **Figure 20**. When the vision succeeded to detect the goal's pole, then the coordinate of the upper and lower pole can be generated as presented on **Figure 21**. This coordinated pole results will be used as the field landmark as simulated on **Figure 22** by the cyan dotted color and magenta dotted color.

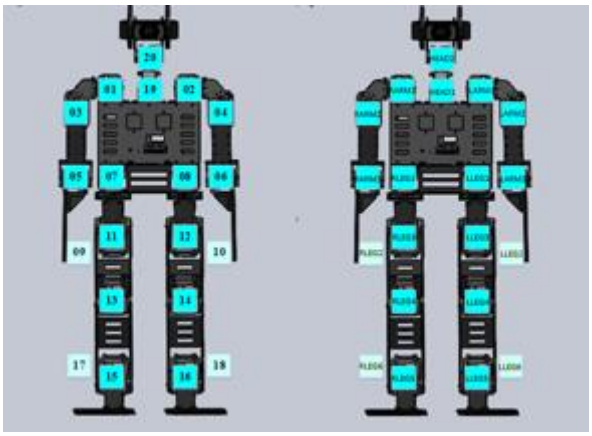

Figure 8: Each of servo ID for receive and transmit servo data to the main program.

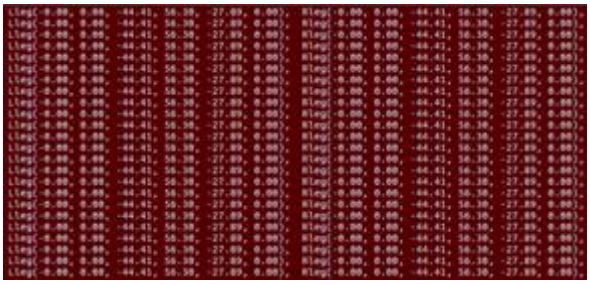

Figure 9: The servo angle data when the robot stood still.

Figure 10: The servo angle data when the robot moved forward.

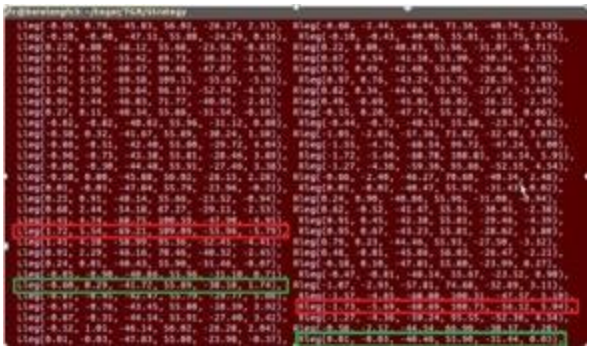

Figure 11: The servo angle data when the robot moved backward.

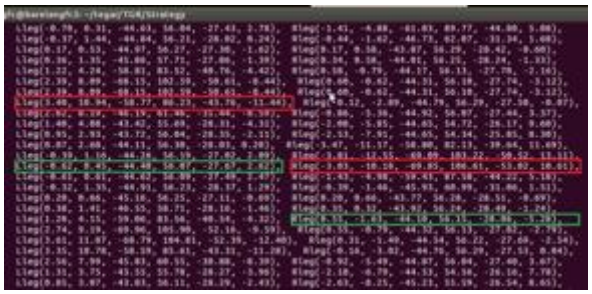

Figure 12: The servo angle data when the robot moved to the right side.

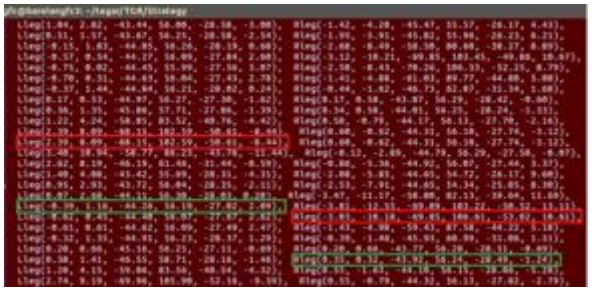

Figure 13: The servo angle data when the robot moved to the left side.

