Proceedings of the World Congress on Engineering and Computer Science 2018 Vol II
WCECS 2018, October 23-25, 2018, San Francisco, USA

Asymmetry Regional Boundary Localization for Indoor Robot Navigation

N. Wanadecha, and P. Nilas

Abstract—This paper presents the new methodology of indoor localization based-on Received Signal Strength Indicator (RSSI) of Wireless Sensor Network (WSN) using Asymmetry Regional Boundary Localization (ARBL) which is designed to support mobile robot navigation. Since many types of obstruction such as walls, poles, and traps in the environment may confuse robot trajectory and avoidance strategy, our main objective is to analyze our desire test bed in the form of asymmetry region with the benefit of RSSI map fingerprinting technique. The regions are created from multiple anchor nodes' boundary intersection which is pre-determined by the working range that provide the finest repeatability value. Each region is classified by supervised neural network. When the robot senses the existence of obstacle, that particular region is declared as occupied zone which is not allowed to enter during path planning process. Once the entire virtual map is acquired, suitable path is obtained by Q-learning algorithm, thus the robot can travel through the field. The experiments are performed with 2 different types of test bed. Each of which shows region classification performance of 71.9% and 85.8%, respectively. Finally, to demonstrate the effective of our method, dealing with obstacles in the field, the comparison between conventional Q-learning algorithm and our proposed method is shown. The proposed method not only gives better learning progress but also spends less computational resources, reducing the complexity of the navigation problem.

Index Terms—Asymmetry Regional Boundary, indoor localization, mobile robot navigation, wireless sensor network

I. INTRODUCTION

Localization and navigation are considered to be the most important issues for mobile robot. In fact, the robot needs to know its current location respect to the world. So, it can find a way that suitable enough for reaching the goal in path planning process. One of the fundamental aspect most of the robot needed is the ability to detect obstacle and avoid undesirable collision that leads to harmful consequences. SLAM (Simultaneous Localization and Mapping) is the well-known efficient technique to locate and navigate robot that has been applied widely in real-world application. It makes use of the cooperation of multi-sensor (sensor fusion) such as wheel encoder, range finder, magnetic compass to provide the accurate information in the map. LiDAR technology based on laser scanners has been applied as well as High-resolution cameras in order to get the real-time physical data with high quality from environment. Thus, the higher performance the sensing part, the more accuracy the map is. Unfortunately, its cost is surprisingly expensive for academic research and high computational embedded processor is required for the robot.

In indoor environment, WSN is already a basis of infrastructure. In term of target localization, it acts as the backbone of the system instead of GPS where signal propagation can't be utilized properly. Several localization schemes have been researched and proposed from many researchers. They can be broadly categorized into two classes: range-based and range-free technique. Since range-based localization relates to finding distance between target node and reference node with physical variable such as RSSI, this method provides the result as exact specific point. On the other hand, range-free method is also known as scene analysis that locates target node via other factor such as network structure, heuristic information etc. The output seems to be less specificity comparing to range-based. However, this method provides less complexity process and more robustness against the severe signal fluctuation which is the major cause of error.

SLAM requires high performance equipment that comes with high cost and computational power to finish the work. With this in mind, this paper presents the optional way to locate target using WSN incorporated with affordable price sensors called Asymmetry Regional Boundary Localization (ARBL). This proposed method aims to reduce the complexity of positioning error correction caused by the nature of radio signal. It also allows mobile robot to perceive the entire map, including obstacles and do path planning in further process. This paper is organized as follows. In section 2, literature review is briefly described including related work. Then, the proposed method is introduced thoroughly in section 3. Section 4, system overview is clarified. Section 5, experiment and result is explained, including the comparison between conventional Q-learning navigation algorithm and Q-learning with ARBL technique in simulation program. Finally, the conclusion and future work are summarized in section 6.

II. LITERATURE REVIEW

In [2], [3], [13], [14], [16], scene analysis model is used to locate the target in term of pattern recognition. It is also known as fingerprinting technique. The main idea is matching information, containing characteristic data of each particular zones between pre-measured data stored in database and current run-time data in online phase.

