http://dx.doi.org/10.5755/j02.eie.30736 ELEKTRONIKA IR ELEKTROTECHNIKA, ISSN 1392-1215, VOL. 28, NO. 4, 2022

Reinforcement Optimization Algorithm for Mobile Robot Sensor Networks Drive Motion Improvement

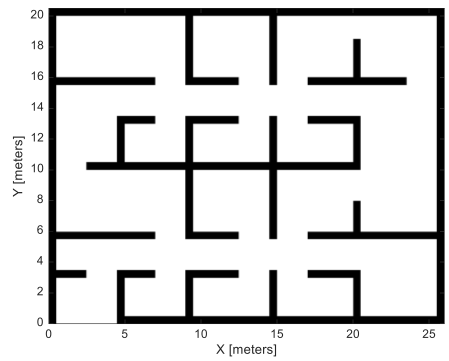
Suryaprakash Shanmugasundaram*, Thirumoorthi Ponnusamy, Tamilarasu Viswanathan
Department of Electrical and Electronics Engineering, Kumaraguru College of Technology, Coimbatore, 641049, India
suryaprakash.vsm@gmail.com

***Abstract*—This paper proposed four optimization algorithms for mobile robot sensor networks that improve the kinematics drive motion in a reference map environment. The standard procedure followed in mobile robot sensor measurements considers a problem statement for relating the sensor measurements with a reference map. The initial path shows that the existing methods lack consideration of more sensor points without considering the boundary constraints and obstacles. The probabilistic path map can be rearranged according to the current location to improve the better drive motion, as well as to obey the fundamental kinematics equations. he obstacle crossing led to the development of new algorithms. Implementation of schemes is achieved in different map environments, and the accuracy of results outperforms conventional methods by 84.21 % to 96.94 %.**

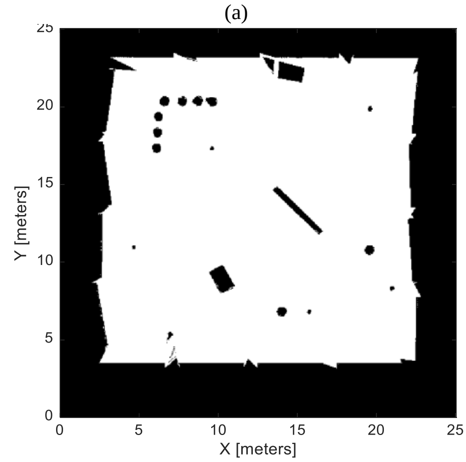
***Index Terms*—Mobile robots; Sensor network; Shortest path; Optimization algorithm.**

I. INTRODUCTION

The estimation of the path of mobile robot sensor networks is being analyzed for a wide range of applications, considering the wide range of parameters [1]–[3]. These parameters are greatly capable with the working environment and the boundaries with increasing size of obstacles. After the introduction of the autonomous concept with technological advancement in learning of new algorithms [4], [5], the measurement of the sensor parameter becomes simple. Application-based pathfinding and localization become complicated after handling a large amount of data in real time with variation in other sensor networks. The introduction of kinematics drive motion improves the evaluation of the path in random situations, which a mobile robot has never experienced [6]–[8]. The constructional features of the robot should match the environment, which reduce the constraint in the system capable of improving the sensor data measurements [9], [10]. Navigation architecture has gained attention in recent development of localization algorithms of existing sensor data units; however, path and obstacle clustering must be improved. The collection of more sensor measurement units reduces the computational error, but takes time to identify the localization regardless of the map environment [11]–[13]. Periodical matching of different sensor data is required to learn and improve the data [14], [15]. These uncertainties improved for path planning from existing literatures by the introduction of four different reinforcement algorithms are discussed in this work.



(a)



(b)

Fig. 1. Reference maps of mobile robot sensor measurements: reference occupancy maps (a) and (b).

