International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-4, November 2019

Characterization of a Mobile Waste-Robot: A Heuristic Method to Path Planning using Artificial Neural Networks

Ralph Sherwin A. Corpuz

Abstract*: In the field of mobile robotics, path planning is one of the most widely-sought areas of interest due to its nature of complexity, where such issue is also practically evident in the case of mobile robots used for waste disposal purposes. To overcome issues on path planning, researchers have studied various classical and heuristic methods, however, the extent of optimization applicability and accuracy still remain an opportunity for further improvements. This paper presents the exploration of Artificial Neural Networks (ANN) in characterizing the path planning capability of a mobile waste-robot in order to improve navigational accuracy and path tracking time. The author utilized proximity and sound sensors as input vectors, dual H-bridge Direct Current (DC) motors as target vectors, and trained the ANN model using Levenberg-Marquardt (LM) and Scaled Conjugate (SCG) algorithms. Results revealed that LM was significantly more accurate than SCG algorithm in local path planning with Mean Square Error (MSE) values of 1.75966, 2.67946, and 2.04963, and Regression (R) values of 0.995671, 0.991247, and 0.983187 in training, testing, and validation environments, respectively. Furthermore, based on simulation results, LM was also found to be more accurate and faster than SCG with Pearson R correlation coefficients of r_x=.975, n_x=6, p_x=0.001 and r_y=.987, n_y=6, p_y=0.000 and path tracking time of 8.47s.

***Index Terms*: Path Planning, Mobile Waste Robotics, Artificial Neural Networks, Levenberg-Marquardt, Scaled Conjugate Gradient**

I. INTRODUCTION

One of the most challenging competencies in the design of a mobile robot is autonomous navigation, which is a process model used to reach a specific goal while avoiding obstacles. Shown on Fig. 1 is the autonomous navigation process composed of four fundamental sub-processes [1]. Initially, the robot must perceive its environment to extract meaningful data through the utilization of sensors. From these data, the robot should be able to navigate and locate its position in the environment. From then, the robot should be able to cognize and plan its path in order to achieve its goal. Lastly, it should be able to control and modulate its motions to achieve the desired trajectory while simultaneously avoiding obstacles [2]. Among these four sub-processes of autonomous navigation, path planning is considered to be the most common issue particularly in unstructured environment. Such issues are likewise practically true in the case of mobile robots used for waste disposal purposes. In path planning, a mobile robot aims to navigate from a start point to an end point, also known as target or goal, and maintains a collision-free trajectory. Some of the strategies sought toward successful path planning are not limited to shortest distance, smoothness of path, minimum energy consumption or a combination thereof such as shortest distance with minimal possible time, among others [3, 4].

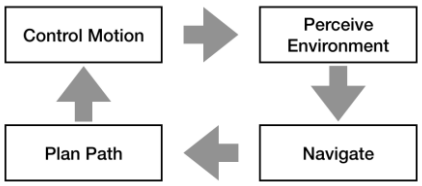


Fig. 1.Autonomous Navigation Process

To overcome issues associated with path planning, researchers explored various classical and heuristic methods. Some of the popular classical methods include cell decomposition, potential field, sampling-based, subgoal network and probabilistic roadmap. These classical methods are preferred for real-time motion planning applications because they are easy to implement, and modestly obtain good results [2, 3]. On the other hand, however, these classical methods do not produce optimal paths and are unstable in dynamic environments with multiple obstacles [2, 5]. In order to overcome the inefficiencies of classical methods, heuristic methods are employed for this purpose. Some of the popular heuristic methods used in mobile waste-robots include Artificial Neural Networks (ANN) [6], Fuzzy Logic Control (FLC) [7], or hybrid algorithms [8] etc. Of all these heuristic methods, ANNs are among the most recently-explored. ANN is a tool that acts like a human brain by learning and modelling complex relationships between input and output data [9] in order to achieve autonomous navigation. They are found to be efficient in terms of nonlinear mapping, learning ability and parallel processing, hence, they are highly recommended for heuristic methods of path planning [2].

Manuscript published on November 30, 2019.
* Correspondence Author
Ralph Sherwin A. Corpuz, Ph.D., Director of Quality Assurance and Assistant Professor, Electronics Engineering Technology, Technological University of the Philippines, Manila, Philippines.

On the other hand, however, ANNs are time-consuming, impractical for simple logic functions, and do not guarantee optimal solutions. With that said, the extent of optimization applicability and accuracy of ANNs still remain an opportunity for further study. The motivation of this paper focuses in characterizing the path planning capabilities of a mobile-waste robot in order to ensure navigational accuracy and path tracking time through the use of Artificial Neural Networks (ANN).

II. METHODOLOGY

A. Mobile Waste-Robot Prototype

The author adopted the waste-robot prototype developed by Corpuz and Orquiza, as shown on Fig. 2, which was intended to address human indifferences of throwing garbage in strategic locations such as home, schools, malls, etc. [7]. Originally designed using Fuzzy Logic Control (FLC), this prototype is capable of finding a sound clap goal produced by human while it avoids obstacles along its trajectory. The prototype was found to be responsive at 0.630 average seconds and 98.31% accurate in executing desired X and Y locations of the sound goal. Unfortunately, the prototype is limited of detecting its exact local frame and is also unstable in comparing sound waves threshold produced by human clap (as goal) with the environmental noise (stimuli). Hence, the optimization of navigational accuracy to reach the goal remains an opportunity for further improvement.

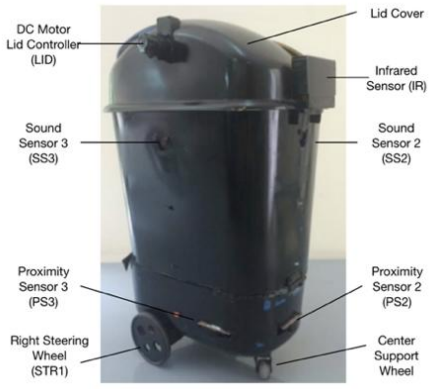


Fig. 2.Mobile Waste-Robot Prototype

For the context of this research, the author adopted only the proximity sensors (PS_1, PS_2, and PS_3) and sound sensors (SS_1, SS_2, and SS_3) as input vectors, and the dual H-bridge DC motors with steering wheels (STR) as target vector of the proposed ANN model. Conversely, the infrared sensor (IR) to detect human interference, the DC motor to control the lid cover (LID) and the Fuzzy Logic Control (FLC) System were not considered.

B. Kinematics Model

The kinematics model, as shown on Fig. 3, is based on a differential driver design [5]. The mobile-waste robot has a stiff body with two (2) steering wheels and one (1) center support wheel navigating in a horizontal plane. For the purpose of modelling, the effect of wheel axles, joints, and degrees of freedom internal to the mobile waste-robot were not included.