Manuscript received June 06, 2018.
N. Wanadecha is a master student with Instrumentation and Control Engineering Department, King Mongkut's Institute of Technology Ladkrabang, Ladkrabang, Bangkok 10520 Thailand (phone: +6685-075-0126; e-mail: 58601083@ kmitl.ac.th).
P. Nilas is with Instrumentation and Control Engineering Department, King Mongkut's Institute of Technology Ladkrabang, Ladkrabang, Bangkok 10520 Thailand (e-mail: phongchai.ni@kmitl.ac.th).

Although, this technique required preparation phase that takes long time for initialization as well as large data storage to keep the record, its advantage is robustness against error from signal noise which mostly occurred to indoor environment. Due to the uncertainty of positioning result, some work doesn't require the specific output. The concept of geometry regional area is applied. For example, Voronoi diagram [5], [7], [8], Multi-granularity [9], Approximate Point-In-Triangle (APIT). These approach seems to be difficult in mathematical terms and algorithmic complexity is high even though some method can provide the error less than 1 meter approximately. Prior knowledge in signal processing may require in order to manage pre-processed data in proper way. It can be noticed that, machine learning is applied in [2], [4], [6], [14], and [16]. Neural Network is preferable technique in non-linear classification problem comparing to others. Furthermore, Deep Neural Network is used in [14] for function approximation, solving RSSI fingerprints in specific environment. The research is [15] predicts the future trend of localization method that hybrid technology and sensor fusion may provide the efficient result using the combination between WSN, Wi-Fi, and Bluetooth Low Energy (BLE). [2] and [13] prove this idea, reducing positioning error to 1.05 m. Additionally, [1] mentions the hardware modification in order to increase signal quality. Since this work claims that signal cannot be approximated consistently, the absorbing plate is installed underneath antenna in order to avoid interference with signal reflected by ground that caused multi-path effect.

The result of investigation can be concluded that range-free method is suitable for indoor localization with no line-of-sight of signal path. Fingerprinting with neural network is the most popular combination technique that works well and robust against signal fluctuation. Area-based localization may offer location result of the target properly instead of specific coordinates.

III. Concept Overview

As mentioned earlier, our objective aims to create proper trajectory, avoiding any obstruction in the field located by RSSIs from WSN. We derive some benefits from related works above. Asymmetry Regional Boundary Localization can be divided into 4 steps

A. *Regional Boundary*

As introduced in [11], [12], iBeacon, an Apple trademark for its implementation of BLE beacons, uses localization technology with RSSI based ranging. Normally, using only signal strength has many chances to miscalculate the real distance. So, in iBeacon, the RSSI boundary has been considered and separated into different tiers such as immediate, near, far, and out of range (labelled as unknown). Of all the reasons, this idea leads to the appliance of Regional Boundary in this work. RSSI Boundary has been categorized into 2 tiers.

1) *In bounds:* the area between anchor's location and imagine boundary threshold.

2) *Out of bounds:* the area between imagine boundary and the rest of the field.

The boundary size of each anchors depends on signal propagation performance respect to the test bed. Plus, there are five anchors scattered in the field. So, the enclosed area, which is occurred from the intersection among boundaries is called "Region".

B. *Repeatability*

The boundary threshold criterion depends on how good the radio module can perform in the field (like equipment's specifications) when it needs to provide RSSI value in the same distance over and over again. In fact, gage repeatability is a common standard that used to test the equipment stability and accuracy. This factor is concerned, in order to measure the capability of radio sources. The main objective using repeatability is to limit the imaginary line of boundary. The type of repeatability in this context can be separated into 3 parts as follows.

1) *Radius repeatability R_r:* the repetition value of reading RSSIs every 1m radius from the center of each radio source.

2) *Entire field repeatability R_F:* the value that defines repeatability of the anchor acts against the entire map.

3) *Threshold repeatability R_T:* a value of radius repeatability with less deviation from its reference.

