Simulation of Mobile Robot Navigation Utilizing Reinforcement and Unsupervised Weightless Neural Network Learning Algorithm

Yusman Yusof
Industrial Automation Section
Universiti Kuala Lumpur Malaysia France Institute
Bandar Baru Bangi, Selangor, Malaysia
yusman@unikl.edu.my

H. M. Asri H. Mansor[1]
H. M. Dani Baba[2]
Faculty of Electrical Engineering
University Technology MARA
Shah Alam, Selangor, Malaysia
pakngah@salam.uitm.edu.my[1]
mdani074@salam.uitm.edu.my[2]

***Abstract*—The approach of transforming human expert knowledge into computer program only allow a system to solve foreseen and tested outcomes compared to a system having self-learning capabilities. This paper will summarize and discuss the research, design and implementation of a novel self-learning algorithm which combines: (a) Q-Learning - A reinforcement learning algorithm; and (b) AutoWiSARD – An unsupervised weightless neural network learning algorithm. The self-learning algorithm was implemented in an autonomous mobile robot navigation and obstacle avoidance system in a simulated environment. The AutoWiSARD algorithm identifies, differentiates and classifies the obstacles and the Q-learning algorithm learns and tries to maneuver through these obstacles. This novel hybrid technique allows the autonomous system to acquire knowledge, learn and record experience thus attaining self-learning state. The final result shows the simulated mobile robot was able to differentiate various shapes of obstacles such as corners and walls; and create complex control sequences of movements to maneuver through these obstacles.**

Keywords—reinforcement learning; Q-learning; AutoWiSARD; autonomous navigation; unsupervised learning; weightless neural network; robot simulation

I. Introduction

A typical approach of developing an autonomous system is by translating pre-defined expert knowledge into computer program or by utilizing Artificial Intelligence (AI) algorithm which was exhaustively trained. Prior to system deployment, the developed autonomous system must be thoroughly tested and trained with various anticipated possible outcomes, and program had to be recoded when new outcomes are discovered. These approaches make the system inflexible and rigid, in which the system was strictly developed to solve only foreseen and tested circumstances. Any unforeseen circumstances which occur when the system is currently running will result to failure.

According to Yousif et al. [1], a Self-Learning System (SLeS) having automated knowledge acquisition capabilities that is able to extract knowledge and learn from past experience, generate the associated data and rules then act upon these rules, is the way to go forward and offers more flexibility. Yousif et al. had successfully demonstrated the self-learning concept, but the technique employs rigid rules based and path planning algorithm that were hard coded into the system.

As an alternative, the hybrid AI self-learning algorithm as proposed in this paper is found to be more adaptive, will make a system less inflexible. The formulated self-learning algorithm employs; a) Unsupervised Weightless Neural Network which handles generalization issues; b) A reinforcement learning algorithm which act as a reward and penalized function by learning the optimal policy from its history of interaction with the environment [2], [3].

In order to realize self-learning autonomous system implementation, simulated application of mobile robot wandering in an unknown environment and avoiding obstacles utilizing self-learning algorithm will be developed. The mobile robot application is chosen because the process of knowledge acquisition and learning from experience while wandering and navigating through obstacles is similar to human learning process.

II. Literature Reviews On Self-Learning and Autonomous System

Followings are the comprehensive findings from reviews conducted on diverse self-learning and autonomous system:

1) Extensive search was conducted on self-learning systems, and the result reveals several relevant technical research which focuses on self-learning development since 2005 [1], [4]–[10]; only Yousif et al. [1] involves in developing self-learning system for an autonomous system implemented in mobile robot and while the rest were meant to be used in offline diagnostic systems;

2) In terms of self–learning algorithm, fault tree, cased-based reasoning, back propagation and rule-based were being applied. All of these algorithms requires prior training, some of the rules and formulation were being hard-coded into the system and consume large amount of memory and require long processing time;