TABLE II

THE DATA COLLECTION WHEN THE ROBOT MOVED FORWARD AND BACKWARD

Variable (x)	Number of Steps	Measured Distance (cm)	Distance/Step (cm)
0.08	30	220	7.33
0.07	30	205	2.63
0.06	30	190	3.36
0.05	30	155	5.17
0.04	30	100	6.33
0.03	30	79	6.83
0	0	0	0
-0.01	10	-8	-0.8
-0.02	10	-12	-1.2
-0.03	10	-18	-1.8

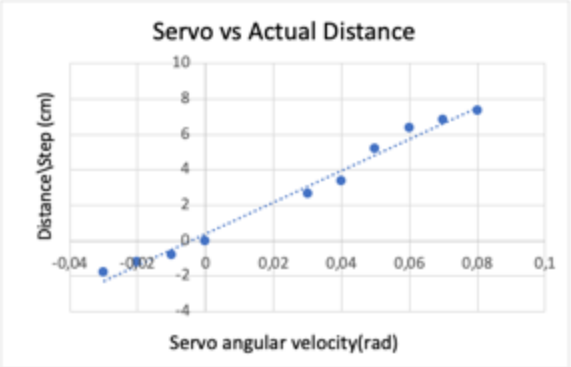

Figure 14: The relation between servo angular velocity and the robot distance per steps for the x coordinate.

TABLE III

THE DATA COLLECTION WHEN THE ROBOT MOVED TO THE RIGHT AND LEFT SIDE

Variable (y)	Number of Steps	Measured Distance (cm)	Distance/Step (cm)
0.03	15	95	6.333333333
0.02	15	63	4.2
0.01	15	30	2
0	15	0	0
-0.01	15	-28.5	-1.9
-0.02	15	-61.5	-4.1
-0.03	15	-95	-6.333333333

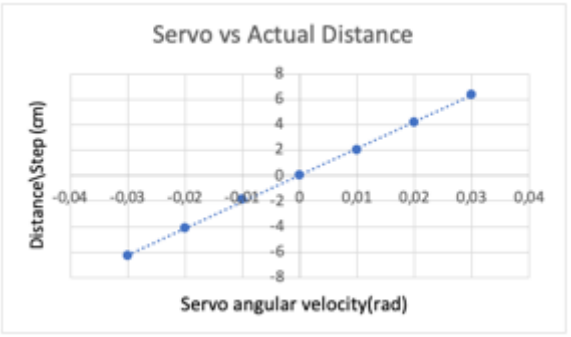

Figure 15: The relation between servo angular velocity and the robot distance per steps for the y coordinate.

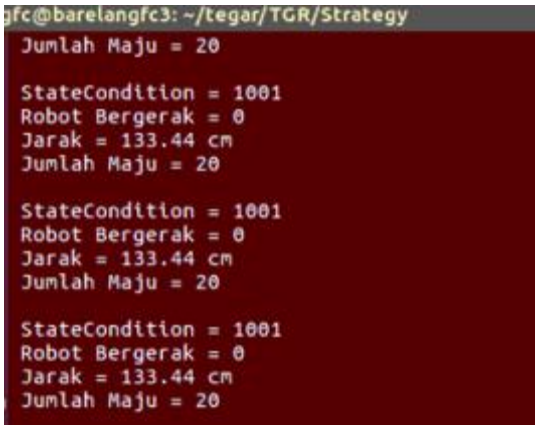

Figure 16: The results of robot displacement based on its footsteps.

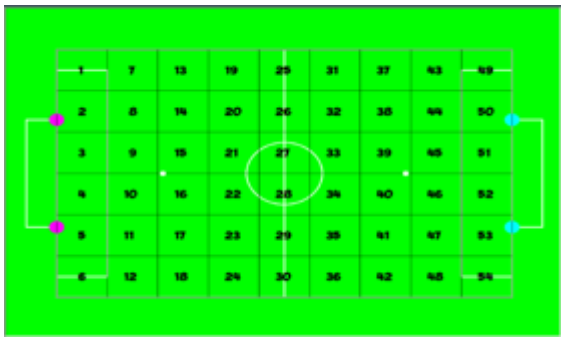

Figure 17: Grid coordinate on the field.

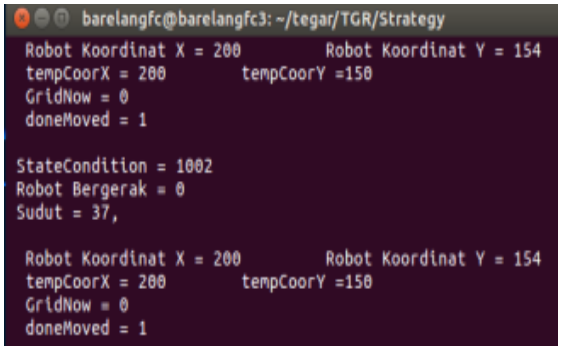

Figure 18: Coordinate estimation of the robot position.

