Hindawi
Journal of Robotics
Volume 2021, Article ID 1129872, 14 pages
https://doi.org/10.1155/2021/1129872

Research Article

Mobile Robot Obstacle Avoidance Based on Neural Network with a Standardization Technique

Karoline Kamil A. Farag,[1,2] **Hussein Hamdy Shehata**,[3,4] **and Hesham M. El-Batsh**[2,5]

[1]*Mechatronics Engineering Department, Alexandria Higher Institute of Engineering and Technology (AIET), Alexandria, Egypt*
[2]*Mechanical Engineering, Faculty of Engineering, Benha University, Benha, Egypt*
[3]*Benha Faculty of Engineering, Benha University, Benha, Egypt*
[4]*Mechatronics Systems Engineering Department, Faculty of Engineering, MSA University, 6th of October City, Egypt*
[5]*High Institute of Engineering and Technology, Mahala El-Kobra, Egypt*

Correspondence should be addressed to Karoline Kamil A. Farag; caroline.kamil@aiet.edu.eg

Received 2 July 2021; Accepted 17 September 2021; Published 3 November 2021

Academic Editor: L. Fortuna

Reactive algorithm in an unknown environment is very useful to deal with dynamic obstacles that may change unexpectantly and quickly because the workspace is dynamic in real-life applications, and this work is focusing on the dynamic and unknown environment by online updating data in each step toward a specific goal; sensing and avoiding the obstacles coming across its way toward the target by training to take the corrective action for every possible offset is one of the most challenging problems in the field of robotics. This problem is solved by proposing an Artificial Intelligence System (AIS), which works on the behaviour of Intelligent Autonomous Vehicles (IAVs) like humans in recognition, learning, decision making, and action. First, the use of the AIS and some navigation methods based on Artificial Neural Networks (ANNs) to training datasets provided high Mean Square Error (MSE) from training on MATLAB Simulink tool. Standardization techniques were used to improve the performance of results from the training network on MATLAB Simulink. When it comes to knowledge-based systems, ANNs can be well adapted in an appropriate form. The adaption is related to the learning capacity since the network can consider and respond to new constraints and data related to the external environment.

1. Introduction

Navigation is an important challenge for autonomous mobile robots [1]. The problem is divided into positioning and path planning, so the major purpose of usage of the mobile robot is the shortest path from an initial point to the final point (target) in minimum time with high accuracy.

Real devices usually operate in systems that are far from ideal, and although defects cannot be avoided in real production processes, they still work. This is often associated with the fact that defects produce hidden dynamics, which, appropriately evoked, have an overall positive effect on the device [2]. We throughout specialize in elegant and incomplete electromechanical structures that can be thought of as a model of incomplete systems. Electrical and mechanical interactions within the structure generate complex patterns of vibration that can prevent the system from succeeding in the right working conditions. We discuss an impact strategy to ensure optimal working conditions support triggering the hidden dynamics of defects, characterizing their impact by referring to the characteristics of the control signals and the facility providing the structure.

A variety of articles on machine learning have been published over the past few years, including how it was implemented to help mobile robots develop their operating capabilities. Navigation is one of the most critical problems in designing and improving the intelligent mobile network. This is about a mobile robot's ability to plan and execute

collision motions within its environment. Robots need to be able to understand the environment's structure. The robots must be equipped with the potential for vision, data analysis, understanding, listening, logic, comprehension, decision, and response to reach their goals without collisions. The reproduction of this kind of intellect is, up to now, a human achievement in the creation and advancement of smart machines and, particularly, autonomous mobile robots. Sensing and understanding are two fundamental criteria to achieve a fair degree of autonomy. Obstacle avoidance is one of the most important problems for any realistic robotics design. There are several strategies in the literature to deal with this problem.

The main aim of the study is to solve the problem of motion planning in robotics. Supplied with an object with an initial location and direction, a goal location and orientation, and several obstacles placed in the workspace, the problem is finding a stable pathway from the initial location to the target location, which prevents clashes with obstacles along the way. In other words, path planning is classified into two subproblems, finding space and a stable pathway.

For more than half a century, robotics has been a feature of modern technology. Such devices are primarily being used for entertainment, security, and intelligence purposes as robots and their peripheral hardware grow more advanced, stable, and remotely controlled. Every robot remotely operated to produce images/videos for different purposes is designed to be a remotely controlled sensing robot. Robots that are remotely operated have significant recovery and security laws.

In this research, a mobile robot is trained by the neural network technique by entering all available paths for most probability for the environment that can expose in a path to reach a goal for the unknown environment and movable goal This state approximately reaches 235000 states using the ANOVA test to train the mobile robot to act for any change in the environment. Also, the standardization technique was used to improve training performance and regression rate to reach the goal. It is programmed by MATLAB connected with an Arduino. But, we take into consideration some criteria such as robot size, robot step size, and safety margin. To plan its motion not only with right positions but also with the reasonable robot's velocity and the relative position of the dynamic virtual obstacle [3].

The ANN uses training datasets that provided high Mean Square Error (MSE) from training on the MATLAB Simulink tool. Search for the reason for this big error found big difference values of datasets, so we make rescaling on the cost function of the hypothesis. Based on these characteristics and using standardization techniques to put all values of datasets in the range from −1 to +1, the performance of results from the training network on MATLAB Simulink is improved.

2. System Requirement, Assumption, and Precondition

This suggestion does not require the redefinition of the speed and position of the robot, the goal, and the obstacles, but during the execution of the action, it detects them. No more than exact online measurements can be achieved by using the IP camera finder and ultrasonic range. This idea assumes certain predictions as follows, to explain the analysis:

(i) The obstacles' size is very small with red colour, and we can accurately online measure their location.

(ii) Six Ultrasonic (US) sensors are used to cover 180° around the mobile robot; each of them is placed with a red colour sensor to identify only obstacles in the environment.

(iii) The IP camera is used to identify the target coordinate by training on Google Net or Alex Net. Experimental comparison with a traditional machine learning technique indicates a target that detects efficiency can be obtained by the proposed process.

(iv) The mobile robot can pass smoothly on the way to the goal; i.e., the steering angle is not restricted.

(v) The speed of the goal is equal to or slower than the speed of the robot movement.