To obtain the radius repeatability of anchor, first, rendering imaginary circles with several radius, let the anchor position as a center until the field is covered. Then, in each set of data, gathering all of the RSSIs from every coordinates lying on each imaginary circles. The radius repeatability equation can be determined as follows.

$$R_r = \sqrt{\frac{\sum_{i=1}^{N}(X_i - M)^2}{N-1}} \tag{1}$$

where $M = \frac{1}{N}\sum_{i=1}^{N} X_i$ and $\mathrm{X} = \frac{1}{\mathrm{C}}\sum_{\mathrm{c}=1}^{\mathrm{C}} \mathrm{x}_{\mathrm{c}}$.

Equation (1), X_i denotes the estimated RSSI from coordinates x_c lying on imagine circumference in one radius of each dataset. C denotes the amount of interesting coordinates. M represents the average RSSI value of the radius. N represents the number of dataset. Suppose, we consider the repeatability of 10 dataset, so N=10. Next, the entire field repeatability comes from the average value of all radius. It means that with the condition of environment characteristic in this case, radio source generally provides this amount of error entire area. We use this feature as a reference for the final step. The entire map repeatability equation is described below in equation (2).

$$R_f = \frac{1}{J}\sum_{j=1}^{J} R_{r_j} \tag{2}$$

where j is number of considered radius which the origin point is the center of anchor. Finally, once the entire field repeatability is acquired, the last process is determining how close each radius repeatability compared to the reference. In equation (3), the threshold repeatability is shown as follows.

$$R_{T_i} = \left| R_f - R_{r_i} \right| \tag{3}$$

where R_T represents the absolute different between radius repeatability from *i* meter and the reference value RF. The imaginary line of boundary is assigned to the radius with the finest value. Due to the different propagation pattern of each anchors, the radius repeatability of each anchor may not equally be the same. This reason leads to the conclusion that the region would be asymmetry shape.

ISBN: 978-988-14049-0-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

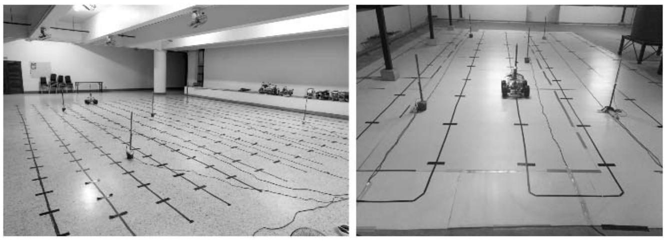

Fig. 1. Indoor test bed with 5 Xbee anchors and mobile robot. (Left) 10m x 10m size. (Right) 10m x 6m size.

C. Region Classification

In MATLAB, there are neural network tools that provides many training algorithms. However, there is no absolute solution that fits to any dataset. Trial and error method is needed to perform neural network hyper-parameter testing in order to obtain appropriate configured model that fits RSSI fingerprint dataset. Afterwards, the trained model is utilized in online mode, classifying the region respect to perceived RSSIs vector.

D. Building complete map

Once the system identifies the region from trained model classification in earlier process, the robot detects any obstacles in the field and records its discovered area into storage memory as an occupied region which is not allowed entrance. For vacant area, on the other hand, it is recorded as unoccupied region. Finally, all of "No trespassing" area and vacant entrance area are gathered and formed together into a complete map.

IV. System design

A. Xbee network and test bed

Wireless Sensors Network in this work is composed of 6 Xbee series 2 modules. Five of them are scattered all over the test bed acting as Beacons or sometimes also known as Anchors. The sixth module is called target node installed to the mobile robot. Every module attaches the 9dBi Omni-directional antenna, allowing the signal spread widely in horizontal axis.

As depicted in fig. 1, indoor clear space test bed is chosen with 2 different sizes: 10m x 10m and 10m x 6m. The test bed has been divided into a grid with 1 meter spacing. For test bed 1, all anchor node from number 1 to 5 are placed at coordinates (2,8), (8,8), (2,2), (8,2), (5,5), respectively. For test bed 2, all anchor nodes from number 1 to 5 are placed at coordinates (1.5, 9.5), (4.5, 9.5), (1.5, 1.5), (4.5, 1.5), (3.5, 5.5), respectively. Black line installed to the field is only for mobile robot tracking purpose in offline mode. The robot tracks the line until black cross is reached. It stops moving, collecting RSSI fingerprint data at every points. Once all location points are collected, the robot moves to home position at the start point and all RSSI fingerprints are considered as one dataset. However, the tracking line has no effect to the process in online mode. It has been removed before run-time mode starts.