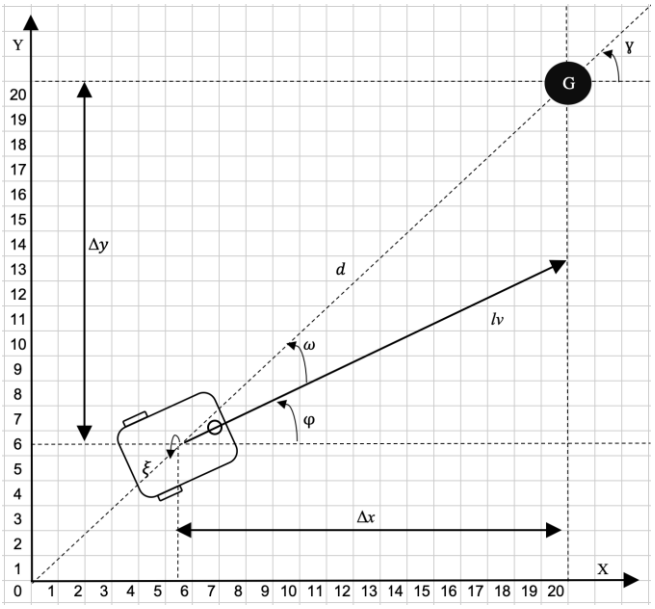


Fig. 3.Kinematics Model

The initial position of the mobile waste-robot is characterized by the inertial local frame $(x, y, \phi)^G$, as expressed further on the following equations:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{pmatrix} = \begin{pmatrix} \cos\phi & 0 \\ \sin\phi & 0 \\ 0 & \end{pmatrix} \begin{pmatrix} lv \\ \xi \end{pmatrix} \quad (1)$$

The control system of the mobile waste-robot includes the determination of linear velocity "lv" and angular velocity "ξ", which facilitate the navigation from initial position "x_0, y_0, ϕ_0" to the goal position "x_G, y_G, ϕ_G". Moreover, the control system computes the distance "d" and the angles "ω" and "γ" in order to determine the path, where $d = \sqrt{\Delta x^2 + \Delta y^2}$; $\omega = \arctan2(\Delta y, \Delta x)$; and $\gamma = -(\omega + \phi)$.

C. Path Planning Framework

Gleaned on Fig. 4 is the proposed path planning framework of the mobile-waste robot. The framework is composed of various modules working collaboratively in order to enhance navigational accuracy through path planning optimization. Initially, sensor data were analyzed through its proximity sensors (PS_1, PS_2, PS_3) to avoid obstacles and sound sensors (SS_1, SS_2, SS_3) to determine the location of the goal. From here, the reference trajectory (x_0, y_0, ϕ_0) were estimated vis-à-vis the sensor data obtained. These data were then transformed in order to determine the location of the sound goal (x_G, y_G, ϕ_G). Once the location of the sound goal was known, linear (lv), angular velocities (ξ), distance (d), the angles "ω" and "γ" were also estimated in order to guide the steering wheels (SL, SR, F, S) and pursue to the right directions. While the framework was assumed to work seamlessly, there were still various factors that could not be controlled, such as the electro-mechanical constraints of the wheels, and smoothness of trajectory.

Moreover, the angular position, distance and angular velocities of the mobile waste-robot were estimated through an odometer system such as wheel-encoder, hence, precise measurements of the location and steering angles were not guaranteed by the proposed framework. Such limitation could be attributed to inevitable stimuli that are internal or external to the mobile waste-robot under itself.

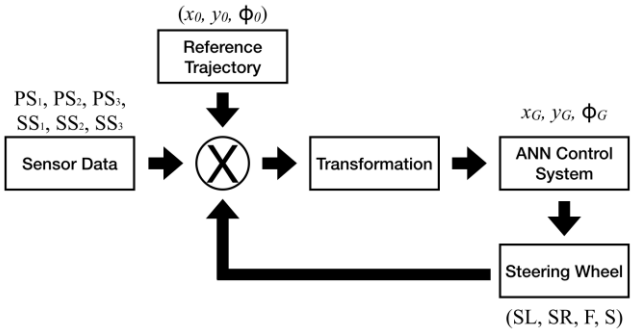


Fig. 4.Path Planning Framework

D. Input Vectors Characterization

The input vectors of the mobile waste-robot consisted the characterization of proximity and sound sensors. Shown on Fig. 5 is a sample proximity sensor module used as reference for obstacle avoidance. It has 2 IRDA sensors as transmitter and receiver circuits with 1 TTL active low logic output and is capable of detecting objects typically at 25 cm on its face. To ensure accuracy and responsiveness, the detection of all proximity sensors were set from 0.1 cm to 10 cm only, where in each range was set to have a corresponding effect to the STR. As detailed out on Table 1, if all proximity sensors do not detect an object (0cm), STR is set to stop. Meanwhile, if PS_I and PS_2 detect an object at 0.1 to 5 cm, STR shifts to the right while PS3 effects the STR to shift to the left for the same range. And lastly, if any of the proximity sensors detects an object at 5.1 to 10cm range, the STR moves forward.

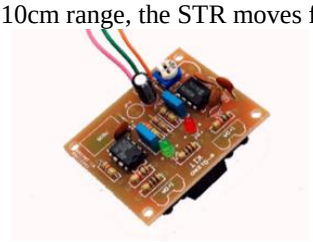


Fig. 5.Proximity Sensor Module

TABLE I. Proximity Sensors Characterization

PS_1	PS_2	PS_3	STR	Direction
0	0	0	0	Stop (S)
0.1-5	0	0	8	Shift Right (SR)
5.1-10	0	0	9	Forward (F)
0	0.1-5	0	8	Shift Right (SR)
0	5.1-10	0	9	Forward (F)
0	0	0.1-5	1	Shift Left (SL)
0	0	5.1-10	9	Forward (F)

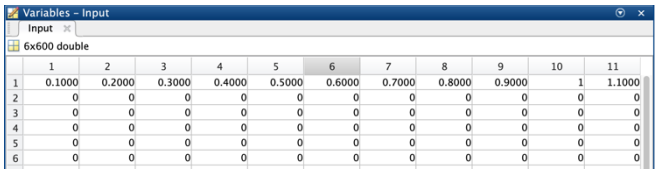
Depicted on Fig. 6 is the sound sensor module used to detect sounds through claps. It is composed of Electret Mic sensor and LM393 comparator IC, which has active low digital output. Summarized on Table 2 is the characterization of the sound sensors when claps are detected and their corresponding effects to the STR. Such that, when no claps are detected in all sensors, STR is set to stop. Meanwhile, with a detected sound range of 1-30dB, SS_1 causes the STR to shift left, while SS_2 causes the STR to move forward, and SS_3 causes the STR to shift right. Furthermore, with 31-59 dB sound range, SS_1 causes STR to shift left then move forward while SS_2 causes STR to move forward only and SS_3 causes the STR to shift right then move forward. Similarly, if any of the sensor detects 60-100dB, STR is set to face the goal then move forward.