3. Kinematics of the Mobile Robot

This section will show the kinematics of the mobile robot that has been taken into consideration when designed the mobile robot. The kinematic model of the mobile robot is shown in Figure 1. It consists of a vehicle frame mounted on the same axis with two driving wheels and a sliding shield at the front point. To achieve motion and orientation, the two driving wheels are independently driven by two motors [4, 5].

The strategy for the navigation of the mobile robot was proposed. The low-level inverse neural controller, which controls the mobile robot's dynamics, is the main component of the motion controller.

$$v_t = \frac{1}{2}(v_r + v_l),$$
$$\omega_t = \frac{1}{\omega}(v_r - v_l), \quad (1)$$
$$v_r = r\omega_r \text{ and } v_l = r\omega_l.$$

The position of the robot is defined by the vector [O, X, Y] notation as follows:

$$q = \begin{bmatrix} x_c & y_p & \theta \end{bmatrix}^T, \quad (2)$$

where

(i) X_c and Y_p are the coordinates of point P in the global coordinate

(ii) θ is the orientation of the local coordinate attached to the robot platform measured from the horizontal axis

(iii) $[PX_cY_p]$ are three generalized coordinates that can describe the configuration of the robot as in equation (2)

The rigid frame is the mobile robot framework considered here, and the wheels are pure rolling and without slippage. It says that the robot can only drive towards the axis

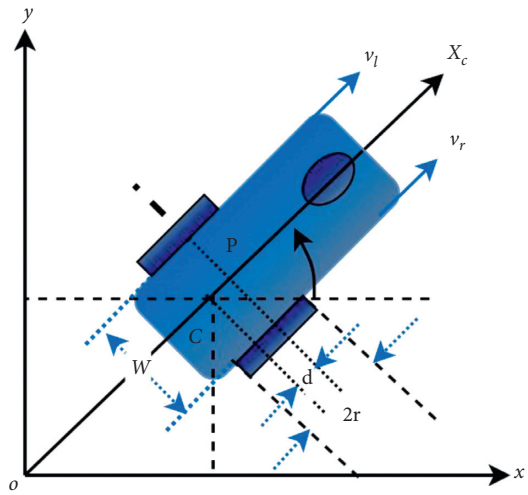


FIGURE 1: Kinematics model of the mobile robot. where: $2r$: the diameter of the two-wheel; W: the distance between two driving wheels; c: the mobile robot's centre of gravity (COG) which is centred at a point; d: the distance between points P and C; P: located at the intersection of a straight line passing through the centre of the vehicle and a line passing through the axis of the two wheels; V_r: linear right wheel velocity; V_l: linear left wheel velocity; V_t: linear tangential velocity; W_r: angular right wheel velocity; W_l: angular left wheel velocity; and W_t: angular tangential velocity.

of the driving wheels in the usual direction. The velocity part is zero at the contact point with the ground and orthogonal to the wheel's plane.

$$\left[\dot{y}_p \cos\theta - \dot{x}_c \sin\theta - d\dot{\theta}\right] = 0. \tag{3}$$

All kinematics constraints are independent of time and can be expressed as

$$A^T(q)\dot{q} = 0, \tag{4}$$

where $A(q)$ is the input transformation matrix associated with the constraints

$$C^T A(q) = 0, \tag{5}$$

where $C(q)$ is the full rank matrix formed by a set of independent vector fields spanning the null space of at (q).

From equations (4) and (5), it is possible to find an auxiliary vector time function $V(t)$ for all time t.

$$\dot{q} = C(q)V(t). \tag{6}$$

The constraint matrix in equation (4) for a mobile robot is given by

$$A^T(q) = \left[-\sin\theta \;\; \cos\theta \;\; -d\right]. \tag{7}$$

The $C(q)$ matrix is given by

$$C(q) = \begin{bmatrix} \cos\theta & -d\sin\theta \\ \sin\theta & d\cos\theta \\ 0 & 1 \end{bmatrix}. \tag{8}$$

Also,

$$V(t) = \left[v \;\; \omega\right]^t, \tag{9}$$

where v is the linear velocity of the point up along with the robot axis and ω is the angular velocity. Therefore, the kinematics equation in (6) can be described as

$$\dot{q} = \begin{bmatrix} \dot{x}_c \\ \dot{y}_p \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & -d\sin\theta \\ \sin\theta & d\cos\theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}. \tag{10}$$

Equation (10) is considered the vehicle's steering mechanism. The main aim desired reference trajectory that can control the track of the system. The control laws are designed to generate sufficient left and right wheel speeds to move the mobile robot to follow the necessary direction trajectories.

4. Neural Network Concepts

A selection of samples, connected with the control system, is constructed using the ANN. In a nonlinear way, these systems transform signals. Nonparametric estimation methods are neural networks that can construct discrete functions based on given case studies of inputs-outputs [5].

A three-layer classifier is a neural network constructed in this research. To train, the number of layers is set experimentally. There are eight neurons in the Input Layer (IL), six for detecting distance values from obstacles (i.e., in the front of the robot at 180°) and two for detecting the goal distance and angle. The inputs to the eighth neuron are set to "zero" if no target is detected and a single neuron in the Output Layer (OL), which produces the steering angle of the robot's motion direction with a linear activation function. Also, the Hidden Layer (HL) has 120 neurons with a Hyperbolic Tangent Sigmoid Activation Function (HTSAF).

$$f(x) = \frac{1}{1+e^{-x}}. \tag{11}$$

These hidden neuron numbers with their input and output signals are shown in Figure 2. The neural network is trained to navigate by describing approximately 235,000 cases representing.