These algorithms are implemented in recent sensor networks, such as multiple energy harvesting sensors [15], Delaunay triangulation [16], Diophantine fuzzy graph [17], Objects with Complex Geometry [18], Partially Known Environments [19], Q-Learning [20], and Recursive fusion estimation [21]. As a result, the proposed algorithms adapt to these sensor networks with effective improvement. The organisation of this work starts with the background study and the identification of the research gap. In Section II, the

Manuscript received 17 February, 2022; accepted 11 May, 2022.

problem statement is detailed with a numerical model, followed by the algorithms proposed in Section III. The results and validation are discussed in Section IV. The limitations and future possibilities of this work are elaborated in Section V.

II. Problem Statement of Mobile Robot Sensor Measurements

The optimization algorithm is developed for a specific mobile robot identified from the sensor measurement values for the specific operating map environment. As a primary progress, the operating map environment should be developed for simulating the sensor values from the basic kinematics drive motion where the robot is operating with the specific velocity. Once the map is developed, the sensor reading decides the boundaries and validates for every measurement unit. The final path developed for the moving the object in the specified map is identified from the binary tenancy grid values. Many literatures proposed a new kind of map environment to control the mobile robot. The objective of this work is to develop an optimization algorithm for mobile robot sensor networks developing in a simple map without many constraints. Reference occupancy maps for performing the initial standardization for the mobile robot are developed as shown in Fig. 1.

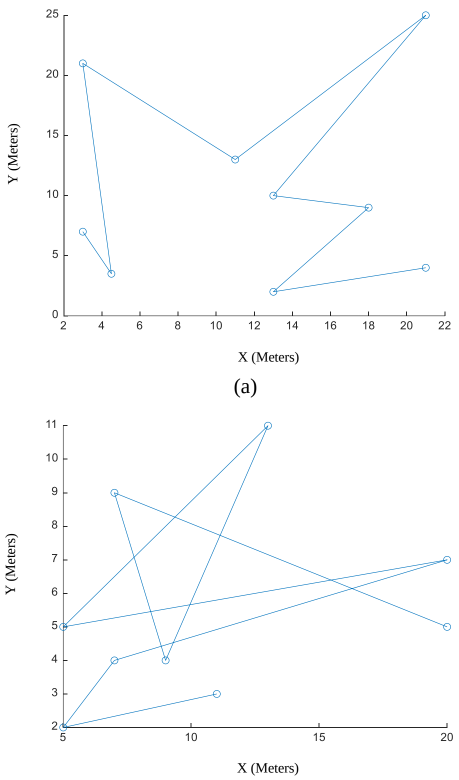


Fig. 2. Reference maps of the mobile robot with initial path: reference occupancy maps (a) and (b).

The mobile robot kinematic moving model develops for specific input and the moving rate by considering the expected moving velocity and maximum operating velocity in fundamental moving controller. The sensor operating ranges for the specified mobile robot decided by the controller and the operational boundary decided by the unknown map are considered random values as mentioned in (1)

$$P(x_i, y_i) = \begin{bmatrix} rand(x_1, y_1) & rand(x_1, y_2) & \cdots & rand(x_1, y_b) \\ rand(x_2, y_1) & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ rand(x_b, y_1) & rand(x_b, y_2) & & rand(x_b, y_b) \end{bmatrix}. \quad (1)$$

Path $P(x_i, y_i)$ opted for the mobile robot within the bounded map (x_b, y_b) values are chosen randomly from the initial sensor values. The result of the connecting points form a path to decide the preliminary control values to operate the sensor redefines (1). The implementation of the path inside the map agreed from the state of mobile robot with starting and ending values should match with the path. The navigation of initial starting values fixing the sensor values matches with boundary values consider for the unidentified map and forms a path as shown in Fig. 2(a) and 2(b). These map environments are considered a common testing case for any mobile robot. The basic control of mobile robot identified for unknown map from the sensor values is illustrated in Fig. 3.