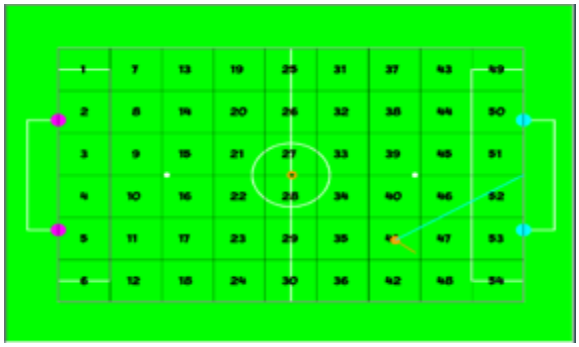

Figure 19: The coordinate estimation for the robot position on the grid area.

TABLE IV

THE COMPARISON BETWEEN ROBOT POSITION ESTIMATION AND ACTUAL POSITION

Gird	Coordinate (X, Y)	Position Estimation (X, Y)	Actual Position (X, Y)
31	100, -250	93, -250	95, -255
36	100,250	91,252	90, 262
38	200, -150	203, -156	208, -152
41	200,150	200, 154	204, 160
44	300, -150	305, -155	312, -160
45	300, -50	308, -49	310, -60
46	300,50	305, 58	320, 60
47	300,150	308, 145	305, 155
50	400, -150	400, -155	405, -160
53	400,150	401, 158	395, 160

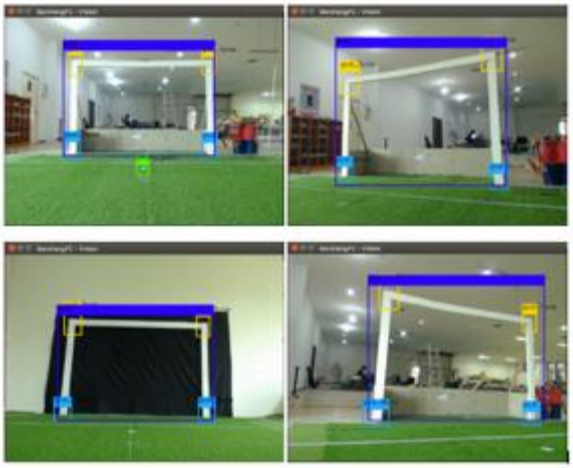

Figure 20: The upper and below goal's pole from the vision system.

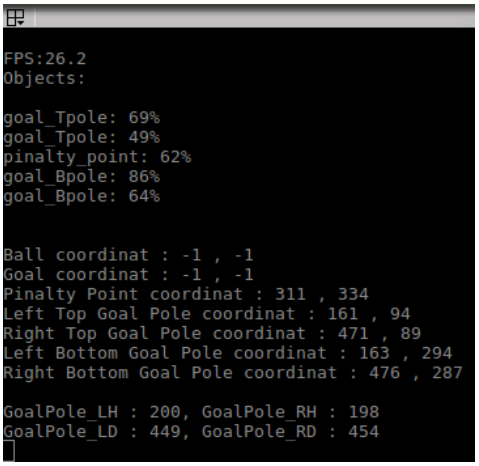

Figure 21: The goal coordinated generated by the vision system.

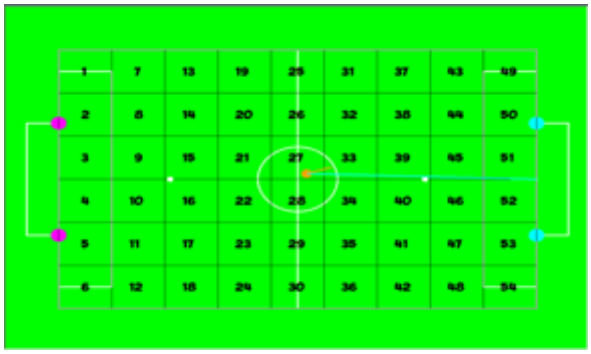

Figure 22: The coordinate estimation of robot position using the landmark from vision system.

5. Conclusions and Future Works

This paper proposed a method which able to estimate the localization of the robot in real-time application. At first, the distance traveled of the robot has been done by using the odometry system. Then, the localization of the robot is done by using the MCL algorithm. On the MCL algorithm, we add the particle filter on the motion model from the odometry to estimate the start position of the robot. And the localization of the robot was construct by using a simple trigonometry function to get the estimation from the heading coordinate of the robot. From the experimental results, this method able to estimate the robot localization in real-time. In the future, the robot balance data when movement in odometry system need to be added to optimize the servo angle when detecting the robot step.

References

[1] O. Sugiyama, R. Kojima and K. Nakadai, "Interactive interface to optimize sound source localization based on microphone array with coarse-to-fine tuning for humanoids," 2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids), Seoul, 2015, pp. 825-830, doi: 10.1109/HUMANOIDS.2015.7363449.