Fig. 6.Sound Sensor Module

TABLE II. Sound Sensors Characterization

SS_1	SS_2	SS_3	STR	Direction
0	0	0	0	S
1-30	0	0	1	SL
31-59	0	0	10	SL, F
60-100	0	0	10	SL, F
0	1-30	0	0	S
0	31-59	0	9	F
0	60-100	0	9	F
0	0	1-30	8	SR
0	0	31-59	17	SR, F
0	0	60-100	17	SR, F

After the desired characteristics of the proximity and sensor modules were defined, their numerical values were convereted to 6x600 double input vectors using MATLAB. Shown on Fig. 7 is a screenshot of input vectors of the ANN model.



	1	2	3	4	5	6	7	8	9	10	11
1	0.1000	0.2000	0.3000	0.4000	0.5000	0.6000	0.7000	0.8000	0.9000	1	1.1000
2	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0

Fig. 7.6x600 Double Input Vectors

E. Target Vectors Characterization

A dual H-bridge geared DC motors driver module was used as reference in the characterization of the target vectors, as shown on Fig. 8. The module is capable of driving 2 geared DC motors with up to 16 Volts Direct Current (DC) and 1.4A current ratings.

Fig. 8.Dual H-Bridge Geared DC Motors Driver Module

Summarized further on Table 3 is the characterization of the target vectors (STR) based on the Binary-Coded-Decimal equivalents of the signal required to drive the module. Firstly, the BCD numbers were converted to decimal equivalent in order to be understood by the network for further modelling. A value of 0 means that the STR has to stop moving, 1 means to shift left, 8 means to shift right, 9 means to move forward, 10 is for shift left then forward, and 17 is for shift right then forward. The resulting pre-processed data, as shown on Fig. 9 includes 1x600 double target vectors used in the ANN model.

TABLE III. Dual H-Bridge Geared DC Motors Characterization

BCD	STR	Direction
0000	0	S
0001	1	SL
1000	8	SR
1001	9	F
0001, 1001	10	SL, F
1000, 1001	17	SR, F

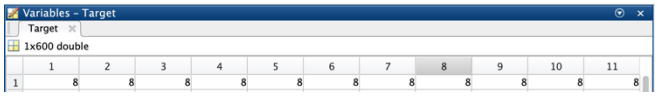


Fig. 9.1x600 Double Target Vectors

F. Artificial Neural Network (ANN) Model

Through MATLAB software, the author designed the control system of the mobile-waste robot using Artificial Neural Networks (ANN). Elaborated on Fig. 10 is the ANN model composed of 6 input layers, 2 hidden layers, and 1 output layer.

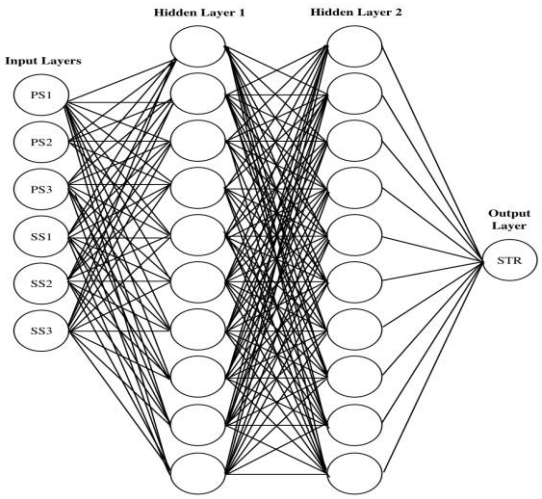


Fig. 10. Artificial Neural Network (ANN) Model

The input layers are composed of sensor data obtained from the proximity and sound sensors, collectively called input vectors. Each input vector is composed 600 elements while the 2 hidden layers have 10 neurons each. The first hidden and second layers are based on the weights and biases of initial and goal locations, and distance, angular and linear velocities of the mobile waste-robot respectively. The output layer, on the other hand, determines the appropriate directions of the steering wheels, in cosideration of the input and hidden layers. The input layers were designed using sigmoid transfer function while the output layer was designed using Softmax transfer function.

The ANN model was then trained in a supervised learning environment using heuristic-based algorithms. Intially, the author utilized the Levenberg-Marquardt (LM) algorithm, which is a fast backpropagation algorithm intended for nonlinear least square problems. Although LM requires more memory than other algorithms, it is found to be efficient in minimizing sum of square errors between data points and functions. LM utilizes gradient descent or Gauss-Newton methods when parameters are far or close to their optimum values, respectively [10, 11, 12]. The LM algorithm is further expressed in the following equation:

$$x_{k+1} = x_k - [J^T J + \mu I]^{-1} J^T e \quad (2)$$

Where "J" is the Jacobian matrix which has the first derivatives of the network errors in terms of weights and biases; "e" is a vector of network errors. When the scalar "μ" is 0, LM performs like the Newton's method where Hessian matrix is estimated as "$H = J^T J$". Meanwhile if the "μ" is large, LM becomes gradient descent computed as "$g=J^T e$" [13, 17].

Afterwards, the ANN model was then trained using Scaled Conjugate Gradient (SCG) algorithm. SCG is a fast, fully-automatic and robust optimization algorithm used in wide range of applications from prediction, classification to pattern recognition purposes etc. [14, 15]. SCG is found to be as fast as LM on function approximation but is faster than LM in cases with large datasets [13]. The SCG algorithm is computed using the following equation where "w_{ji}" are the weight values; "v" is number of x vectors; "l" is the learning coefficient' and "S" is the search direction [16]:

$$w_{ji}(v+1) = w_{ji}(v) + l\,S \quad (3)$$

In order to ensure internal validity of the ANN model, the author set the training, validation, and testing samples to 70%, 15%, and 15% respectively. Training was set to automatically stop when maximum number of iteration was reached; maximum amount of training time was exceeded; performance was minimized to the goal; performance gradient has fallen below minimum gradient value; μ has reached its maximum value (for LM); and if validation performance has increased more than the maximum failure threshold. Moreover, validation was conducted to ensure that the network generalization would stop training before overfitting while testing was done to ensure network generalization without affecting the training results.

The author then trained the ANN model in order to get the lowest possible errors. This was done by iterating the algorithms with at least 10 trials each using a computer with a 2.5 GHz Central Processing Unit (CPU) and 8GB, 1600 MHz DDR3 Random Access Memory (RAM) specifications.