The overall output can calculating by the following equations:

$$\text{sop} = \begin{bmatrix} i/p_1 & i/p_2 & i/p_3 & \dots & i/p_8 & 1 \end{bmatrix}_{1*9} \begin{bmatrix} w_{1,1} & w_{2,1} & \cdots & w_{120,1} \\ w_{1,2} & w_{2,2} & \cdots & w_{120,2} \\ w_{1,3} & w_{2,3} & \cdots & w_{120,3} \\ \vdots & \vdots & \cdots & \vdots \\ w_{1,8} & w_{2,8} & \cdots & w_{120,8} \\ B_{1,9} & B_{2,9} & \cdots & B_{120,9} \end{bmatrix}_{9*120}, \tag{12}$$

$$\text{sop} = \begin{bmatrix} \text{sop}_1 \\ \text{sop}_2 \\ \text{sop}_3 \\ \vdots \\ \text{sop}_{120} \end{bmatrix}_{1*120}, \tag{13}$$

where i: number of neurons in HL seriously, which change from 1 : 120, j: number of neurons in IL seriously, which change from 1 : 8, A: the only neuron in OL, sop_i is the sum of the product for each i neuron in HL, $w_{i,j}$: weights between i neuron in HL and j neuron in IL, and $B_{j,9}$: bias which is the additional weights for each neuron in HL.

$$o/p = \begin{bmatrix} o/p_1 \\ o/p_2 \\ o/p_3 \\ \vdots \\ o/p_{120} \end{bmatrix}_{1*120}, \tag{14}$$

where o/p_i is the output from each sop_i after using the HTSAF according to equation (11).

Finally, we calculate the overall output o/p_{overall} using the following equations:

$$o/p_{\text{overall}} = \begin{bmatrix} w_{A,1} & w_{A,2} & \dots & w_{A,120} & B_{A,121} \end{bmatrix}_{1*121} \begin{bmatrix} o/p_1 \\ o/p_2 \\ o/p_3 \\ \vdots \\ o/p_{120} \\ 1 \end{bmatrix}_{121*1}, \tag{15}$$

$$o/p_{\text{overall}} = [\text{output}]_{1*1}, \tag{16}$$

where $w_{A,i}$: weights between A neuron in OL and i neuron in HL.

5. Methodology and Network Architecture

This section shows the mobile robot construction and configuration for each component in the experimental datasets of a mobile robot and shows the desired ANN dataset and the training method to produce a robot controller [6, 7]. Figure 3 shows the designed workspace of the arranged environment used in this work.

The design of the grid front mobile robot is divided into 6 sectors shown in Table 1 concerning the angle.

It is divided into 3 ranges as shown in Table 2 concerning distance from the mobile robot.

5.1. Datasets

5.1.1. Inputs to Trained ANN. We have 8 inputs to the Arduino MEGA Controller divided into the following:

(a) Obstacle avoidance situation: 6 inputs indicate obstacles found in 180° from the workspace around the mobile robot in each step by the US connected consequently. Each of them covers 30° and 1 m distance as in [8].

(i) The designed ANN each sensor has 3 values shown in Table 3 indicating the obstacle's distance from the mobile robot.

(b) Goal position situation: 2 inputs from the IP camera which measure GOAL coordinate (x, y) to specify the following:

(i) In the designed ANN, the goal location has 3 values shown in Table 4 that indicate goal distance from the mobile robot which is located at the predefined ranges (Table 2).

$$\text{Goal location } r = \sqrt{x^2 + y^2}. \tag{17}$$

(ii) Also, the goal angle has 6 values shown in Table 5 indicating the goal sector which is located at the predefined sectors in Table 1.

$$\text{Goal angle } \theta = \tan^{-1}\frac{y}{x}. \tag{18}$$

5.1.2. Output from Training. This only one output from the trained ANN to specify the action movement angle of the mobile robot angle.

In the designed ANN, output comes from a rotating servo motor that has 6 values shown in Table 6 indicating the action of the mobile robot.

5.2. Description Example of the Possible Environmental Situation As Shown in Figure 4. An example of the input's situation on the real workspace is shown in Table 7 and its output to the servo motor of the mobile robot.

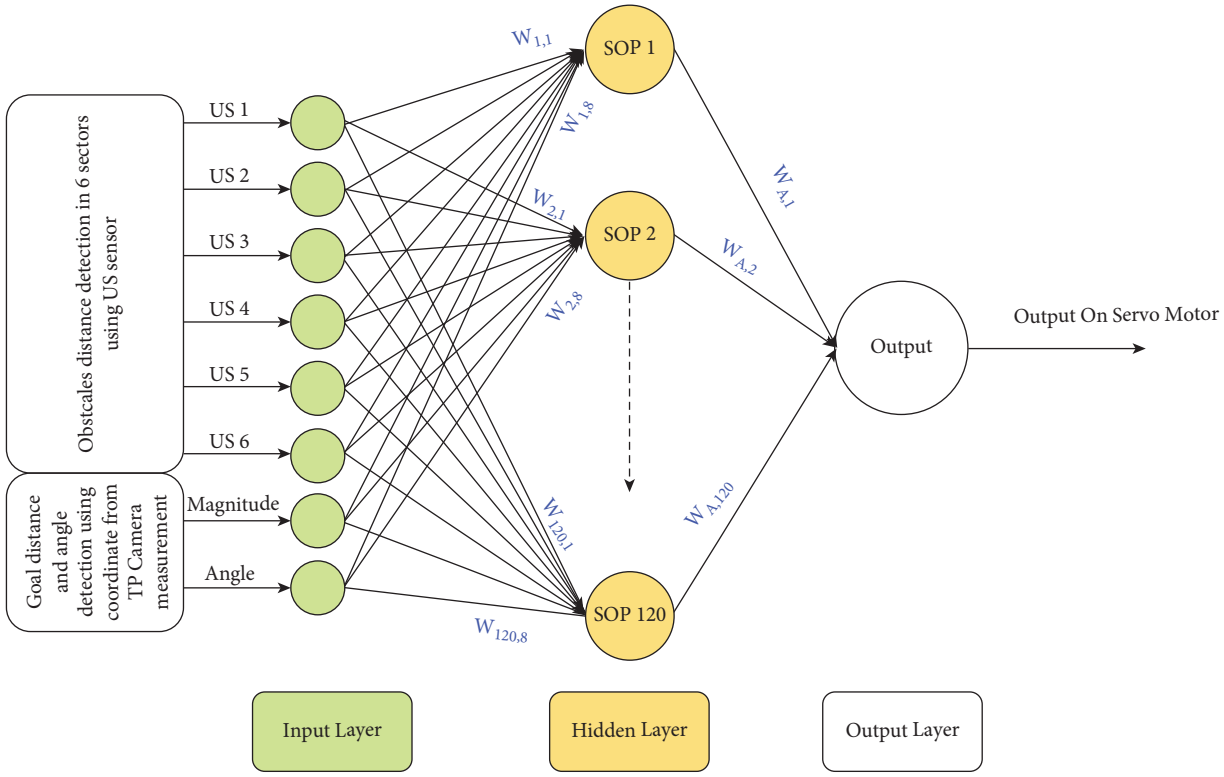


Figure 2: Three-layer neural network for robot navigation (neural network architecture).