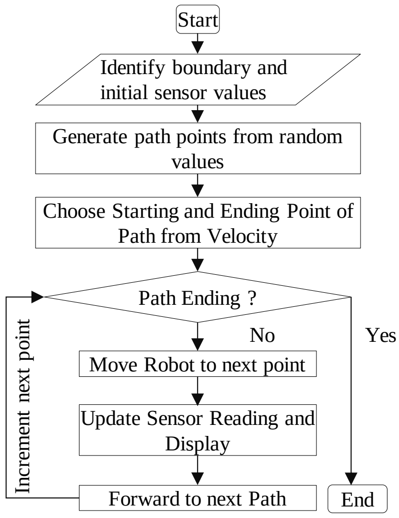


Fig. 3. Flowchart of measurement of the mobile robot sensor for a given map.

The input of a mobile robot moving given for the specified map for collecting the sensor measurements initiates with control period of the specific path decided from the moving velocity and maximum operating velocity. Considering the initial position to the end position of each path, the mobile robot simulated in the specified time control the mobile robot wheels for every map feature as shown in the flowchart.

The probabilistic path map should be created to perform the free robot moving space with adjustable velocity speed among locations decided from (1). Path calculation from these spots to avoid the boundaries that are present due to the robot's size is dynamically changing. The robot operating circumference from the obstructions is presented on the probabilistic path map. Reference from the binary tenancy from the random nodes is decided from the simulation times from the standard properties path distance and update values. The feasible route for probabilistic path is updated for every point created for more complicated boundaries to ensure the existing flowchart noted the sensor measurement.

III. PROPOSED REINFORCEMENT LEARNING OPTIMIZATION FOR MOBILE ROBOT SENSOR NETWORKS

Introduction of obstacles in the mobile robot path from sensor readings has an impact on linear movement and control movement. Selection of reference from the literatures [11]–[13] bounded values to create a probabilistic path map from the proposed proximal policy optimization, trust region policy optimization, policy gradient reinforcement learning optimization, and deep "Q" network reinforcement learning optimization.

Algorithm 1 introduces an improved path for the identified trajectory from every sampling path to do the proper repetition. It identifies the path the overall estimated computed from (1) and expressed as

$$P^{'}(x_{io}, y_{io}) = P\left[\frac{(x_b, y_b)}{(x_o, y_o)} x^{'}_{o}\right]. \quad (2)$$

The inclusion of obstacles of $x^{'}_{o}$ within the boundary decided from the (x_o, y_o) of the path changed into a new variation in the path $P^{'}(x_o, y_o)$. Sampling of $P(x_b, y_b) / P(x_o, y_o)$ in every point to identify the loss functions makes impact on square error for the better distributions from the proximal policy optimization.

Algorithm 1. Proposed proximal policy optimization for mobile robot sensor networks.

1. Initialize the starting point of unknown map (x_1, y_1)
2. Randomize the values of map within boundary
3. Initialize path $P(x_i, y_i)$
4. Identify the navigation randomize path points from (1)
5. for every point do
6. for next point (1, 1) ... (b, b) do
7. Sample time "t" for particular "i"
8. if t ∈ b then
9. for every limits 1, 2, ..., t do
10. Increment for a particular position
11. Generate $P(x_b, y_b)$
12. Move limit for next random position for i + 1
13. end for
14. else if t ∈ O then
15. for every limit 1, 2, ..., O do
16. Increment for a next obstacles position
17. Identify the limit $P(x_b/x_o)$
18. Compute the $P(x_b/x_o)$ from new x'_o
19. Varied for *x* to *y* position
20. end for
21. else if t ∈ b then
22. for every limit 1, 2, ..., b do
23. Increment for a next obstacles position
24. Identify the limit $P(y_b/y_o)$
25. Compute the $P(y_b/y_o)$ from new y'_o
26. end for
27. end if
28. Compute obstacles from new random position
29. Varied for *x* to b position
30. end for
31. else if t ∈ b then
32. end for
33. Optimize $P^{'}(x_b, y_b)$
34. end for
35. Path 1 to next path
36. $P_1^{'}(x_b, y_b)$
37. $P_1^{'}(x_b, y_b)$ to $P_2^{'}(x_b, y_b)$
38. Complete rollout for all the path $P_b^{'}(x_o, y_o)$
39. Update to next point for entire path from (2)
40. end for