3) Most of the Artificial Intelligence algorithms consume resources such as computing power and storage capacity, both of which seem at odds with an embedded controller used by autonomous system [11]. Embedded autonomous system face resource-constrained issues [12], [13]; processor speed, storage capacity, run-time memory and other hardware related matters;

4) Autonomous system constitutes of finite states/situations and actions; and expert knowledge is translated into computer program used to activate these actions in order to manipulate the environment and can be classified as followings:

a) A system whereby anticipated states are known beforehand, therefore can be generalized by using pre-trained Neural Networks [14]–[17] and expert knowledge was pre-programmed to handle number of actions;

b) Or, rather than using pre-programmed expert knowledge, a reinforcement learning algorithm can be applied in order to make the system learn as time progresses [2], [18], [19].

5) Autonomous system applications developed by Bagnall, Claveau, Nurmaini, Strauss and others [3], [14], [16], [20], [21] demonstrates that both reinforcement learning and weightless neural network algorithm can be successfully applied in autonomous systems which implemented in resource constraint environment;

6) Both weightless neural network and reinforcement learning algorithm are fast and only uses small amount of memory. Therefore, it can be implemented in embedded controller for self-learning autonomous system implementation.

III. Development of Self-Learning Algorithm

Based on findings, autonomous systems will likely have the tendency to fail because the system was programmed and trained to handle anticipated outcomes only. Therefore, a self-learning method would be a more viable solution. When programmed with self-learning algorithm, the autonomous system will be able to:

- Discover, classify and memorize new knowledge and;
- Infer based on the newly acquired knowledge.

If the system able to perform task better than before, therefore it can be said that learning is achieved when the system is capable of adapting to a task, such that when the task is repeatedly performed the system will perform the task efficiently than previous [22].

From the literature review, it was established that the combination of reinforcement learning and unsupervised weightless neural network able to be transformed into self-learning algorithm. Out of the various existing algorithm, the Q-Learning and AutoWiSARD were chosen because of their simplicity and efficiency.

A. Unsupervised Weightless Neural Network

Past researches by others had successfully demonstrated that WiSARD [23], a type of Weightless Neural Networks (WNN) can be successfully applied to classify the states or different types of obstacles or environments in reactive system such as mobile robot applications [15], [16], [24], [25]. By using WiSARD these applications require exhaustive training and retraining of anticipated situations or states.

In order to self-learn, unsupervised learning will be utilized to create new information about states or situations or environments. AutoWiSARD is an unsupervised learning version of WiSARD algorithm. Shown in Fig. 1 is the WiSARD WNN.

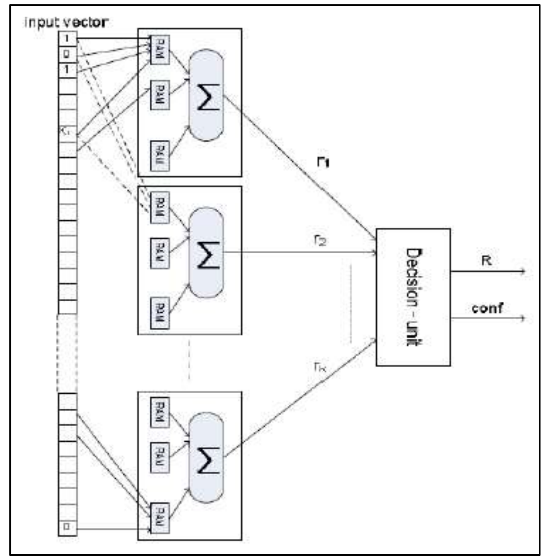

Fig. 1. WiSARD WNN structure [17]

Using WiSARD, sensor input are mapped as binary pattern onto a RAM type discriminator during training process. When recognizing, the input will be matched with each and every RAM discriminator in the WNN. The outcome *R*, of each RAM discriminator will be used to calculate the confidence using equation (1). The input is said to belong to a class represented by the RAM discriminator if it has the highest *c* value.

$$c = R \,/\, \textit{Number of tuples per RAM} \tag{1}$$

AutoWiSARD is a modified WiSARD algorithm which makes the WiSARD learn by unsupervised training. By manipulating the *R* value with a learning window shown in Fig. 2 and applied with the learning policy shown in Fig. 3.