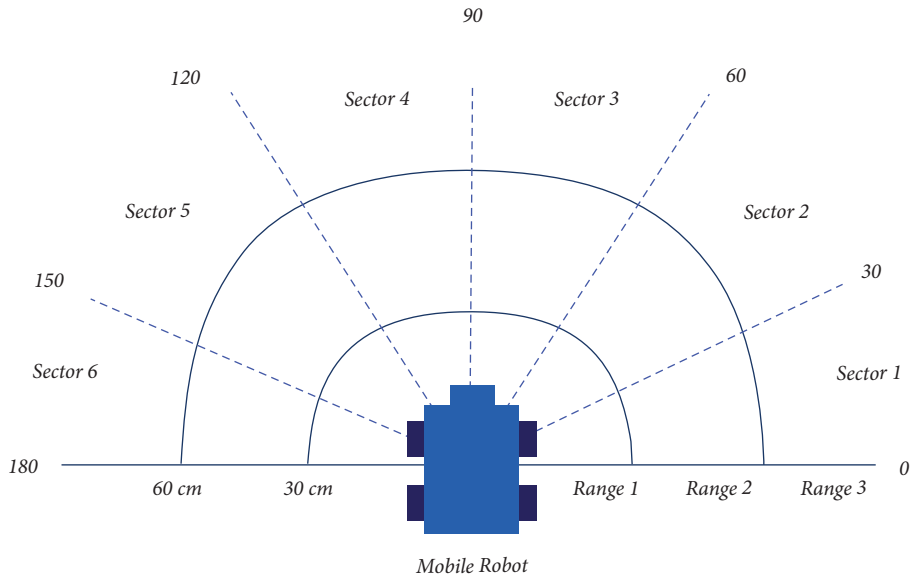


Figure 3: Workspace of the arranged environment's data.

6. ANN Learning

The mobile robot adapts to an environment without human interaction. We should provide it with the capacity to extract environmental information. It is, therefore, recommended to use a US range finder that scans the workspace with a 30° field of view and an operational range of 4 meters. The environment model is constantly modified after each scan, and the target is fixed. The US can measure the distance from the obstacle, and the IP camera measures the target coordinate.

In Figure 5, the protocol of our algorithm is described. If the US 6 is mounted on the robot front 180°, the following algorithm should be implemented.

Table 1: Division of forward workspace referring to the angles from the position of the mobile robot.

Sectors	Angle range
Sector 1	$0^\circ \le \theta \le 30^\circ$
Sector 2	$31^\circ \le \theta \le 60^\circ$
Sector 3	$61^\circ \le \theta \le 90^\circ$
Sector 4	$91^\circ \le \theta \le 120^\circ$
Sector 5	$121^\circ \le \theta \le 150^\circ$
Sector 6	$151^\circ \le \theta \le 180^\circ$

Table 2: Division of workspace referring to the distance from the position of the mobile robot.

Ranges	Distance from the mobile robot
Range 1	$2\,\text{cm} \le l \le 30\,\text{cm}$
Range 2	$31\,\text{cm} \le l \le 60\,\text{cm}$
Range 3	$l \ge 61\,\text{cm}$

Table 3: Sample input values from the US used in ANN to identify obstacles' location referring to the defined ranges in Table 2.

Values	Obstacles' distance from the mobile robot
1	Obstacles at range 1
2	Obstacles at range 2
0	Obstacles at range 3

Table 4: Sample input values from the IP camera used in ANN to identify the goal location with respect to the recent mobile robot location referring to the defined ranges in Table 2.

Magnitude value	Goal distance from the mobile robot
1	The goal at range 1
2	The goal at range 2
0	The goal at range 3

Table 5: Sample input values from the IP camera used in ANN to identify goal angle with respect to the recent mobile robot location referring to defined sectors in Table 1.

Angle value (°)	Goal sector
15	The goal at sector 1
45	The goal at sector 2
65	The goal at sector 3
105	The goal at sector 4
135	The goal at sector 5
165	The goal at sector 6

Table 6: Sample output values to the servo motor used in ANN to move the mobile robot by correction angle referring to the defined sectors in Table 1.

Output angle value (°)	Movement angle of the mobile robot at
15	Sector 1
45	Sector 2
65	Sector 3
105	Sector 4
135	Sector 5
165	Sector 6

7. Training Desired ANN and Improving Results Using Data Normalization and Standardization

The program exhibits cognition like that of a human being to solve problems, and there are two subfield of the AIS. There are many types of machine learning such as classification, regression, clustering, dimensional reduction analysis, and so on. The initial state of the backpropagation algorithm is always a point in the region of the origin of the coordinate space, while the search space scale greatly shifts the distance to the desired minimum. Therefore, requirements that minimize the search space to a unitary hyperbolic geometry compress the distance to a minimum, essentially speeding up the algorithm for backpropagation.

It is important to keep in mind the neural network. It typically initializes weights to random values in the range of −1 to 1 to explain why proper data normalization will increase neural network training speed and efficiency.

The system has implemented datasets trained many times using a backpropagation network tool on MATLAB. Various networks were developed and tested with random initial weights, activation functions, and a different number of epochs, and Table 8 shows the result from this training based on the given input and output datasets.

7.1. MATLAB Training Results for a Database Using Neural Network Toolboxes. Experimental results have a very high mean square error (MSE) and take more training time, as shown in Figure 6, so we will use the data normalization and standardization technique.

7.2. Data Normalization and Standardization Technique. It is found that if the features are on the same scale during gradient descent, then the algorithm appears to perform better than when the features in the same range are not properly scaled. The plot in Figure 7 shows the influence of feature scaling on the contour plot of the cost function of the hypothesis based on these characteristics.

As shown in Figure 7, learning steps will take longer to converge if the contours are biased as the steps will be more sensitive to oscillatory behaviour as seen in Figure 7. The plot is uniformly distributed if the characteristics are correctly scaled, and the steps of gradient descent have a stronger convergence profile.