Algorithm 2 selects a new optimization for choosing the path from the proposed trust region policy optimization for mobile robot sensor networks. It is identified from general modelling for every obstacle as

$$\min_{b} P^{'}(x_o, y_o) = \sum_{t \in O} \left\| x^{'}_{io}(x_{io}, y_{io}) - f(x_{ib}, y_{ib}) \right\|_{x}^{y}. \quad (3)$$

Minimization of the operating mobile robot path is determined from the multivariant parameters such as path position from the random position from the initiating obstacles *rand*(x_o, y_o). The new sensor measurement is computed for the preliminary values and incorporated into the new set of samples presented in the actual path. Trusted region is processed inside the boundary in a particular function $f(x_{ib}, y_{ib})$ to update sensor measurements.

Algorithm 2. Proposed trust region policy optimization for mobile robot sensor networks.

1. Initialize a boundary randomize position *rand*(x_1,y_1)
2. Initialize the points of obstacles *rand*(x_o, y_o)
3. Repeat for new sensor measurements
4. Collect samples from the actual map from the path $P(x_b, y_b)$
5. Train the robot for every boundary "b" using $f(x_{ib}, y_{ib})$.
6. Repeat for new sensor measurements
7. Collect fabricated map samples from boundaries *"b"* and obstacles *"O"*
8. Update the minimize path using (3)
9. Estimate the new path with $P'(x_o, y_o)$
10. until the path reaches its end
11. until the map difference from $x^{'}_{io}(x_o, y_o)$.

The new fabricated map for the boundaries and obstacles learned from the data to minimize the path for all distributed from the gradient. The new estimation multiplied for the other path in (3) finds the map difference $x'_{io}(x_o, y_o)$.

The two-dimensional path of a given map is identified from the policy gradient reinforcement learning of the policy gradient adapted from the obstacles and boundaries with a new group of solutions from the proper selection and developed as in existing literatures. Algorithm 3 initialization starts with the mapping of obstacles considered at various random points for every fitness value available at random points. Compute the group of shortest paths to choose the velocity varied dynamically to choose the exponential values from the combination *(O-b)* selecting the tournament assortment. Reflect the computation to do the crossover for every sample within the random samples for the mutate obstacles. Algorithm 3 identifies the new path P' from the computation of the new sample and updates with small samples computed with the tested policy gradient. Finally, the target sensor groups with the next production path and duplicates within the boundary values for all the small paths.

Map environments are considered as variable obstacles and paths for every set to minimise the two-dimensional end points observed for every specified "Q" value. The initial global observation is chosen from the individual points for every trained value of learning for each map as specified in the Algorithm 4.

The iteration of each path starts from the obstacles present in the map boundary, contemplating all the path possibilities to check the possibilities that occur in the tournament for specified "Q" values. Random values ensure for the next state $i + 1$ of the bounded value computed from the discontinued from the previous state of (1) to modernize the value $P'(x_i, y_i, O, b)$ to choose the path of computation that matches the boundary inside the obstacles.

Algorithm 3. Proposed policy gradient reinforcement learning for mobile robot sensor networks.

1. Initialize the mapping position and boundaries
2. Initialize the position (x_0, y_0)
3. Initialize obstacles inside the mapping for the possible paths
4. Define number of obstacles in each path and random number points r ∈ (O, b)
5. for $x = i$, b do
6. for obstacles P ∈ (O, b) do
7. fitness, P = Evaluate(x_i, y_i, O, b)
8. end for
9. Rank the path based on fitness velocity
10. Select the first grade path $P \in (O, b)$ as chosen where exponential growth
11. Select (O-b) combination from P' to form Set "t" using tournament assortment with another path
12. while |t| < (O-b) do
13. Use crossover between a randomly sampled $t \in P$ and $O \in P'$ and attach to P
14. end while
15. for every sample $O \in P'$ Set b do
16. if random points < (b, O) then
17. Mutate O
18. end if
19. end for
20. P' = Evaluate($x_{io}, y_{io}, b = O, t = i$)
21. Sample a random group of paths of "t" sample
22. Compute $P'(x_{oi}, y_{oi}) = f(x_i + y_i) + P(x_{ob}, y_{ob})$
23. Update P' by lessening the loss
24. Inform (x_{oi}, y_{oi}) utilizing the tested policy gradient
25. Soft fill in target sensor groups
26. if production mod $i + 1 = 0$ then
27. Duplicate the $P'(x_{oi}, y_{oi})$ value into the boundary value
28. end if
29. end for

Algorithm 4. Proposed deep "Q" network reinforcement learning optimization for mobile robot sensor networks.