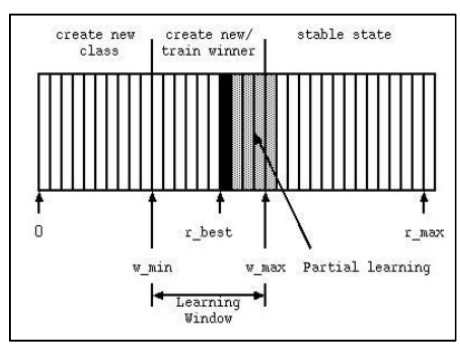

Fig. 2. AutoWiSARD *R* learning window [26]

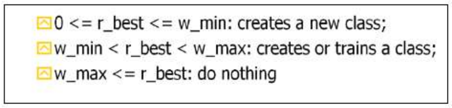

Fig. 3. AutoWiSARD learning policy [26]

AutoWiSARD was originally used to classify hand written characters [26], with modifications and fine tuning, it can be adopted to be used in SLeS to autonmously classifies different states.

B. Reinforcement Learning

Using Q-Learning (a reinforcement learning algorithm), the information about state-actions pairs is presented in the form of table. Fig. 4 shows the Q-Learning algorithm where: $Q(s, a)$ – component of Q table, s – current state, s' – next state, a – current action, a' – next action, r – reward, α – learning rate, and γ – discount factor.

Initialize $Q(s,a)$ arbitrarily
Repeat (for each episode):
  Initialize s
  Repeat (for each step of episode):
    Choose a from s using policy derived from Q (e.g. ε-greedy)
    Take action a, observer r, s'
    $Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma max_{a'} Q(s',a') - Q(s,a)]$
    $s \leftarrow s'$

Fig. 4. The Q-Learning algorithm [27]

In Q-Learning, the information about state-actions pairs is presented in the form of table. Referring to algorithm shown in Fig. 5(a), Q values in the state action-pairs in the table will be updated progressively. For example, in the case of mobile robot, as it wanders, encounters and avoiding obstacles the Q values of the robot action will be updated. As time progresses, the state-actions that produces positive result will have higher Q values and the robot will select action with highest Q value to navigate and avoiding obstacles.

C. Self-learning Algorithm

Autonomous system which employs WiSARD or Q-Learning requires rigorous training to handle situation originated from predetermined states or situations. As an alternative self-learning algorithm which integrates both AutoWiSARD and Q-Learning does not require prior knowledge, instead they will progressively distinguish the states and will learn to react. Both AutoWiSARD and Q-learning algorithm are small and fast which suited for resource constrained embedded system implementation.

Fig. 5 highlight the difference between hybrid Self-learning System (SLeS) and Q-learning. In SLeS, AutoWiSARD will discover and classify different states, while Q-learning will learn and determine the action to be taken. In summary the proposed self-learning system (SLeS) will consist of:

- An unsupervised weightless neural network utilizing AutoWiSARD algorithm will be used to classify and create new states or obstacles;
- A reinforcement learning which employs Q-Learning algorithm will progressively learn to avoid obstacles by providing reward if the outcome of the action taken is positive and to penalize if the outcome is negative.

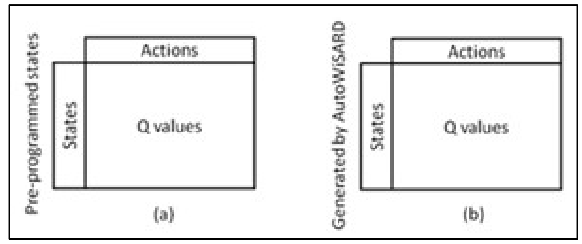

Fig. 5. Comparison of (a) Q-learning and (b) hybrid self-learning algorithm

IV. MOBILE ROBOT SIMULATION

Mentioned earlier, simulation the self-learning algorithm will be implemented in mobile robot navigation application in order to facilitate and expedite the SLeS algorithm development.