By splitting the characteristics by max, the scaling of features between 0 and 1 is implemented. This helps to hold all the qualities within acceptable ranges. The goal is to keep the characteristics preferably within the range of −1 to 1.

The terms normalization and standardization are used interchangeably, but generally, they apply to different behaviours. Normalization usually involves scaling a number to a value between 0 and 1; thus, standardization translates information to a mean of zero and a normal deviation between −1 and 1 [9, 10], and with the following formula, each value on datasets can be standardized:

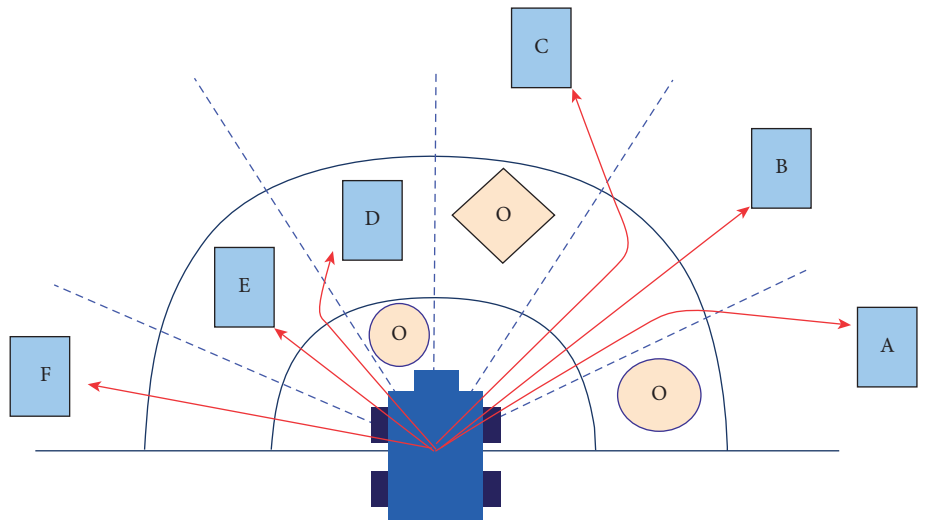


Figure 4: Example of case studies and their output according to the previous one. *O*: the position of obstacles in all states. *A*, *B*, *C*, *D*, *E*, *F*: the different positions of the goal.

Table 7: Situations on the real environment.

Different positions of goal	Inputs								Output to the servo motor (°)
	From the ultrasonic sensor						From the IP camera		
	US 1	US 2	US 3	US 4	US 5	US 6	Goal location	Goal angle (°)	
A	2	0	2	1	0	0	0	15	45
B	2	0	2	1	0	0	0	45	45
C	2	0	2	1	0	0	0	75	45
D	2	0	2	1	0	0	2	105	135
E	2	0	2	1	0	0	2	135	135
F	2	0	2	1	0	0	0	165	165

$$x_{\text{new}} = \frac{x_{\text{old}} - \mu}{\sigma}, \tag{19}$$

$$I = \left[\, i/p_1 - \mu_1 \;\; i/p_2 - \mu_2 \;\; \ldots \;\; i/p_8 - \mu_8 \,\right]_{1*8} * \text{Diag}\left[\frac{1}{\sigma_1} \;\; \frac{1}{\sigma_2} \;\; \ldots \;\; \frac{1}{\sigma_8}\right]_{8*8}, \tag{20}$$

where x_{new}: value after standardization; x_{old}: original values of the datasets; μ_j: mean value for each j neuron in IL; σ:standard deviation; and I: input after standard.

We use equation (20) in matrix form $1 * 8$ to get a new value for 8 inputs on datasets and substitute by using new values in equations (12), (13), (14), (15), and (16) to find overall output after standardization $o/p_{\text{overall.stand}}$.

To research the methods of normalization and data standardization to improve the backpropagation network, the model was applied. Using MATLAB, the simulations were carried out. Various networks with random initial weights were successfully tested. The network is trained ten times, and at various epochs, the efficiency objective is achieved. Taking the best of ten runs and calculating in terms of classification accuracy, the experimental effects can be analyzed. Table 9 indicates that when the standardization process is used, the accuracy has been increased.

7.3. MATLAB Training Results after Standardized Datasets Using Neural Network Toolboxes. The result of experimental data in Figure 8 showed a high improvement in the efficiency with the shortest training time of the neural networks' diabetes data classification model based on the methods of the data standardization technique.

8. Simulation and Experimental Software of Results

It would be ideal to test the performance of the proposed method on real-world data [11]. The dataset is the specific environment dataset, which contains various robot observations of several domestic environments in the form of arranged sensor and IP camera readings, which are collected from high numbers of situations in a different environment. The complete implementation of our algorithm was carried