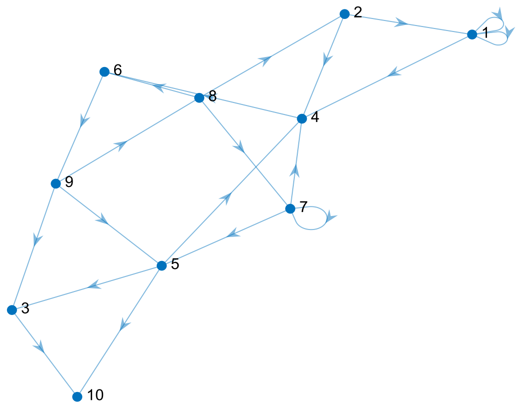
1. initialize map with boundary values
2. initialize random position inside the map limits
3. initialize obstacles from the possible paths
4. for i = 1 to starting points do
5. sample actions $P \in (O, b)$ with closed path
6. accumulate possibilities $P(x_i, y_i, O, b)$ in bounded values
7. gather tournament assortment in "t" from "Q" values
8. store "I" in (b, O)
9. implement (x_{0b}, y_{0b}) and change to next state $i + 1$
10. roll up recompense random values from bounded limit
11. if part is finished then
12. compute discounted rewards (x_{0t}, y_{0t}) from b
13. estimate gradients from previous state of (1)
14. modernize $P'(x_i, y_i, O, b)$ with the projected gradient
15. empty (O, b)
16. end if
17. if existing path ends then
18. reset path
19. end if
20. place O = b
21. end for

IV. RESULTS AND DISCUSSION

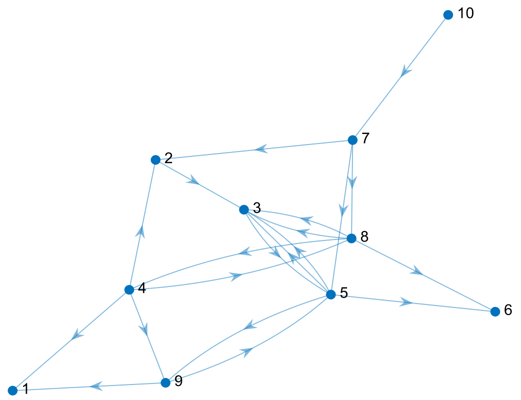
The implementation of the proposed optimization for mobile robot sensor networks completed in MATLAB/Simulink environment for two different maps is discussed in Section II. The complex map and ternary map environment is considered as the fundamental environment to achieve optimal operation.

Figure 4 shows the optimized path identified by mobile robot sensor networks under 10 obstacles on a complex map using proposed methods.

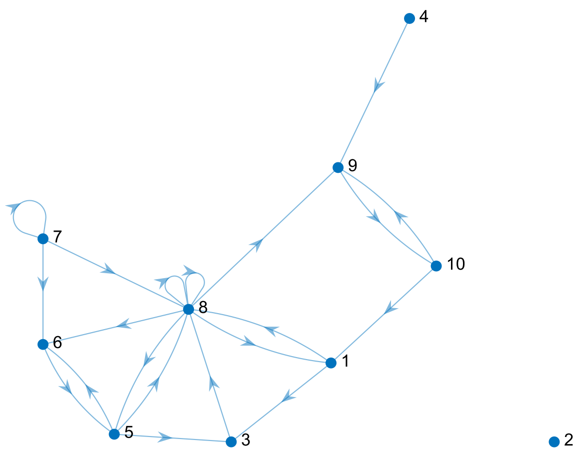
Tables I and II show the comparison of the improvement of accuracy in mobile robot sensor networks for different obstacles applying the proposed algorithm with the conventional method [2]–[5]. As the level increases, both complex map environments exhibit higher accuracy of 96.94 % and 86.75 % for deep "Q" network and trust region policy method.