A. Simulation Environment

The self-learning algorithm was simulated in an open-source Simple 2D Robot Simulator[28] in Python+Pygame simulator developed by M. Agapie.

1) Mobile Robot Setup: Fig. 6 shows the simulated mobile robot attached with thirteen sonar sensors. Each of this sensor will provide an approximation of the distance of an object. The distance data will then be mapped onto a 6x5 array for AutoWiSARD input.

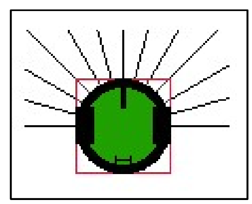

Fig. 6. Mobile robot with 13 sonar sensors

2) Sensor Data for AutoWiSARD Input: Table I represents the 6x5 array which holds the value of 0 or 1 used as input pattern for AutoWiSARD. Given sensor input of s_x, where x ranging from 0 to 12 representing the thirteen sensor attached to the robot and t which represents distance threshold; if $s_x \leq t$, the column value will either be set to 1 or otherwise 0.

TABLE I. SENSOR PATTERN MAPPING RULES REPRESENTED BY 6X5 ARRAY FOR AUTOWISARD INPUT

$s_5 \leq t$	$s_0 > t$	$s_6 \leq t$	$s_7 \leq t$	$s_{12} > t$
$s_4 \leq t$	$s_1 > t$		$s_8 \leq t$	$s_{11} > t$
$s_3 \leq t$	$s_2 > t$		$s_9 \leq t$	$s_{10} > t$
$s_2 \leq t$	$s_3 > t$		$s_{10} \leq t$	$s_9 > t$
$s_1 \leq t$	$s_4 > t$		$s_{11} \leq t$	$s_8 > t$
$s_0 \leq t$	$s_5 > t$	$s_6 > t$	$s_{12} \leq t$	$s_7 > t$

Fig. 7 and 8 shows the examples of the pattern represented by different obstacles when sensor data were mapped onto 6x5 array. The grey column represent 0, while the black column represent 1.

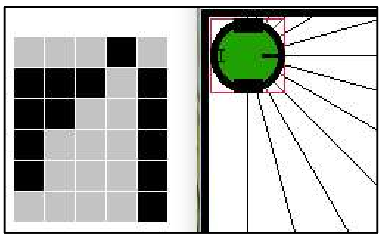

Fig. 7. Mapped sensor pattern when robot not facing obstacle

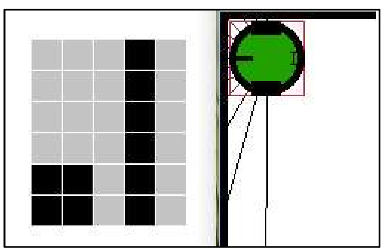

Fig. 8. Mapped sensor pattern when robot facing obstacle

3) Q-learning Key Parameters: All the Q-learning parameters used throughout the experiment are shown in Table II and III. Compared with a typical Q learning algorithm the states must be predetermined but in the self-learning algorithm the states were dynamically classified by AutoWiSARD based on sensor information.

TABLE II. Q-LEARNING REWARD FUNCTION

Actions	Reward
Move forward and hit obstacle	-0.7
Move forward and did not hit obstacle	1
Spin in same direction	-0.1
Spin in opposite direction	-0.3

TABLE III. Q-LEARNING PARAMETERS

Parameters	Value
α – learning rate	0.1
γ – discount factor	0.9
ε-greedy exploration algorithm (ε value)	0.1

4) Self-learning Algorithm

The self-learning algorithm is a loop can be described as follows:

- ***System***: *Read the sensors.*
- ***AutoWiSARD***: *Classifies and determines in what state the robot is.*
- ***Q-learning***: *Select the action according to the state and values of the corresponding action.*
- ***Q-Learning***: *execute the selected action.*
- ***System***: *Read the sensors*
- ***Q-Learning***: *Computes the reward, calculates and updates the state-action pair(s) Q values.*

V. MOBILE ROBOT SIMULATION

Based on the set parameters, the system were tested on three different environment and the results were promising.