The results were then evaluated using Mean Square Error (MSE) or the average squared difference between the target and output, and Regression R Values (R) or the measure of correlation between target and output. It was assumed that the lower the MSE, the lower the errors of the model. Meanwhile, the closer the value of R to 1, the better was the network performance.

The following equations (4) (5) were used to compute the MSE and Regression R values, respectively:

$$MSE\ (x,w) = \frac{1}{2}\sum_{p=1}^{P}\sum_{m=1}^{M} e^{2}_{p,n} \quad (4)$$

Where "x" is the network input; "w" is the weight of the network; "p" is the number of patterns; "m" is the number of outputs; "d" is the required output; "o" as the actual output; and "$e_{p,m}$" is the training error, which is equal to $d_{p,m} - o_{p,m}$.

$$y=\beta_0+\beta_1 x+e \quad (5)$$

Where "β_0" is the y-intercept; "β_1" is the regression coefficient; and "e" is the error.

Lastly, the resulting ANN model was further implemented using Pure Pursuit Algorithm and Binary Occupancy Grid in order to simulate the path planning capabilities of the mobile waste-robot. In here, the waypoints generated by LM and SCG algorithms were analyzed if there was a significant difference using Paired Samples T-test and as expressed further in the following formulas [7]:

$$H_0:\mu_1=\mu_2=\mu_3 \quad (6)$$

$$t = \frac{\bar{x}_{diff}-0}{S\bar{x}} \quad (7)$$

$$S\bar{x} = \frac{S_{diff}}{\sqrt{n}} \quad (8)$$

Where "H_0" is the null hypothesis; "μ_1" is the population mean of the reference trajectory; "μ_2" is the population mean of LM-generated waypoints; "μ_3" is the population mean of SCG-generated waypoints; "t" is the test statistic for Paired Samples T-test to test the significance difference of the three paired samples with 95% confidence level; "$\bar{x}diff$" is the sample mean of the differences; "n" is the sample size; "S_{diff}" is the sample standard deviation of the differences; and "$S\bar{x}$" is the estimated standard error of the mean $\frac{S}{\sqrt{n}}$.

Furthermore, to establish the significant relationship between the ideal waypoints and the actual simulation results, Pearson R Correlation Analysis was used for this purpose with the following formula:

$$r_{xy} = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum_i^2 x-(\sum x_i)^2}\sqrt{n\sum_i^2 y-(\sum y_i)^2}} \quad (9)$$

Where "r_{xy}" is the Pearson r correlation between x and y variables; "n" is the number of observations; "x_i" is the value of x for the nth observation and "y_i" is the value of y for the nth observation.

III. RESULTS AND DISCUSSION

The results of modelling and simulation using MATLAB are elaborated further on the following figures and tables, clustered in terms of ANN best performance results, progress results, error histogram, validation performance, regression plots, and the simulation results . Reflected on Table 4 are the best performance results of both LM and SCG algorithms generated by the ANN model. Out of 600 total samples analyzed, 420, 90, and 90 samples were used for training, testing, and validation purposes, respectively. With 10 trials conducted for each algorithm, the LM algorithm performed better than SCG where it generated a training MSE of 1.75966 and R value of 0.95671.

TABLE IV. Best ANN Model Performance Results

Algorithm	Mode	Sample	MSE	R
SCG	Training	420	2.59711	.929148
	Validation	90	2.35483	.951600
	Testing	90	2.94880	.926347
LM	Training	420	1.75966	.995671
	Validation	90	2.67946	.991247
	Testing	90	2.04963	.983187

Meanwhile, summarized on Table 5 is the progress results of the ANN model during validation. Interestingly, SCG trained faster (0 s) with lesser number of iterations (79) compared with LM (3 s, 145 iterations). On the other hand, however, LM still resulted to lower errors (0.174 MSE) and more minimized gradient values (0.247) than SCG (2.53 MSE, 2.27 gradient).

TABLE V. Validation Progress Results

Algorithm	Parameters	Values
SCG	Epoch Time	79 Iterations
	Time	0 Seconds
	Performance	2.53
	Gradient	2.27
	Validation Checks	6
LM	Epoch Time	145 Iterations
	Time	3 Seconds
	Performance	0.174
	Gradient	0.247
	Validation Checks	6

Furthermore, the best validation performance plots of SCG and LM are depicted in Fig. 11 and 12. As shown, LM generated a better validation performance than SCG with 0.26795 MSE but with more iterations recorded at 139. Conversely, SCG generated its best validation performance of 2.3548 MSE at epoch 73.

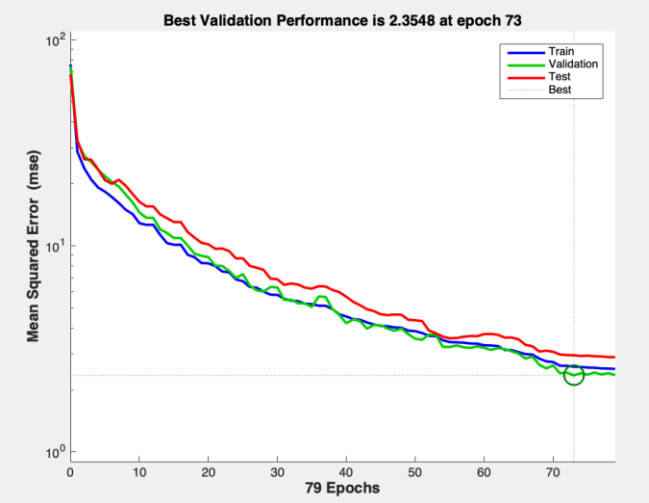


Fig. 11. SCG Best Validation Performance

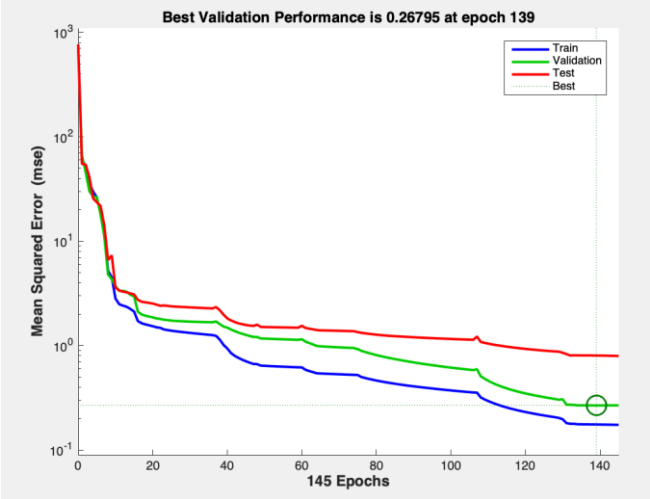


Fig. 12. LM Best Validation Performance

Illustrated in Figure 13 and 14 are the error histograms of SCG and LM respectively. With a total of 20 bins analyzed, LM shown to have lesser errors in all training, validation, and testing samples than SCG. This is manifested on the lesser instances of errors, which is measured by the difference of targets and outputs of the ANN model.