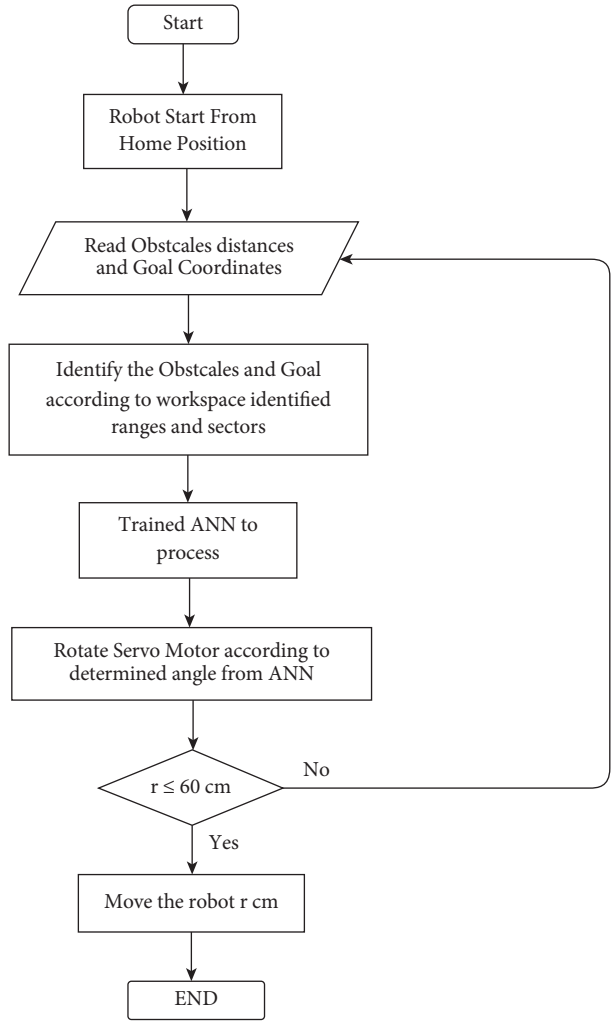


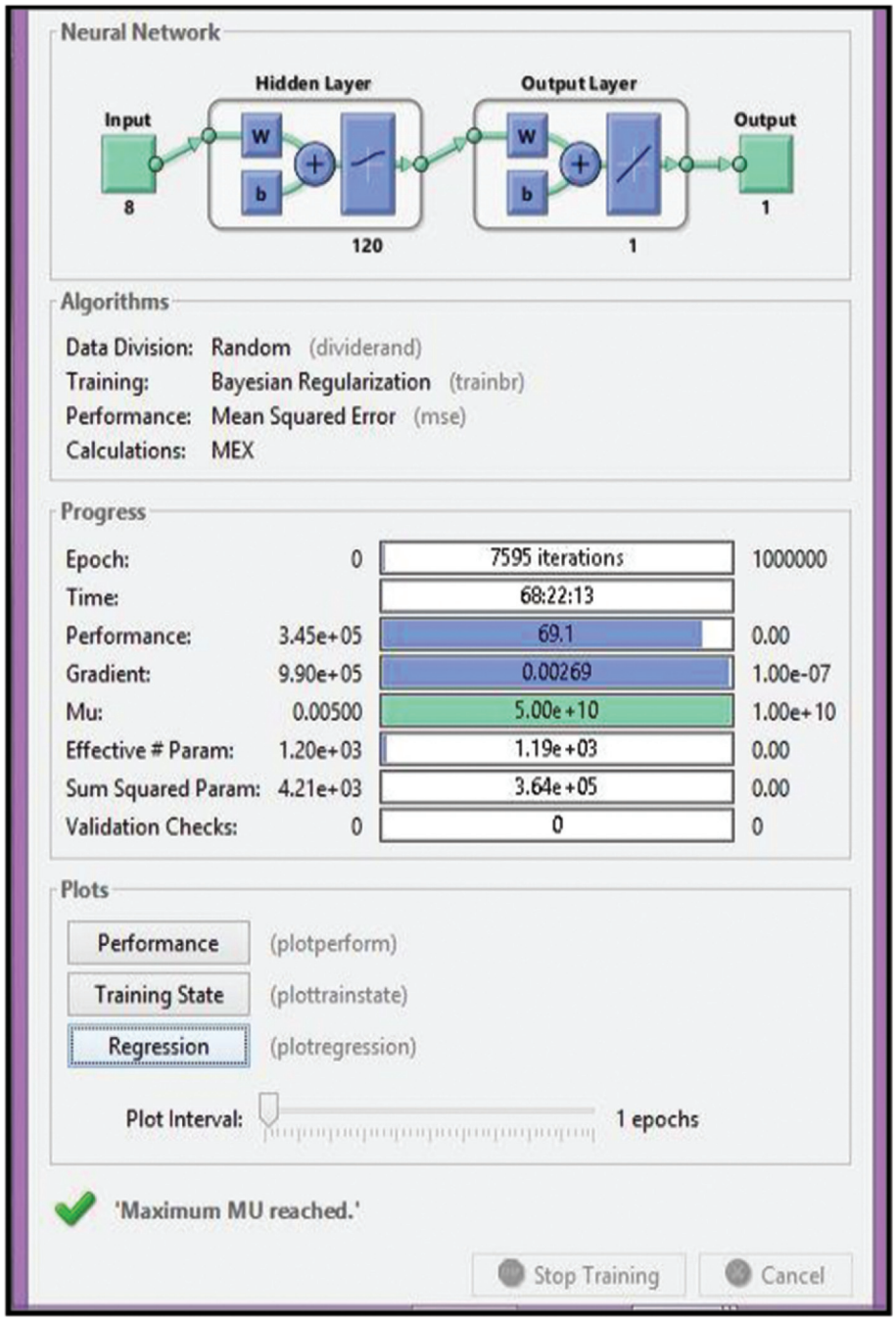
Figure 5: The flowchart of the proposed system.

Table 8: ANN training results using the backpropagation algorithm.

MATLAB—nntool				
Activation function	Hidden layer Output layer	Sigmoid **LOGSIG** Linear **PURELN**	Performance (MSE)	69.1268
Training algorithm		TRAINBR	Iteration	7594
Number of neurons		120	Regression	0.98664
Number of epochs		1000000	Training time	68 : 22 : 13