[2] Kumagai et al., "Complementary integration framework for localization and recognition of a humanoid robot based on task-oriented frequency and accuracy requirements," 2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids), Birmingham, 2017, pp. 683-688, doi: 10.1109/HUMANOIDS.2017.8246946.

[3] M. Fourmy, D. Atchuthan, N. Mansard, J. Solà and T. Flayols, "Absolute humanoid localization and mapping based on IMU Lie group and fiducial markers," 2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids), Toronto, ON, Canada, 2019, pp. 237-243, doi: 10.1109/Humanoids43949.2019.9035005.

[4] M. F. Fallón, M. Antone, N. Roy and S. Teller, "Drift-free humanoid state estimation fusing kinematic, inertial and LIDAR sensing," 2014 IEEE-RAS International Conference on Humanoid Robots, Madrid, 2014, pp. 112-119, doi: 10.1109/HUMANOIDS.2014.7041346.

[5] V. Sushrutha Raghavan, D. Kanoulas, C. Zhou, D. G. Caldwell and N. G. Tsagarakis, "A Study on Low-Drift State Estimation for Humanoid Locomotion, Using LiDAR and Kinematic-Inertial Data Fusion," 2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids), Beijing, China, 2018, pp. 1-8, doi: 10.1109/HUMANOIDS.2018.8624953.

[6] T. Zhang, E. Uchiyama and Y. Nakamura, "Dense RGB-D SLAM for Humanoid Robots in the Dynamic Humans Environment," 2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids), Beijing, China, 2018, pp. 270-276, doi: 10.1109/HUMANOIDS.2018.8625019.

[7] R. Sheikh, S. OBwald and M. Bennewitz, "A Combined RGB and Depth Descriptor for SLAM with Humanoids," 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, 2018, pp. 1718-1724, doi: 10.1109/IROS.2018.8593768.

[8] Antoine Rioux, Wael Suleiman, Autonomous SLAM based humanoid navigation in a cluttered environment while transporting a heavy load, Robotics and Autonomous Systems, Volume 99, 2018, Pages 50-62, ISSN 0921-8890, https://doi.org/10.1016/j.robot.2017.10.001. (http://www.sciencedirect.com/science/article/pii/S0921889015303043)

[9] J. Delfin, H. M. Becerra and G. Arechavaleta, "Humanoid localization and navigation using a visual memory," 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids), Cancun, 2016, pp. 725-731, doi: 10.1109/HUMANOIDS.2016.7803354.

[10] Josafat Delfin, Héctor M. Becerra, Gustavo Arechavaleta, Humanoid navigation using a visual memory with obstacle avoidance, Robotics and Autonomous Systems, Volume 109, 2018, Pages 109-124, ISSN 0921-8890, https://doi.org/10.1016/j.robot.2018.08.010.

[11] H. Minakata et al., "A method of single camera robocup humanoid robot localization using cooperation with walking control," 2008 10th IEEE International Workshop on Advanced Motion Control, Trento, 2008, pp. 50-55, doi: 10.1109/AMC.2008.4516040.

[12] M. N. Sudin, M. F. Nasrudin and S. N. H. S. Abdullah, "Humanoid localisation in a robot soccer competition using a single camera," 2014 IEEE 10th International Colloquium on Signal

Processing and its Applications, Kuala Lumpur, 2014, pp. 77-81, doi: 10.1109/CSPA.2014.6805724.

[13] L. Guohua, X. Xiandong, Y. Xiang, W. Yadong and Q. Tianwei, "An Indoor Localization Method for Humanoid Robot Based on Artificial Landmark," 2015 Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC), Qinhuangdao, 2015, pp. 1854-1857, doi: 10.1109/IMCCC.2015.394.

[14] Mahdi Fakoor, Amirreza Kosari, Mohsen Jafarzadeh, Humanoid robot path planning with fuzzy Markov decision processes, Journal of Applied Research and Technology, Volume 14, Issue 5, 2016, Pages 300-310, ISSN 1665-6423, https://doi.org/10.1016/j.jart.2016.06.006.(http://www.sciencedirect.com/science/article/pii/S1665642316300700)

[15] X. Xu, B. Hong and Y. Guan, "Humanoid robot localization based on hybrid map," 2017 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC), Shenzhen, 2017, pp. 509-514, doi: 10.1109/SPAC.2017.8304331.