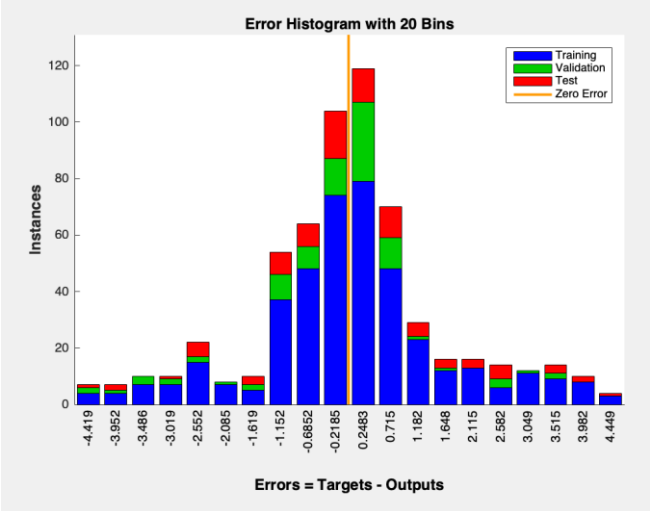


Fig. 13. SCG Error Histogram

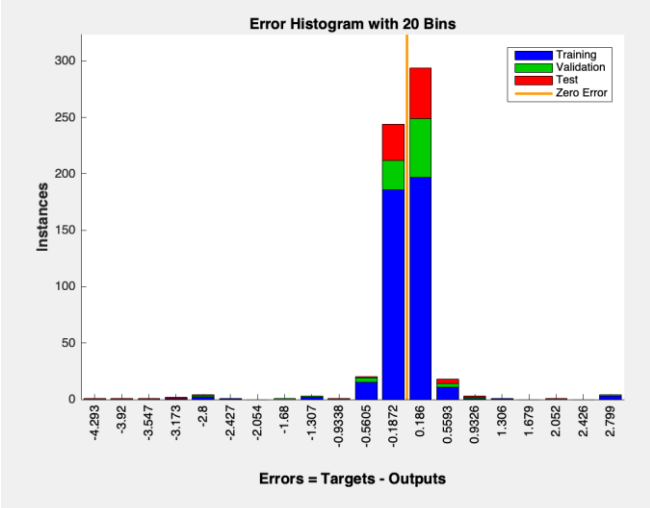


Fig. 14. LM Error Histogram

Reflected on Figure 15 and 16 are the regression plots of SCG and LM algorithms in training, validation, testing, and overall performance. Each regression plot interprets the correlation between target and output values per algorithm analyzed, wherein the closer the correlation or R value to 1, the better is the performance. As observed, LM algorithm generated closer relationship between the target and output vectors than SCG. This is manifested on the data plotted closely along the regression trend line. Also, LM generated higher R values in all training, testing, validation, and overall performance than SCG. This means that LM generated lesser errors in characterizing the path planning performancce of the mobile waste-robot under study.