B. Mobile robot

Robot is driven by 4-wheel 12VDC motor which is controlled by Arduino microcontroller. It comprises of sensors, line following infrared sensor and infrared range

TABLE I: THRESHOLD REPEATABILITY OF TEST BED 1

Threshold Repeatability : test bed 1 (10 datasets)										
Radius / Anchor	1m	2m	3m	4m	5m	6m	7m	8m	9m	10m
A1	0.93	0.56	0.62	0.54	0.72	0.85	0.98	0.79	0.81	0.77
A2	0.16	0.83	0.3	0.24	0.04	0.3	0.05	0.82	0.39	0.43
A3	0.21	0.42	0.17	0.62	0.06	1.02	0.33	0.22	0.75	0.38
A4	0.39	0.29	0.66	0.29	0.62	0.12	0.35	0.83	0.53	0.59
A5	0.11	0.13	0.12	0.33	0.77	0.45	0.5	-	-	-

TABLE II: THRESHOLD REPEATABILITY OF TEST BED 2

Threshold Repeatability : test bed 2 (10 datasets)										
Radius / Anchor	1m	2m	3m	4m	5m	6m	7m	8m	9m	10m
A1	-	0.48	0.07	0.42	0.77	0.17	0.17	0.97	0.22	-
A2	-	0.37	0.48	0.64	0.02	0.06	0.99	0.39	0.15	-
A3	-	0.10	0.06	0.21	0.60	0.18	1.31	0.16	0.33	-
A4	-	0.56	0.41	0.41	0.32	0.06	0.25	0.68	0.50	-
A5	0.22	0.16	0.89	0.37	0.21	1.84	-	-	-	-

finder for obstacle detection purpose. They are attached in front of the robot, providing object detection range about 0.2m. Moreover, Xbee target node is controlled by second micro-controller separately, reading RSSI from 5 anchor nodes with time interval of 500ms each. Finally, the robot communicates to Firebase, cloud database via Wi-Fi, requesting and responding all of necessary information such as RSSI values, update counts, movement control signal, obstacle detection signal etc.

C. Operator

The pre-programmed application, written for Android platform, is capable of several tasks. For instance, operating mode option (Fingerprint data collection or run-time mode), saving raw information in .xls file, controlling robot, and monitoring region output from cloud database.

D. Base station

MATLAB is responsible for training and testing classification neural network with raw data sets, collected from entire field in offline mode. Its responsible also includes data pre-processing and model performance analyzing. In online mode, m-file script is written and executed to handle classification task with current RSSIs vector, sending back the classified output to database. Also, it creates communication between the station and cloud database via RESTful API.

V. Experiment and result

A. Establishing regional boundary: Repeatability estimation

Repeatability of Xbee module assigned as anchor node is statistically tested with pre-collected 10 datasets. The range that provides good repeatability is assigned as a threshold of the boundary. The whole process is clearly depicted in concept overview. Table 1 and 2 show the calculated deviation repeatability from 10 datasets of test bed 1 and 2, respectively. The finest repeatability value is the least value in a row. For each anchor, the selected range with the best repeatability value is assigned as a boundary threshold and any enclosed shape becomes asymmetry regional area with several size. Fortunately, both test bed can be divided into 13 regions. It can be noticed that, both tables have some blank cells. For test bed 1, maximum coverage area of anchor 5 is 7m radius. Further than that, the radius is out of the test bed. Same reason happens to test bed 2. However, for 1m radius of anchor 1 to 4, there is no collected RSSI data to be calculated in that particular point. Therefore, the repeatability in this range is ignored.