A. Test Result: Environment 1

Graphs in Fig. 9 describes the outcome when the robot was initially introduced to a new environment shown in Fig. 10. Prior to this, the robot did not have any knowledge about the obstacles and ways to avoid them.

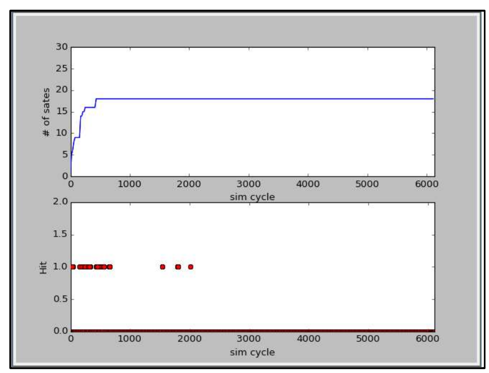

Fig. 9. Observed result for environment 1

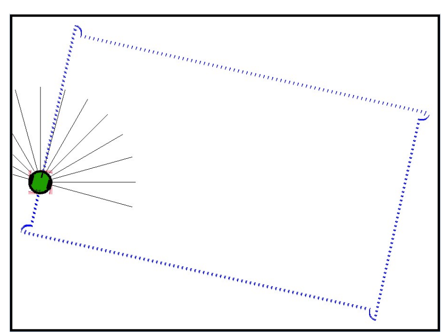

Fig. 10. Traced path in environment 1

Shown in Fig. 9, it took 500 cycles for the robot to discover seventeen classes of obstacles and find ways to navigate through them. The finalized path taken by the robot wanders in this environment is shown in Fig. 10

B. *Test Result: Environment 2*

It was observed that after gathering knowledge on various obstacles and finding ways to navigate in the first environment, the robot effortlessly wanders in the second environment without hitting any obstacles because discover any new class of obstacle as shown in Fig. 11 and Fig. 12 respectively.

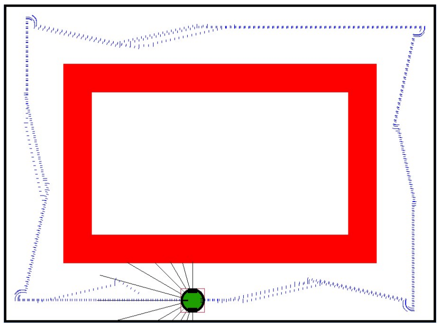

Fig. 11. Traced path in environment 2

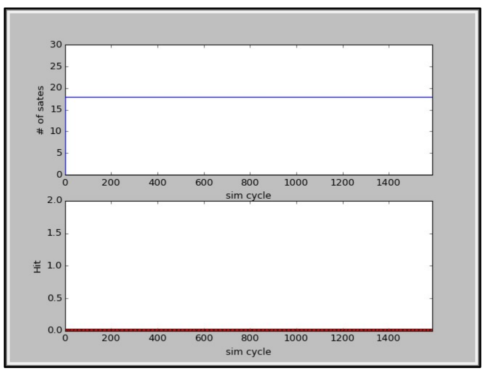

Fig. 12. Observed result for environment 2

C. *Test Result: Environment 3*

The result shown in Fig. 13 and 14 indicates that while wandering in a new environment the robot was able to discover three new classes of obstacles and able to find other ways of avoiding them.

D. *Key Findings*

Compared to research done by others using either weightless neural network or Q-learning, the autonomous system must be thoroughly trained, tuned and to avoid predetermined obstacles. Using self-learning algorithm, the mobile robot was able to gather knowledge on avoiding obstacles while wandering in dynamically changing environment and prior knowledge about anticipated obstacles or states was not required to be programmed into the systems.