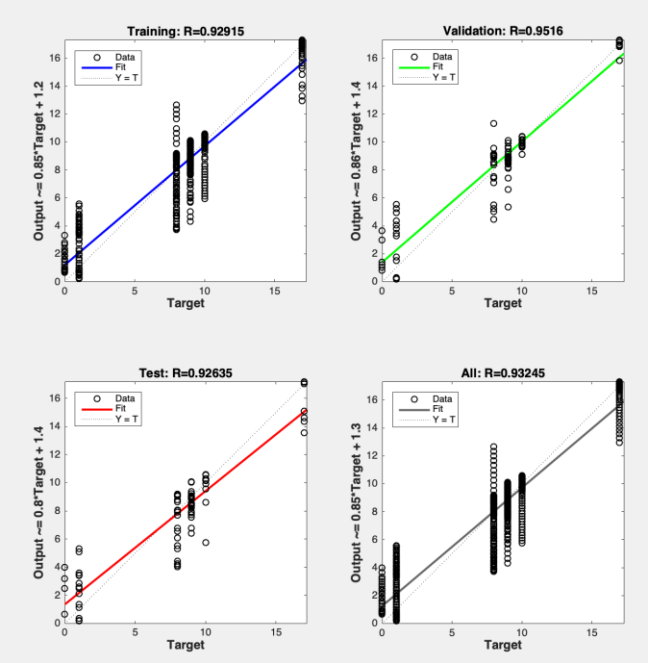


Fig. 15. SCG Regression Plot

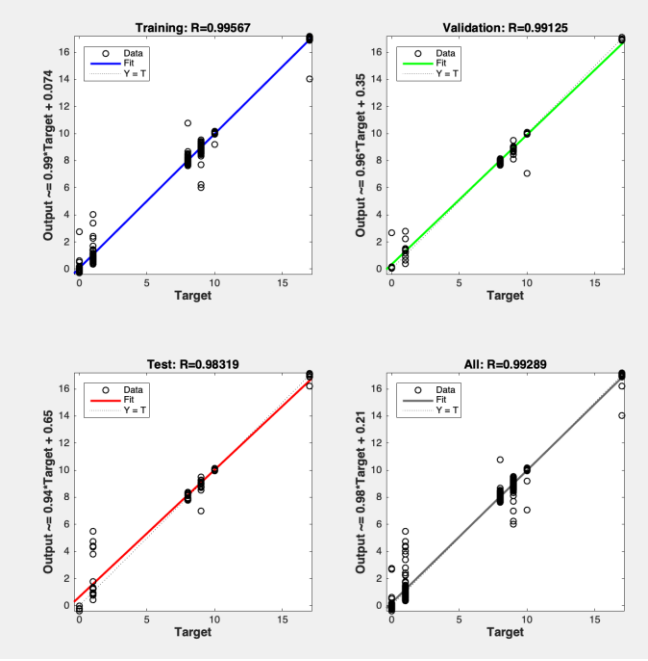


Fig. 16. LM Regression Plot

After the ANN modelling, the author then simulated the path planning performance of the mobile waste-robot. This was done by implementing the LM and SCG algorithms into executable path-tracking Pure Pursuit algorithm. The Pure Pursuit algorithm automatically computes the angular velocity command that moves the mobile waste-robot from its initial position to reach the goal position while the linear velocity is set as constant. This effects the mobile waste-robot to continuously chase the desired waypoints in front of it and avoid obstacles. Shown on Fig. 17 is a sample screenshot of the implemented Pure Pursuit algorithm while Fig. 18 depicts the graphical presentation of the initial pose used as reference of the simulation ($\Phi_0=0^o$).

```
Command Window
robotRadius = 0.2;
mapInflated = copy(map);
inflate(mapInflated,robotRadius);
prm = robotics.PRM(mapInflated);
prm.NumNodes = 100;
prm.ConnectionDistance = 10;
startLocation = [2.0 1.0];
endLocation = [12.0 10.0];
path = findpath(prm, startLocation, endLocation)
hold on
show(prm, 'Map', 'off', 'Roadmap', 'off');
hold off
release(controller)
controller.Waypoints = path;
robotCurrentLocation = path(1,:);
robotGoal = path(end,:);
initialOrientation = 0;
robotCurrentPose = [robotCurrentLocation initialOrientation];
robot = ExampleHelperDifferentialDriveRobot(robotCurrentPose);
distanceToGoal = norm(robotCurrentLocation - robotGoal);
goalRadius = 0.1;
while( distanceToGoal > goalRadius )

    % Compute the controller outputs, i.e., the inputs to the robo
    [lv, xi] = step(controller, robot.CurrentPose);

    % Simulate the robot using the controller outputs
    drive(robot, lv, xi)

    % Extract current location information from the current pose
    robotCurrentLocation = robot.CurrentPose(1:2);

    % Re-compute the distance to the goal
    distanceToGoal = norm(robotCurrentLocation - robotGoal);
end
```

Fig. 17. Sample Pure Pursuit Algorithm Screenshot

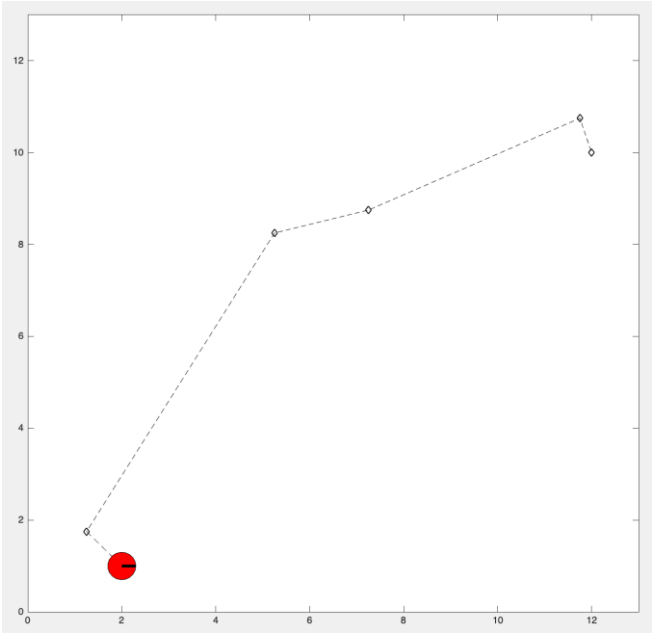


Fig. 18. PurePursuit Initial Pose Graphical Presentation

In order to validate the path planning performance of the ANN model, the author ran the Pure Pursuit algorithm in a Binary Occupancy Grid simulation environment with 26 x 27 grid size and 2-meter scale resolution. Listed on Table 6 are the 6 sets of way points where X_0 and Y_0 are reference coordinates, and X_{SCG} and Y_{SCG}, and X_{LM} and Y_{LM} are the resulting coordinates of the SCG and LM algorithms, respectively.

TABLE VI. Pure Pursuit Waypoints Results

Way point	X_0	Y_0	X_{SCG}	Y_{SCG}	X_{LM}	Y_{LM}
1	2.0000	1.0000	2.0000	1.0000	2.0000	1.0000
2	1.2500	1.7500	2.2328	1.0437	2.2956	1.2324
3	5.2500	8.2500	3.0516	5.5313	3.9882	6.3758
4	7.2500	8.7500	5.1576	7.7163	5.4899	8.2385
5	11.7500	10.7500	11.9864	9.7731	12.0417	9.9265
6	12.0000	10.0000	12.0000	10.0000	12.0000	10.0000

The following Fig. 19 and 20 illustrate the Pure Pursuit path generated by the SCG algorithm. As shown, SCG has followed 6 waypoints where path = [2.0000 1.0000; 2.232 1.0437; 3.0516 5.5313; 5.1576 7.7163; 11.9864 9.7731; 12.0000 10.0000], and the initial and goal positions were recorded at $\Phi_0=0^o$ and $\Phi_G=48^o$, respectively. The path was completed from initial to goal positions with total simulation time of 10.51s.