[16] B. Tian, Chuen-Leong Ng and C. Chew, "Self-localization of humanoid robots with fish-eye lens in a soccer field," 2010 IEEE Conference on Robotics, Automation and Mechatronics, Singapore, 2010, pp. 522-527, doi: 10.1109/RAMECH.2010.5513138.

[17] H. Li et al., "A humanoid robot localization method for biped navigation in human-living environments," 2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), Shenyang, 2015, pp. 540-544, doi: 10.1109/CYBER.2015.7287997.

[18] D. Maier, A. Hornung and M. Bennewitz, "Real-time navigation in 3D environments based on depth camera data," 2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012), Osaka, 2012, pp. 692-697, doi: 10.1109/HUMANOIDS.2012.6651595.

[19] Y. Omori, T. Furukawa, T. Ishikawa and M. Inaba, "Humanoid Vision Design for Object Detection, Localization and Mapping in Indoor Environments," 2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), Philadelphia, PA, 2018, pp. 1-6, doi: 10.1109/SSRR.2018.8468604.

[20] G. Oriolo, A. Paolillo, L. Rosa and M. Vendittelli, "Vision-based Odometric Localization for humanoids using a kinematic EKF," 2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012), Osaka, 2012, pp. 153-158, doi: 10.1109/HUMANOIDS.2012.6651513.

[21] G. Oriolo, A. Paolillo, L. Rosa and M. Vendittelli, "Vision-based trajectory control for humanoid navigation," 2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids), Atlanta, GA, 2013, pp. 118-123, doi: 10.1109/HUMANOIDS.2013.7029965.

[22] Y. Minami Shiguematsu, M. Brandao, K. Hashimoto and A. Takanishi, "Effects of Biped Humanoid Robot Walking Gaits on Sparse Visual Odometry Algorithms," 2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids), Beijing, China, 2018, pp. 160-165, doi: 10.1109/HUMANOIDS.2018.8625015.

[23] R. Carrillo Mendoza, P. Vera Bustamante, Brian, E. Hernández Castillo and J. M. Ibarra Zannatha, "3D self-localization for humanoid robots using view regression and odometry," 2015 12th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), Mexico City, 2015, pp. 1-5, doi: 10.1109/ICEEE.2015.7357988.

[24] Saeed Saeedvand, Hadi S. Aghdasi, Jacky Baltes, Novel lightweight odometric learning method for humanoid robot localization, Mechatronics, Volume 55, 2018, Pages 38-53, ISSN 0957-4158, https://doi.org/10.1016/j.mechatronics.2018.08.007.(http://www.sciencedirect.com/science/article/pii/S0957415818301338)

[25] Wei Hong, C. Zhou and Y. Tian, "Robust Monte Carlo Localization for humanoid soccer robot," *2009 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Singapore, 2009, pp. 934-939, doi: 10.1109/AIM.2009.5229889.

[26] I. Nagi, W. Adiprawita and K. Mutijarsa, "Vision-based Monte Carlo localization for RoboCup Humanoid Kid-Size League," 2014 13th International Conference on Control Automation Robotics & Vision (ICARCV), Singapore, 2014, pp. 1433-1438, doi: 10.1109/ICARCV.2014.7064526.

[27] A. Muzio, L. Aguiar, M. Maximo and S. 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, 2016 pp. 334-339. doi: 10.1109/LARS-SBR.2016.63https://doi.ieeecomputersociety.org/10.1109/LARS-SBR.2016.63

[28] A. C. Almeida, A. H. R. Costa and R. A. C. Bianchi, "Vision-based monte-carlo localization for humanoid soccer robots," 2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR), Curitiba, 2017, pp. 1-6, doi: 10.1109/SBR-LARS-

R.2017.8215310.do

[29] Hartfill, Judith. (2019). Feature-Based Monte Carlo Localization in the RoboCup Humanoid Soccer League. 10.13140/RG.2.2.19044.73602.

[30] Susanto, F. Azmi and R. Analia, "Trigonometry Algorithm for Ball Heading Prediction of Barelang-FC Goal Keeper," 2018 International Conference on Applied Engineering (ICAE), Batam, 2018, pp. 1-6, doi: 10.1109/INCAE.2018.8579361.

[31] Susanto, Febri Alwan Putra and R. Analia, "XNOR-YOLO: The High Precision of the Ball and Goal Detecting on the Barelang-FC Robot Soccer," 2020 International Conference on Applied Engineering (ICAE), Batam, 2020 (virtual presentation on 7th -8th October 2020)

[32] RoboCup Humanoid Technical Committee. Laws of the Game 2019. http://www.robocuphumanoid.org/wp-content/uploads/ RCHL-2019-Rules-final.pdf, 2019.