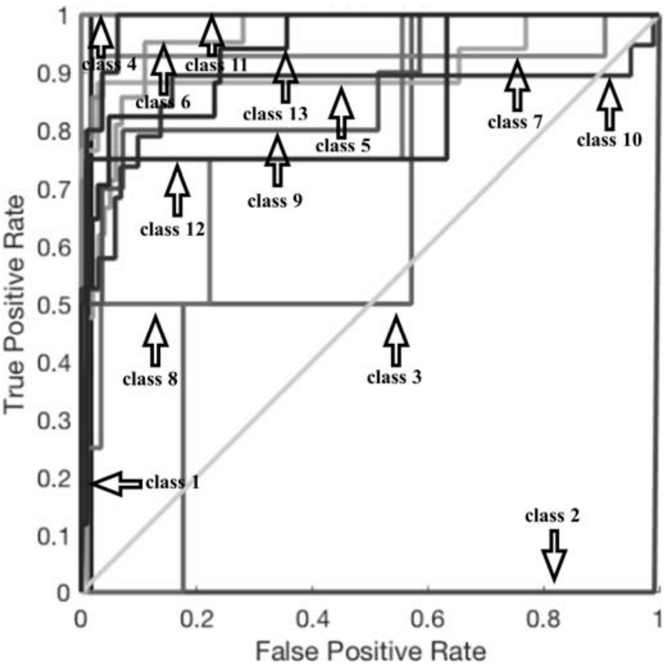

Fig. 2. Receive Operating Characteristic (ROC) graphs show classification result of test bed 1. This graph represents testing phase result.

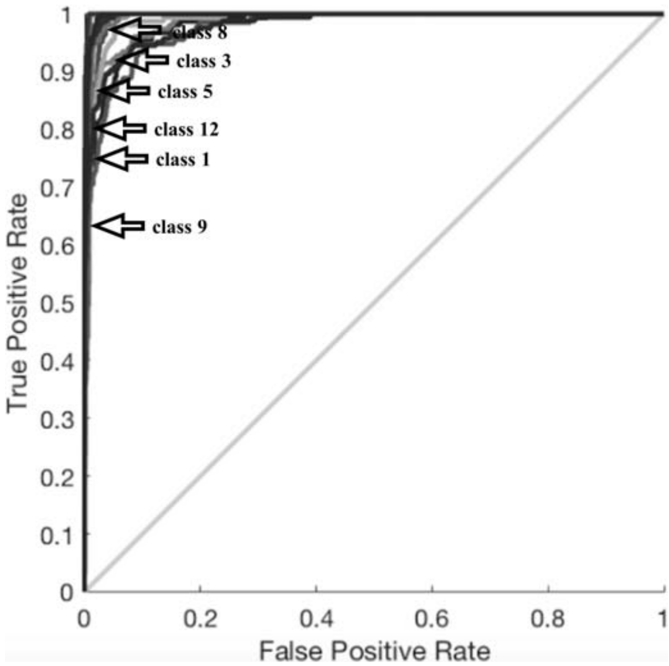

Fig. 3. Receive Operating Characteristic (ROC) graphs show classification result of test bed 2. This graph represents testing phase result.

B. Neural network hyper-parameter testing and regional classification performance

In this part of experiment, artificial neural network is involved as a regional classification model. Since there are 2 types of environment, neural network needs to be carefully design to suit its own data characteristic. The network structure for test bed 1 is configured as follows. Input layer consists 5 input nodes (5 nodes from 5 anchor raw RSSI). Hidden layer consists of 3 layers with 12 nodes each. Hyperbolic tangent Sigmoid is applied as activation function. Output layer consists of 13 nodes representing the 13 region needed to be identified in the form of one-hot encoding vector. Bayesian Regularization, training algorithm, is applied as a training algorithm. This configuration is randomly chosen based on MATLAB documentation without any experience or priori knowledge.

The result shows that training and testing performance are 83.5% and 71.8%, respectively. Also, receiver operating characteristic (ROC) graph shows the classification result from testing phase. As illustrated in fig. 2, true positive rate means model correctly classifies the data as expected and false positive rate means the model gives absolute false prediction. Obviously, the model is too complexity. It seems to be learning noise data instead of its true pattern. Although, it gives over-fitting behavior, the trained model still provides some correct region output using current RSSI data in run-time mode.