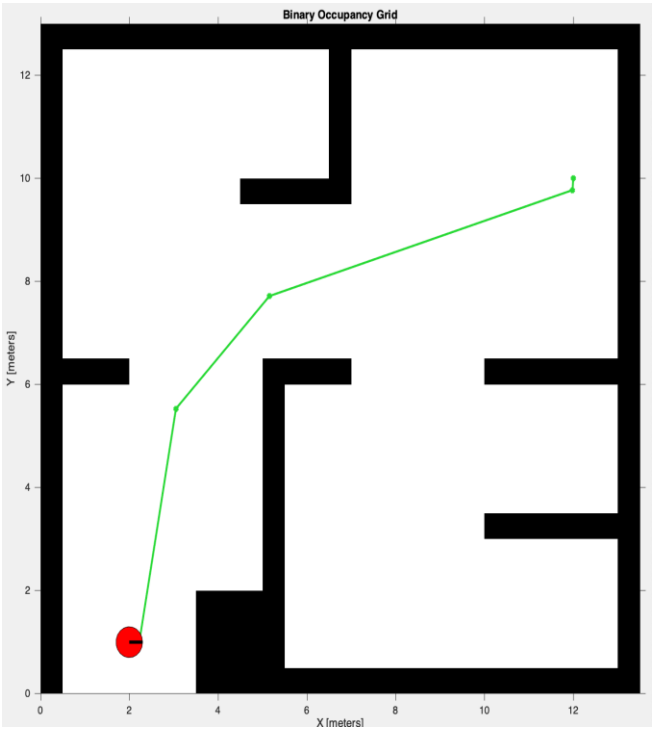


Fig. 19. SCG-Generated Pure Pursuit Path at Initial Position

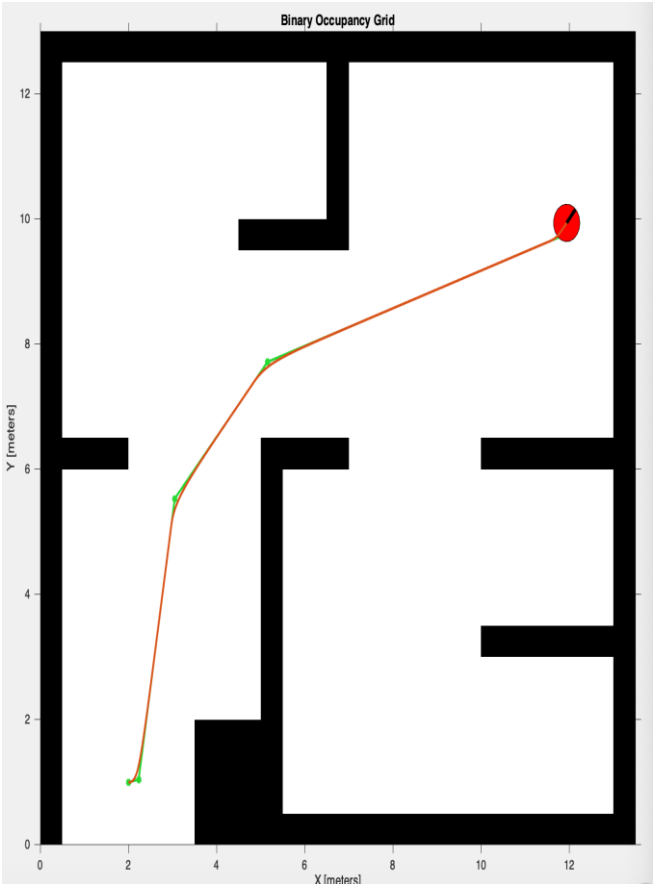


Fig. 20. SCG-Generated Pure Pursuit Path at Goal Position

Meanwhile, Fig. 21 and 22 elucidate the Pure Pursuit path generated by the LM algorithm. With 6 waypoints established, where path = [2.0000 1.0000; 2.2956 1.2324; 3.9882 6.3758; 5.4899 8.2385; 12.0417 9.9265; 12.0000 10.0000], the initial and goal positions were also determined at $\phi_0=0^o$ and $\phi_G=40^o$, respectively. The simulation was completed at 8.47s, which was faster than the simulation time of the SCG algorithm.

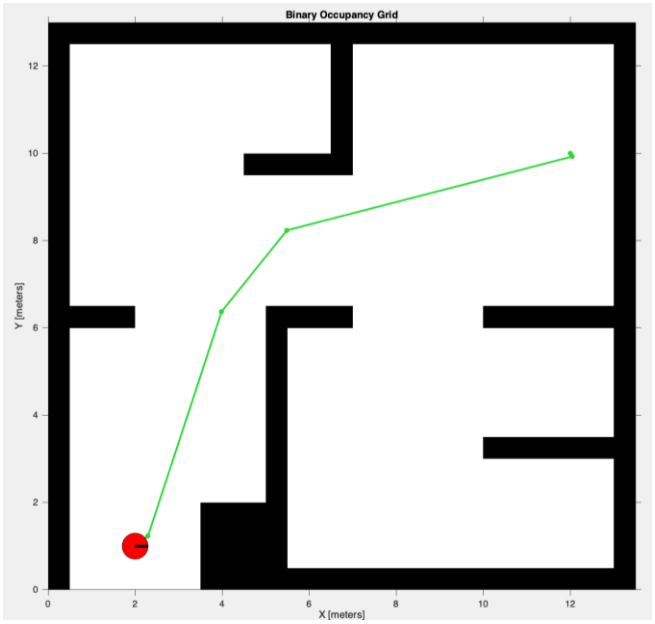


Fig. 21. LM-Generated Pure Pursuit Path at Initial Position

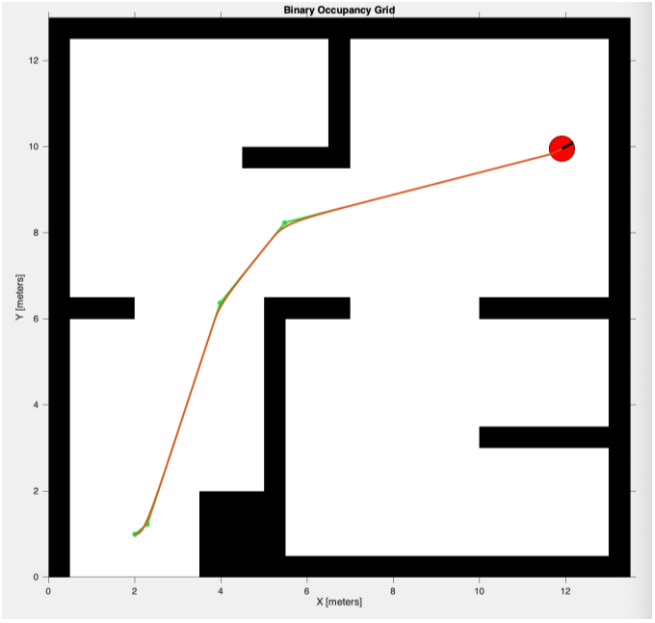


Fig. 22. LM-Generated Pure Pursuit Path at Goal Position

The author then compared the Pure Pursuit waypoints generated by both SCG and LM algorithms using Paired T-test. As shown on Table 7, results revealed that there is no significant difference between the waypoints tracked by both algorithms vis-à-vis the reference x and y coordinates. This conclusion is based on the result of T-test and probability values of t_1 (5) = 0.953, p_1=0.384; t_2 (5) = 2.222, p_2=0.077; t_3 (5) = 0.662, p_3=0.537; and t_4 (5) = 2.194, p_4=0.080.

TABLE VII. Paired T-Test Results

Pair	t	df	p (2-tailed)
X_0 and X_{SCG}	0.953	5	0.384
Y_0 and Y_{SCG}	2.222	5	0.077
X_0 and X_{LM}	0.662	5	0.537
Y_0 and Y_{LM}	2.194	5	0.080

Furthermore, the author analyzed the relationship between the simulated path with the reference trajectory using Pearson product-moment correlation (r). The results of analysis confirmed that the path tracked by LM algorithm is positively more correlated with the reference trajectory than the one generated by the SCG algorithm where r_x=.975, n_x=6, p_x=0.001 and r_y=.987, n_y=6, p_y=0.000. This means that the x and y waypoints generated by the LM are more accurately executed than the SCG as summarized on the following Table 8.