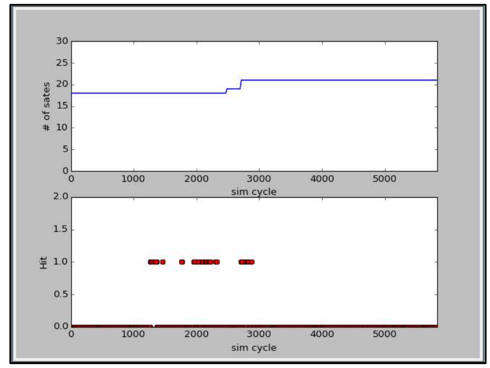

Fig. 13. Observed result for environment 3

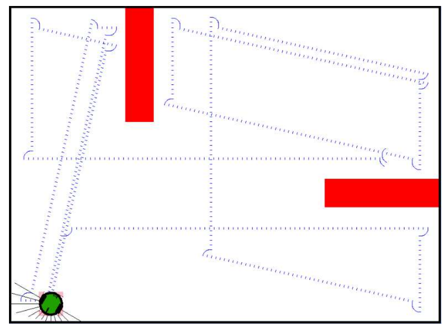

Fig. 14. Observed result for environment 3

Conclusion, Concurrent And Future Work

In this paper we formulated a novel hybrid self-learning algorithm by merging AutoWiSARD and Q-learning algorithm. The self-learning algorithm was implemented in an autonomous mobile robot navigation simulation. The simulation result verifies that the algorithm enables the robot to self-learn. Without prior knowledge of its environment the robot was able to differentiate various types of obstacles, keeps learning and correcting itself to avoid them with more efficiency.

In another concurrent work, the self-learning algorithm was implemented in a physical mobile robot which wanders in a real world while navigating through dynamically changing environment. The implementation was successfully and produces promising results.

In the future the self-learning algorithm will be implemented in other types of application in order to demonstrate its adaptability and capability of solving other problems.

The currently formulated self-learning algorithm will be extended to include an autonomous discovery of reward thus making the more truly independent from human intervention.

Other approach of formulating self-leaning algorithm will be investigated and implemented utilizing other combination of unsupervised and reinforcement learning algorithm; and implemented in other system which requires autonomous learning.

REFERENCES

[1] R. W. Yousif and M. A. H. Mansor, "Design and Simulation of a New Self-Learning Expert System for Mobile Robot," in *World Academy of Science, Engineering and Technology*, 2009, pp. 997–1002.

[2] H.-G. A.-M. Víctor Ricardo Cruz-Álvarez, Enrique Hidalgo-Peña, "A line follower robot implementation using Lego's Mindstorms Kit and Q-Learning," *Acta Univ.*, vol. 22, pp. 113–118, 2012.

[3] C. Strauss and F. Sahin, "Autonomous navigation based on a Q-learning algorithm for a robot in a real environment," *2008 IEEE Int. Conf. Syst. Syst. Eng. SoSE 2008*, 2008.

[4] L. Chen and J. Li, "Development and application of blast furnace expert system with self-learning function based on pattern recognition," *Dongnan Daxue Xuebao (Ziran Kexue Ban)/Journal Southeast Univ. (Natural Sci. Ed.*, vol. 42, no. SUPPL. 1, pp. 117–121, 2012.

[5] H. Fan, Y. Yan, and F. Yang, "Research of dynamic linked lists for the fault tree analysis process," in *Proceedings of 2012 International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering, ICQR2MSE 2012*, 2012, pp. 93–96.

[6] U. Kamath, "Self-learning expert systems using rule classifier in detection engines," in *International Conference on Artificial Intelligence and Pattern Recognition 2008, AIPR 2008*, 2008, pp. 224–227.

[7] C. Kirby, A. Sadlier, C. Wood, and M. Vinther, "Filling the Experience gap in the Drilling Optimization Continuous Improvement Cycle Through a Self-Learning Expert System." Society of Petroleum Engineers.