However, for test bed 2, to prevent the over-fitting model, hyper-parameter testing of neural network is performed in order to derive the best model performance that fits RSSI fingerprinting map properly. Several strategies are applied to the model. For example, L2 regularization preventing the influence from outlier data, class imbalance fix with added White Gaussian noise increasing robustness of the model. The configuration values are set up as follows. Learning rate and regularization value is chosen within the array of [0.1, 0.01, 0.001, 0.0001, 0.00001]. Learning algorithm is tested between *trainscg* and *trainrp*. Activation function is tested between *tansig* and *poslin*. The testing includes hidden layer number of 1 and 2 and hidden nodes from 20, 40, 60, and 80. Output layer is assigned with softmax function in order to normalize the predicted value between 0-1. Lastly, since training data affects model performance, 10-fold cross validation is utilized in order to prove the true performance by randomly split raw data into 2 partitions: 90% of entire data is assigned to be training set, the rest 10% is assigned to be testing set. This process is repeated until all of the data completely shuffled. Aftermath, the average performance is calculated. Hyper-parameter testing result shows that, the model that providing best result ends up with following configuration. Learning rate and Regularization = 0.01. Training algorithm = trainscg. Activation function = tansig. Hidden layer = 2. Hidden node = 80. Average training and testing performance derived from confusion matrix are 87.61% and 85.76%, respectively. Furthermore, ROC graph depicted in fig. 3 shows true positive rate and false positive rate of classification result in testing phase. The entire data falls into true positive rate. In other words, this model is good fitting to RSSI-fingerprinting map of test bed 2, providing good generalization result.

C. Comparison between conventional Q-learning and Q-learning with ARBL

To demonstrate the effectiveness of the proposed algorithm, the navigation experiment is setup in simulation program written in python. In [10], the mobile robot navigation based Q-learning technique addresses the problem of robot behavior when facing different types of obstacle. For example, objects, walls, and traps. Also, another drawback is, Q-learning algorithm is a tabular method requiring large size of memory to store Q-value of relate state, action and reward. Although, [10] offers fuzzy logic implementation reducing the complexity of the navigation, the robot still needs some strategy to handle emergency situation like U-trap. On the other hand, ARBL identifies any obstacle as a region. Therefore, the opportunity getting stuck in the trap is relieved.

The simulation of test bed is created as a grid. The robot movement restricts to only 4 actions: [up, down, left, right]. Discount factor is set as 0.9. Learning rate is 0.1. Epsilon-greedy for random exploration is initialized with 0.5. Every step the agent takes is penalized with -0.1. Reward at goal position is 10. Test bed simulation is illustrated in fig. 4.

Red triangle represents goal position. Green circle represents start position of the robot. Black square represents occupied zone which is not allowed to pass. Agent needs to find the optimal policy (represented by blue line) to get to the goal. Q-value implies how good the taken action can be valuable for the agent. It learns from experience in each episode, moving through the field. Figure 5 and 6 illustrate differentiate of Q-learning process along with episode until convergence is reached (differentiate

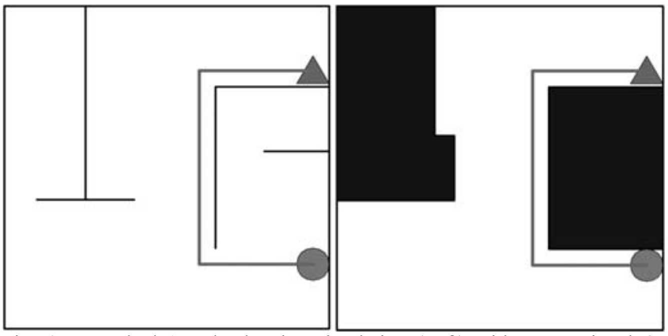

Fig. 4. Test bed 1 path planning simulation (Left) with conventional Q-learning, (Right) ARBL with Q-learning.

remains zero), meaning the optimal policy is obtained. Figure 5 shows test bed 1's learning performance of Q-learning with ARBL (red graph) against conventional method (blue graph). Figure 6 shows test bed 2's. In test bed 1, it can be noticed that, our proposed method offer better progress, reaching convergence at episode 200 earlier than the conventional one. Since the obstacle has been considered as region, the complexity of path planning problem is reduced. Our proposed method outperforms in computational time and memory size. Q-learning with ARBL spends 1.12 minutes and Q-value hash table size of 59 while conventional method spends 2.04 minutes with hash table size of 81.