TABLE VIII. Pearson R Correlation Analysis Results

		X_0	Y_0	X_{SCG}	Y_{SCG}	X_{LM}	Y_{LM}
X_0	r	1	.916*	.961**	.977**	.975**	.959**
	p	-	.010	.002	.001	.001	.003
Y_0	r	.916*	1	.787	.972**	.827*	.987**
	p	.010	-	.063	.001	.042	.000
X_{SCG}	r	.961**	.787	1	.888*	.997**	.851*

	p	.002	.063	-	.018	.000	.032
Y_{SCG}	r	.977**	.972**	.888*	1	.913*	.997**
	p	.001	.001	.018	-	.011	.000
X_{LM}	r	.975**	.827*	.997**	.913*	1	.881*
	p	.001	.042	.000	.011	-	.020
Y_{LM}	r	.959**	.987**	.851*	.997**	.881*	1
	p	.003	.000	.032	.000	.020	-

Note: ** - correlation is significant at 0.01 level (2-tailed); * - correlation is significant at 0.01 level (1-tailed)

IV. CONCLUSIONS

In this paper, the author explored the feasibility of utilizing Artificial Neural Networks in characterizing the path planning capabilities of a Mobile Waste-Robot particularly with the main intent to determine improvements on navigational accuracy and path tracking time. Along with a pre-defined kinematics model, a path planning framework was proposed using an ANN as model of the control system. The ANN model was designed using 600 elements of proximity and sound sensors as input vectors, 2 hidden layers with 10 neurons each, and geared DC motors as output vector.

To enhance the path planning capability of the Mobile Waste-Robot, LM and SCG algorithms were used to train the ANN model with 420, 90, and 90 training, validation, and testing samples, respectively. After 10 trials conducted for each algorithm, results revealed that the LM algorithm performed better than SCG with lower average MSE of 1.75966. Likewise, LM resulted to lower validation errors (0.174 MSE) and more minimized gradient values (0.247) than SCG (2.53 MSE, 2.27 gradient). LM was also found to have higher Regression performance of 0.995671, 0.991247, and 0.983187 in training, validation, and testing environments. Conversely, SCG was found to have faster validation speed (0 s) with lesser number of iterations (79) compared with LM (3 s, 145 iterations).

The resulting algorithms generated by both SCG and LM were further implemented using Pure Pursuit algorithm and simulated in a 26x27 resolution, 2-m scaled Binary Occupancy Grid. While LM and SCG were found to be not statistically different from each other based on their capability to track the reference trajectory, however, LM performed better on navigational accuracy as manifested on the higher Pearson R correlation coefficient of r_x=.975, n_x=6, p_x=0.001 and r_y=.987, n_y=6, p_y=0.000 than those of the SCG's. Additionally, the LM completed the path tracking simulation faster than SCG at 8.47s.

REFERENCES

1. R. Siegwart and I.R. Nourbakhsh, Introduction to Autonomous Mobile Robot. Cambridge, MA: The Massachusetts Institute of Technology Press, 2004.
2. T. T. Mac, C. Copot, C., D.T. Tran, and R. De Keyser, "Heuristic approaches in robot path planning: A survey," Robotics and Autonomous Systems, vol. 86, pp. 13–28, 2016.
3. M. N. Zafar and J.C. Mohanta, "Methodology for Path Planning and Optimization of Mobile Robots: A Review," In Proc. International Conference on Robotics and Smart Manufacturing (RoSMa2018), 2018, pp. 141-152.
4. P. Krishna, and V. Kumar, "Convolutional Neural Network Based Path Navigation of a Differential Drive Robot in an Indoor Environment," International Journal of Recent Technology and Engineering, vol. 8, no. 2S3, 2019, pp. 547-549.
5. N. Ayadi, B. Maalej, and N. Derbel, "Optimal path planning of mobile robots: a comparative study," In Proc. 15th International Multi-Conference on Systems, Signals, and Devices (SSD), 2018, pp. 988-994
6. J. Bai, S. Lian, Z. Liu, K. Wang, and D. Liu, "Deep Learning Based Robot for Automatically Picking up Garbage on the Grass," IEEE Transactions on Consumer Electronics, vol. 64, no. 3, pp. 382-389, 2018.
7. R. S. Corpuz and J. C. Orquiza, "Utilization of Fuzzy Logic Control in a Waste Robot," In Proc. 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM 2018), 2018, Doi: 10.1109/HNICEM.2018.8666242
8. T.P. Tunggal, A. Supriyanto, R.N.M. Zaidatur, I. Faishal, I. Pambudi, and T. Iswanto, "Pursuit Algorithm for Robot Trash Can Based on Fuzzy-Cell Decomposition," International Journal of Electrical and Computer Engineering (IJECE), vol. 6, no. 6, pp. 2863-2869, 2016.
9. I. Valenzuela, et al., "Quality Assessment of Lettuce using Artificial Neural Network," In Proc. IEEE 9th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM 2017), 2017, Doi: 10.1109/HNICEM.2017.8269506
10. B. C. Aissa, and C. Fatima, "Neural Networks Trained with Levenberg-Marquardt-Iterated Extended Kalman Filter for Mobile Robot Trajectory Tracking," Journal of Engineering Science and Technology, vol. 10, no. 4, pp. 191-198, 2017.
11. Z.V.P. Murthy, "Nonlinear Regression: Levenberg-Marquardt Method," in Encyclopedia of Membranes, E. Dioli and L. Giorno, Eds. Berlin: Springer, 2014, pp. 1–3.
12. L. Duc-Hung, P. Cong-Kha, N.T.T. Trang, and B.T. Tu, "Parameter extraction and optimization using Levenberg-Marquardt algorithm," In Proc. 2012 Fourth International Conference on Communications and Electronics (ICCE), 2012, pp 434–437.
13. M. Beale, M.T. Hagan, and H.B Demuth, Neural network toolbox user's guide. Natick, MA: The MathWorks Inc., 2004.
14. R.S.A. Corpuz, "Implementation of artificial neural network using scaled conjugate gradient in ISO 9001:2015 audit findings classification," International Journal of Recent Technology and Engineering, vol. 8, no. 2, pp. 420-425, 2019.
15. A.Y. Quiñonez, M. Ramirez, C. Lizarraga, I. Tostado, and J. Bekios, "Autonomous Robot Navigation Based on Pattern Recognition Techniques and Artificial Neural Networks," Bioinspired Computation in Artificial Systems, J. Vicente, J. Sancgez, F. Lopez, F. Moreo, and H. Adeli, Eds. Berlin: Springer, 2015, pp. 320–329.
16. S. Aydin, I. Kilic, and H. Temeltas, "Using Linde Buzo Gray Clustering Neural Networks for Solving the Motion Equations of a Mobile Robot," Arabian Journal for Science and Engineering, vol. 36, no. 5, 2011, pp. 795-807.
17. M.T. Hagan, and M. Menhaj, "Training feed-forward networks with the Marquardt algorithm," IEEE Transactions on Neural Networks, vol. 5, no. 6, pp. 989–993, 1999.

AUTHOR'S PROFILE

Dr. Ralph Sherwin A. Corpuz obtained his degree of Doctor of Philosophy in Technology Management with specialization in Electronics Engineering Technology from the Technological University of the Philippines (TUP) – Manila. He currently serves as the Director of Quality Assurance and Assistant Professor of the Electronics Engineering Technology Department at TUP. His research interests include applied artificial intelligence, quality management systems, robotics and embedded systems.