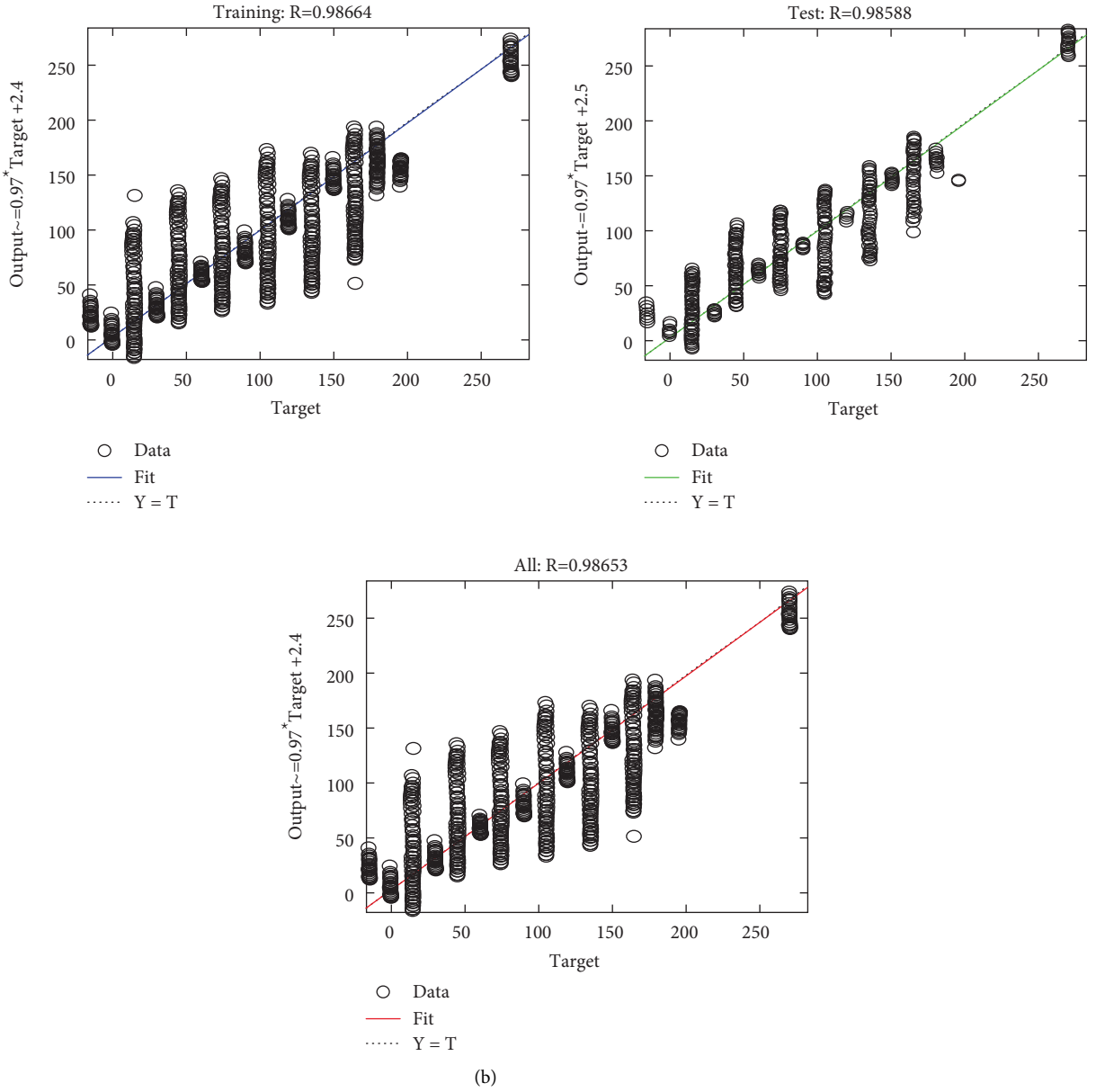
out in a MATLAB environment. By putting the robot and the goal in various environments and testing the robot motion behavior in different situations and demonstrated the good performance of the proposed method. Performance plots for training standardization and rescaling techniques are depicted in Figures 9(a) and 9(b), respectively. As can be seen from the figures, performance plots for training, validation, and test sets show similar characteristics, which means that the networks are very well trained and have a good generalization performance.

As we discussed earlier, the focus of the study is mainly on obstacle avoidance with low mean square errors and high regression. Thus, it would be better to predefine ranges and sectors on the environment around mobile robots using US sensors and IP cameras. To make the dataset applicable for system design, it is necessary to perform data preprocessing before going into the navigation simulation of the robot. After preprocessing, the sensor field of view is limited to 180° and the sensor range equals 4 m. The dimensions of the map are 500 × 500 cm. The obstacles can be static or dynamic, and the robot has no prior knowledge of the environment. The assumption control time step is 0.5 s, and the robot moves with a maximum speed of 0.5 m/s towards the target. As can be seen from the following simulation result, achieving the



(a)

Figure 6: Continued.



(b)

Figure 6: ANN characteristics output from backpropagation training.

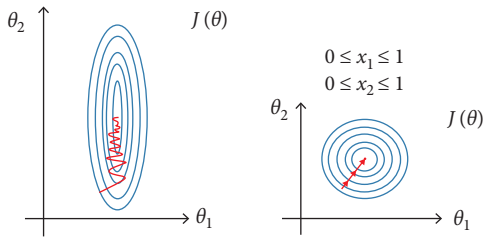
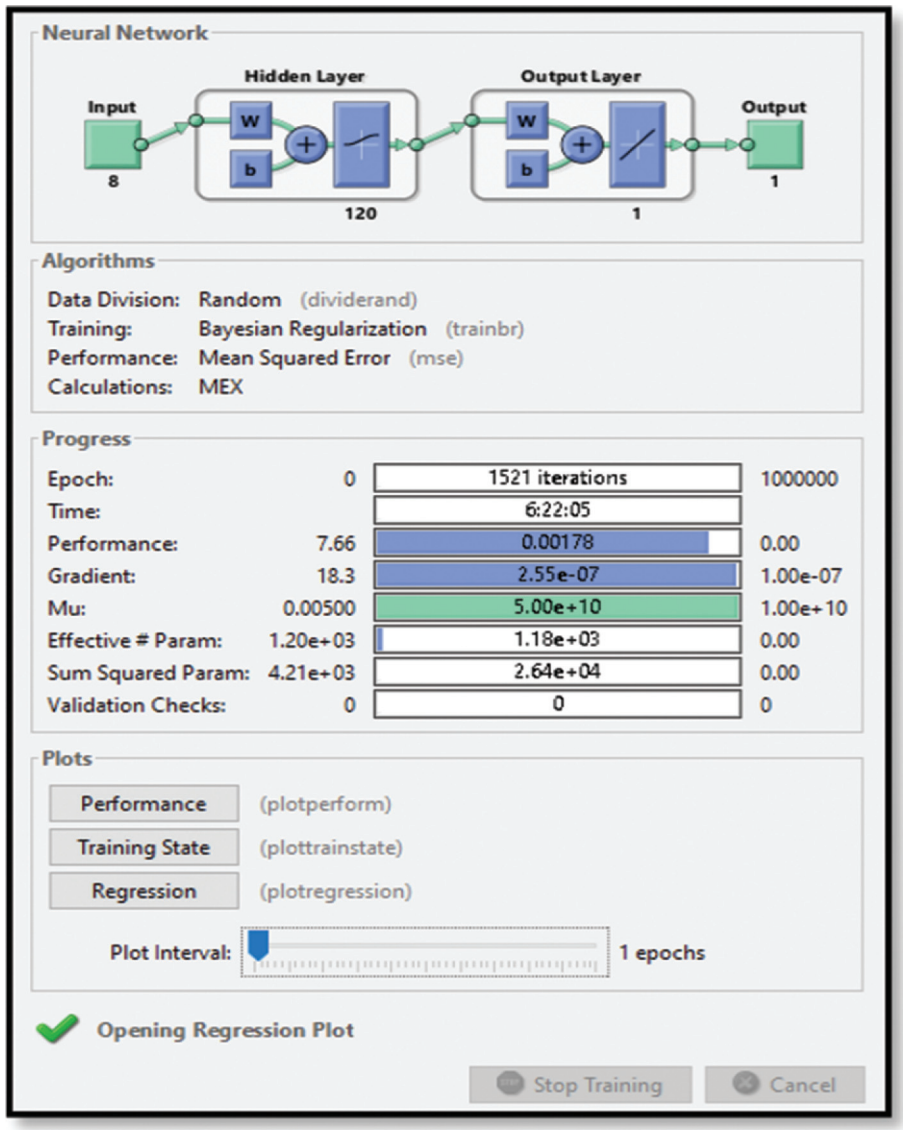