For test bed 2, learning progress appeared to be the same trend. Due to the smaller size of environment, computational time, hash table size, and learning progress is less than test bed 1's situation. Both methods reach convergence at 200 episode approximately. Although our proposed method learning progress seems to be fluctuated before converged, computational time is only 41 seconds and 32 of hash table size is spent while conventional method wastes 1.17 minute and 48 hash table size to solve the problem.

VI. Conclusion

This paper proposes Asymmetry Regional Boundary Localization (ARBL) designed to support indoor navigation for mobile robot. Our proposal exploits the advantage of area-based range-free localization with RSSI from WSN in the field to locate mobile target and obstacles in the form of asymmetry region. Asymmetry region is established from the intersection of each anchor's boundary which occurred from estimated communication radius with the finest

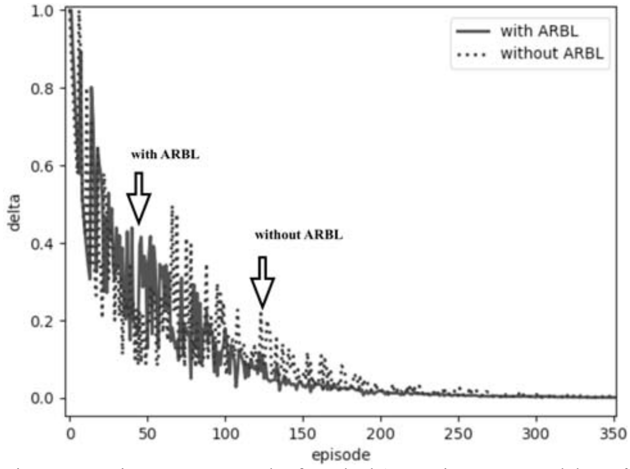

Fig. 5. Learning progress graph of test bed 1. Y axis represents deltas of Q-value. X axis represents episode.

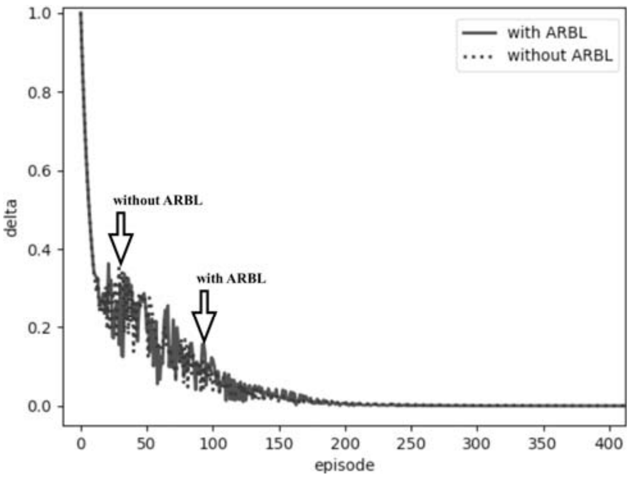

Fig. 6. Learning progress graph of test bed 2. Y axis represents deltas of Q-value. X axis represents episode.

repeatability of reading RSSI in offline phase. For test bed 1, although the model is over-fitting, trained neural network gives the classification performance of 71.9% to classify and locate region with real-time RSSI fingerprint vector, acquiring from mobile robot. However, for test bed 2, hyper-parameter test is performed in advance to fix over-fitting problem. Eventually, the model turns out to be good with classification performance of 85.8%. ROC graphs show the correspondence of both of the results. Finally, in navigation estimation task, ARBL is applied to Q-learning algorithm, comparing with conventional approach in simulation. The result shows that our method outperforms in case of computational time and memory resource reduction. Also, in test bed 1, the proposed method's learning progress reaches convergence point with less episode than the other. Due to the fact that the obstacle is considered as regional area, the complexity of navigation problem is decreased. Moreover, the complication of avoidance strategy when the robot encounters emergency situation or gets stuck within U-trap, is no longer required because the suitable path is already planned. This method accomplishes in cost and high computational resource reduction for the robot's component.