[8] L. Liu, Y. Wang, and M. Guo, "The research and application of the self-learning expert system based on BP network," in *Fourth International Conference on Machine Learning and Cybernetics*, 2005, no. August, pp. 18–21.

[9] D. Malzahn, "Assessing - learning - improving, an integrated approach for self assessment and process improvement systems," in *Proceedings of the 4th International Conference on Systems, ICONS 2009*, 2009, pp. 126–130.

[10] E. Piga and A. Geschiere, "Self learning expert system (SLES) for power transformers," in *IET Conference Publications*, 2009, no. 550 CP.

[11] C. Gregg, "Genetic Algorithms in Autonomous Embedded Systems," 2009.

[12] M. Pohronská, "Implementing Embedded Expert Systems via Programmable Hardware," *Inf. Sci. Technol.*, vol. 4, no. 2, pp. 10–19, 2012.

[13] C. Seitz and R. Schönfelder, "Rule-based owl reasoning for specific embedded devices," *Semant. Web–ISWC 2011*, 2011.

[14] M. Gillham, B. McElroy, G. Howells, S. Kelly, S. Spurgeon, and M. Pepper, "Weightless Neural System Employing Simple Sensor Data for Efficient Real-Time Round-Corner, Junction and Doorway Detection for Autonomous System Path Planning in Smart Robotic Assisted Healthcare Wheelchairs," *2012 Third Int. Conf. Emerg. Secur. Technol.*, pp. 161–164, Sep. 2012.

[15] B. Mcelroy, M. Gillham, G. Howells, S. Spurgeon, S. Kelly, J. Batchelor, and M. Pepper, "Highly efficient Localisation utilising Weightless neural systems .," in *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2012, no. April, pp. 25–27.

[16] S. Nurmaini and B. Tutuko, "A New Classification Technique in Mobile Robot Navigation," *TELKOMNIKA,* vol. 9, no. 3, pp. 453–464, 2011.

[17] S. Nurmaini, "Modular Weightless Neural Network Architecture for Intelligent Navigation," *Int. J. Soft Comput. Appl.*, vol. 1, no. 1, pp. 1–18, 2009.

[18] S. Dini and M. Serrano, "Combining Q-Learning with Artificial Neural Networks in an Adaptive Light Seeking Robot," 2012.

[19] M. Tokic, "Teaching Reinforcement Learning using a Physical Robot," 2011.

[20] B. Bagnall, "Building a Light-seeking Robot with Q-learning," pp. 1–7, 2002.

[21] D. Claveau, "Progress towards a humanoid robot that learns to stand," *2012 IEEE Int. Conf. Dev. Learn. Epigenetic Robot.*, pp. 1–2, Nov. 2012.[22] S. M. Duggal and R. Chhabra, "Learning Systems and Their Applications : Future of Strategic Expert System," *Issues Inf. Syst.*, vol. 3, pp. 179–185, 2002.

[23] I. Aleksander, W. Thomas, and P. Bowden, "WISARD· a radical step forward in image recognition," *Sens. Rev.*, no. July, 1984.

[24] N. G. Haider and A. Karim, "RAM based neural-network controlled vehicle: path-tracking & collision avoidance," *2013 3rd IEEE Int. Conf. Comput. Control Commun.*, pp. 1–8, Sep. 2013.

[25] Q. Yao and D. Beetner, "A RAM-based neural network for collision avoidance in a mobile robot," in *International Joint Conference on Neural Networks*, 2003, no. figure 2, pp. 3157–3160.

[26] I. Wickert and F. França, "AUTOWISARD: Unsupervised Modes for the WISARD," *Connect. Model. Neurons, Learn. …*, 2001.

[27] H. Wicaksono, "Q learning behavior on autonomous navigation of physical robot," *Ubiquitous Robot. Ambient Intell. (…*, pp. 50–54, 2011.

[28] M. Agapie, "Simple Python robot simulator 2D download | SourceForge.net," 2013. [Online]. Available: http://sourceforge.net/projects/pyrobosim2d/. [Accessed: 14-Sep-2015].