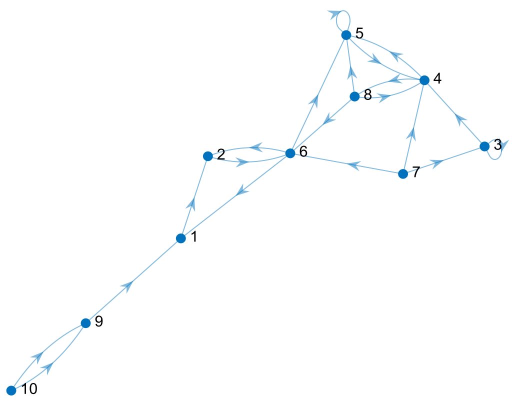
(a)



(b)



(c)



(d)

Fig. 4. Optimized path identified by mobile robot sensor networks under 10 obstacles in complex map using (a) proposed proximal policy optimization, (b) proposed trust region policy optimization, (c) proposed policy gradient reinforcement learning optimization, and (d) proposed deep “Q” network reinforcement learning optimization.

TABLE I. COMPARISON OF ACCURACY IMPROVEMENT IN MOBILE ROBOT SENSOR NETWORKS FOR DIFFERENT OBSTACLES APPLYING PROPOSED ALGORITHM WITH CONVENTIONAL METHOD [2]–[5] UNDER COMPLEX MAP ENVIRONMENT.

Map Level	**Proximal policy**			**Trust region policy**		
	5	**10**	**15**	**5**	**10**	**15**
1	84.97	92.02	94.77	83.19	86.62	88.2
2	85.12	88.37	87.29	96.21	91.49	87.43
3	88.47	83.58	88.89	86.98	96.3	86.5
4	86.09	85.48	84.5	92.55	87.55	91.28
5	92.82	92.59	86.75	89.06	86.76	95.36
Map Level	**Policy gradient**			**Deep “Q” network**		
	5	**10**	**15**	**5**	**10**	**15**
1	86.2	87.67	96.53	90.16	96.34	83.64
2	94.81	90.95	92.34	86.84	93.63	90.94
3	93.69	91.12	89.4	93.05	89.14	88.84
4	89.86	87.02	95.99	94.52	83.4	86.5
5	93.31	87.41	89.84	93.35	96.94	94.55

TABLE II. COMPARISON OF ACCURACY IMPROVEMENT IN MOBILE ROBOT SENSOR NETWORKS FOR DIFFERENT OBSTACLES APPLYING PROPOSED ALGORITHM WITH CONVENTIONAL METHOD [2]–[5] UNDER TERNARY MAP ENVIRONMENT.

Map Level	**Proximal policy**			**Trust region policy**		
	5	**10**	**15**	**5**	**10**	**15**
1	83.07	91.04	92.38	89.36	85.89	85.28
2	90.7	93.54	89.1	95.92	91.8	93.63
3	87.6	85.86	94.66	95.24	83.95	83.95
4	91.85	83.07	90.9	92.77	88.28	85.15
5	87.99	94.42	94.91	84.89	85.52	91.81
Map Level	**Policy gradient**			**Deep “Q” network**		
	5	**10**	**15**	**5**	**10**	**15**
1	85.61	86.58	86.78	93.76	89.78	90.98
2	91.11	90.86	85.34	84.21	86.94	95.31
3	83.6	90.61	95.19	93	87.9	91.49
4	94.51	89.36	93.61	89.67	83.8	88.04
5	84.08	84.25	93.96	91.22	92.27	84.41

Similarly, the ternary map environment was implemented with 95.92 % of accuracy for trust region policy and 84.21 % for policy gradient optimization. In [21], [22], the adaptability of sensor algorithms to any type of network is idealized to be 0 %. Table III presents the algorithm proposed implemented with various network types [16]–[22] regardless of the map environment. The adaptability of the proposed algorithms varies from -10 % to 10 %, which shows their practical feasibility.