References

[1] J. Graefenstein and M. Essayed Bouzouraa, "Robust Method for Outdoor Localization of a Mobile Robot Using Received Signal

Strength in Low Power Wireless Networks" in *IEEE International Conference on Robotics and Automation*, Pasadena, CA, May, 19-23, 2008, pp. 33–38.

[2] Z. Farid, R.Nordin, M. Ismail, and N.F. Abdullah, "Hybrid Indoor-Based WLAN-WSN Localization Scheme for Improving Accuracy Based on Artificial Neural Network*".* Mobile information system, Vol. 2016, May, 2016, pp. 123–135.

[3] R. Yasmine and L. Pei, "Indoor Fingerprinting Algorithm for Room Level Accuracy with Dynamic Database*".* IEEE UPINLBS, Shanghai, P.R., China, Nov, 3-4, 2016, pp. 113-121.

[4] N.A. Sabto and K. Al Mutib, "Autonomous mobile robot localization based on RSSI measurements using RFID sensor and neural network BPANN," Journal of King Saud University - Computer and Information Science. Oct, 23, 2012, pp. 137-143.

[5] S. Cai, H. Pan, Z. Gao, N. Yao, and Z. Sun "Research of localization algorithm based on weighted Voronoi diagrams for wireless sensor network" *Journal on Wireless Communications and Networking 2014*.

[6] L-Z. Zhao, X-b. Wen, and D. Li, "Amorphous Localization Algorithm Based on BP Artificial Neural Network" *International Journal of Distributed Sensor Networks. Vol. 2015*, Jan, 2015.

[7] J. Qin, S. Sun, Q. Deng, L. Liu, and Y. Tian, "Indoor Trajectory Tracking Scheme Based on Delaunay Triangulation and Heuristic Information in Wireless Sensor Networks" Sensor 2017, June 2017.

[8] D. Blanco, B.L. Boada, and L. Moreno, "Localization by Voronoi Diagram Correlation" IEEE International Conference on Robotics & Automation, Seoul, Korea, May, 21-26, 2001, pp.4232-4237.

[9] F. Shang, Y. Jiang, A. Xiong, W. Su, and L. He, "A Node Localization Algorithm Based on Multi-Granularity Regional Division and the Lagrange Multiplier Method in Wireless Sensor Networks". Sensor 2016, Nov, 2016.

[10] L. Khriji, F. Touati, K. Benhmed, and A. AI-Yahmedi, "Mobile robot Navigation Based on Q-learning Technique" International Journal of Advanced Robotic Systems, Vol. 8, No.1, 2011, pp. 45-51.

[11] "Getting Started with iBeacon Version 1.0", June, 2, 2014.

[12] N. Newman, "Apple iBeacon technology briefing" Journal of Direct, Data and Digital Marketing Practice, vol. 15, no.3, pp. 222-225, Jan, 2014.

[13] P. Kriz, F. Maly, and T. Kozel, "Improving Indoor Localization Using Bluetooth Low Energy Beacons", Mobile Information System, Vol. 2016, March, 2016.

[14] S. Timotheatos, G. Tsagkatakis, P. Tsakalides, and P. Trahanias, "Feature Extraction and Learning for RSSI based Indoor Device Localization".

[15] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of Wireless Indoor Positioning Techniques and Systems", IEEE TRANSACTIONS ON SYSTEM, MAN, AND CYBERNATICS-PART C: APPLICATION AND REVIEWS, Vol.37, No.6, Nov, 2008, pp.1067-1078.

[16] C. Laoudias, D. G. Eliades, P. Kemppi, C.G. Panayiotou, and M. M. Polycarpou, "Indoor Localization Using Neural Networks with Location Fingerprints", ICANN 2009, Part II, 2009, pp. 954-963

ISBN: 978-988-14049-0-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)