TABLE III. IMPLEMENTING THE PROPOSED ALGORITHM WITH VARIOUS NETWORK TYPES [16]–[22] REGARDLESS OF THE MAP’S.

Network Type	**Proximal policy**	**Trust region policy**	**Policy gradient**	**Deep “Q” network**	**Adapt Method**
Multiple sensors [15]	-8.77	-10.63	3.15	-3.29	Policy gradient
Delaunay triangulation [16]	9.68	4.94	-4.9	-4.06	Policy gradient
Diophantine fuzzy graph [17]	8.06	4.08	9.07	7.17	Proximal policy
Objects with Complex Geometry [18]	9.4	7.72	-1.26	-5.71	Trust region policy
Partially Known Environments [19]	8.46	6.33	6.23	2.36	Proximal policy
Q-Learning [20]	1.55	-4.81	10.76	6.89	Deep “Q” network
Recursive fusion estimation [21]	-10.12	-10.52	9.88	-6.8	Proximal policy

V. CONCLUSIONS

The four-reinforcement learning method is proposed for the mobile robot kinematic moving model from the moving velocity rate inside the sensor operating regions from the boundary limit. The standardized path planning method considers as a reference from literatures [11]–[13], which are clearly considered to be connecting points from sensor values. Probabilistic path map is created for the random nodes from the starting and end path of the boundary considering the obstacles. Randomization path for each obstacle identified from the proximal policy optimization, sample of minimize path among the small nodes computed from trust region policy optimization, policy gradient method improved the cross over chances of small paths from the chosen velocity, and the recompensing of random values of deep “Q” method shows individual uniqueness for learning. The accuracy improved from 96.94 % to 84.21 % than the conventional method proposed in the literature. In the future, the advantages of proposed optimization schemes can be implemented for other different environment maps considering a greater number of sensor data.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

[1] J. G. Martin, J. M. Maestre, and E. F. Camacho, "Spatial irradiance estimation in a thermosolar power plant by a mobile robot sensor network", *Solar Energy*, vol. 220, pp. 735–744, 2021. DOI: 10.1016/j.solener.2021.03.038.

[2] Y. Lu, B. Shen, and Y. Shen, "Recursive filtering for mobile robot localization under an energy harvesting sensor", *Asian Journal of Control*, 2021. DOI: 10.1002/asjc.2672.

[3] M. V. Sreenivas Rao and M. Shivakumar, "IR based auto-recharging system for autonomous mobile robot", *Journal of Robotics and Control (JRC)*, vol. 2, no. 4, pp. 244–251, 2021. DOI: 10.18196/jrc.2486.

[4] Y.-C. Liu, T.-C. Lin, and M.-T. Lin, "Indirect/Direct learning coverage control for wireless sensor and mobile robot networks", *IEEE Transactions on Control Systems Technology*, vol. 30, no. 1, pp. 202–217, 2021. DOI: 10.1109/TCST.2021.3061513.

[5] W. Rahmaniar and A. Wicaksono, "Design and implementation of a mobile robot for carbon monoxide monitoring", *Journal of Robotics and Control (JRC)*, vol. 2, no. 1, pp. 1–6, 2021. DOI: 10.18196/jrc.2143.

[6] A. M. Abdulazeez and F. S. Faizi, "Vision-based mobile robot controllers: A scientific review", *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 6, pp. 1563–1580, 2021. DOI: 10.17762/turcomat.v12i6.2695.

[7] X. He, E. Hashemi, and K. H. Johansson, "Navigating a mobile robot using switching distributed sensor networks", 2021. arXiv: 2106.13529.

[8] J. Li, J. Wang, S. Wang, and C. Yang, "Human–robot skill transmission for mobile robot via learning by demonstration", *Neural Computing and Applications*, pp. 1–11, 2021. DOI: 10.1007/s00521-021-06449-x.

[9] T. Q. Tran, A. Becker, and D. Grzechca, "Environment mapping using sensor fusion of 2D laser scanner and 3D ultrasonic sensor for a real mobile robot", *Sensors*, vol. 21, no. 9, p. 3184, 2021. DOI: 10.3390/s21093184.

[10] V. L. Popov, N. G. Shakev, A. V. Topalov, and S. A. Ahmed, "Detection and following of moving target by an indoor mobile robot using multi-sensor information", *IFAC-PapersOnLine*, vol. 54, no. 13, pp. 357–362, 2021. DOI: 10.1016/j.ifacol.2021.10.473.

[11] A. J. Barreto-Cubero, A. Gómez-Espinosa, J. A. Escobedo Cabello, E. Cuan-Urquizo, and S. R. Cruz-Ramírez, "Sensor data fusion for a mobile robot using neural networks", *Sensors*, vol. 22, no. 1, p. 305, 2021. DOI: 10.3390/s22010305.

[12] Y. Lu and H. R. Karimi, "Recursive fusion estimation for mobile robot localization under multiple energy harvesting sensors", *IET Control Theory & Applications*, vol. 16, no. 1, pp. 20–30, 2021. DOI: 10.1049/cth2.12201.

[13] A. Filotheou, "Correspondenceless scan-to-map-scan matching of homoriented 2D scans for mobile robot localisation", *Robotics and Autonomous Systems*, vol. 149, art. 103957, 2022. DOI: 10.1016/j.robot.2021.103957.

[14] S. K. Pattnaik, D. Mishra, and S. Panda, "A comparative study of meta-heuristics for local path planning of a mobile robot", *Engineering Optimization*, vol. 54, no. 1, pp. 134–152, 2022. DOI: 10.1080/0305215X.2020.1858074.

[15] N. Akai, "Mobile robot localization considering uncertainty of depth regression from camera images", *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1431–1438, 2022. DOI: 10.1109/LRA.2021.3140062.

[16] D. Figurowski and P. Dworak, "Environment mapping algorithm using semantic description and constrained Delaunay triangulation", *Elektronika ir Elektrotechnika*, vol. 25, no. 6, pp. 4–7, 2019. DOI: 10.5755/j01.eie.25.6.24818.

[17] K. Prakash, M. Parimala, H. Garg, and M. Riaz, "Lifetime prolongation of a wireless charging sensor network using a mobile robot via linear Diophantine fuzzy graph environment", *Complex & Intelligent Systems*, vol. 8, pp. 2419–2434, 2022. DOI: 10.1007/s40747-022-00653-5.

[18] T. Fyleris and E. Jasiuniene, "Analytic approach for 2D phased array delay law calculation in case of inspection of objects with complex geometry", *Elektronika ir Elektrotechnika*, vol. 25, no. 2, pp. 28–31, 2019. DOI: 10.5755/j01.eie.25.2.23200.

[19] S. G. Loizou and E. D. Rimon, "Mobile robot navigation functions tuned by sensor readings in partially known environments", *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3803–3810, 2022. DOI: 10.1109/LRA.2022.3148466.

[20] M. Prauzek and J. Konecny, "Optimizing of Q-learning day/night energy strategy for solar harvesting environmental wireless sensor networks nodes", *Elektronika ir Elektrotechnika*, vol. 27, no. 3, pp. 50–56, 2021. DOI: 10.5755/j02.eie.28875.

[21] R. Morales, A. Fernández-Caballero, J. A. Somolinos, and H. Sira-Ramírez, "Integration of sensors in control and automation systems 2020", *Journal of Sensors*, vol. 2022, art. ID 9765679, 2022. DOI: 10.1155/2022/9765679.

[22] S. K. Chaurasiya, A. Biswas, P. K. Bandyopadhyay, A. Banerjee, and R. Banerjee, "Metaheuristic load-balancing-based clustering technique in wireless sensor networks", *Wireless Communications and Mobile Computing*, vol. 2022, art. ID 8911651, 2022. DOI: 10.1155/2022/8911651.