Figure 7: Feature scaling and contour plot.

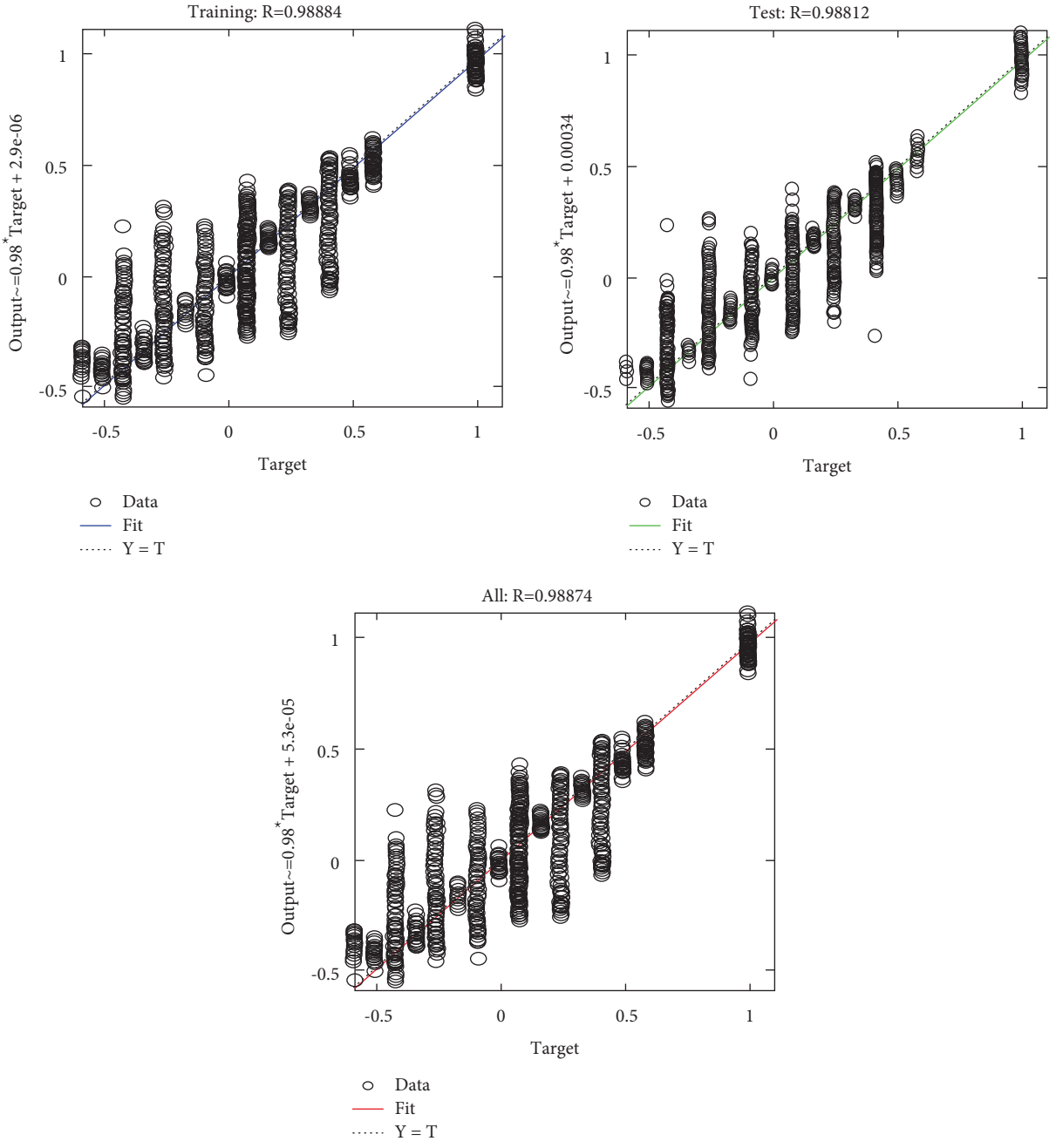
Table 9: Results from training datasets using MATLAB based on the methods of data standardization.

Training after stand.—nntool				
Activation function	Hidden layer Output layer	Sigmoid **LOGSIG** Linear **PURELN**	Performance (MSE)	0.00178
Training algorithm		TRAINBR	Iteration	1445
Number of neurons		120	Regression	0.98884
Number of epochs		1000000	Training time	6 : 22 : 05



(a)

Figure 8: Continued.



(b)

Figure 8: Results from training datasets using MATLAB based on the methods of data standardization.

steering angle leads to the generation of a smooth and safe path after standardization input and output datasets from the start point to the target position.

So, for the concept of the evaluation, d-infinite and largest Lyapunov exponent [12] were used. A nonlinear analysis method was used to review the complex dynamics of air bubbles carried by water and flowing through a microfluidic snake channel. The experimental observation of bubbles' motion shows an upscale sort of nonlinear dynamics and flow patterns. Schembri proposed a set of dimensionless parameters to classify the nonlinearity of the method showing also its sensitivity to input flow variations.

Also, this system discussed can be applied it for real environment "Dynamic and Unknown" such as transport shipments or heavyweights inside the factory to a specific location without attaching any object or offset, it can effect the industrial movement, and so on.

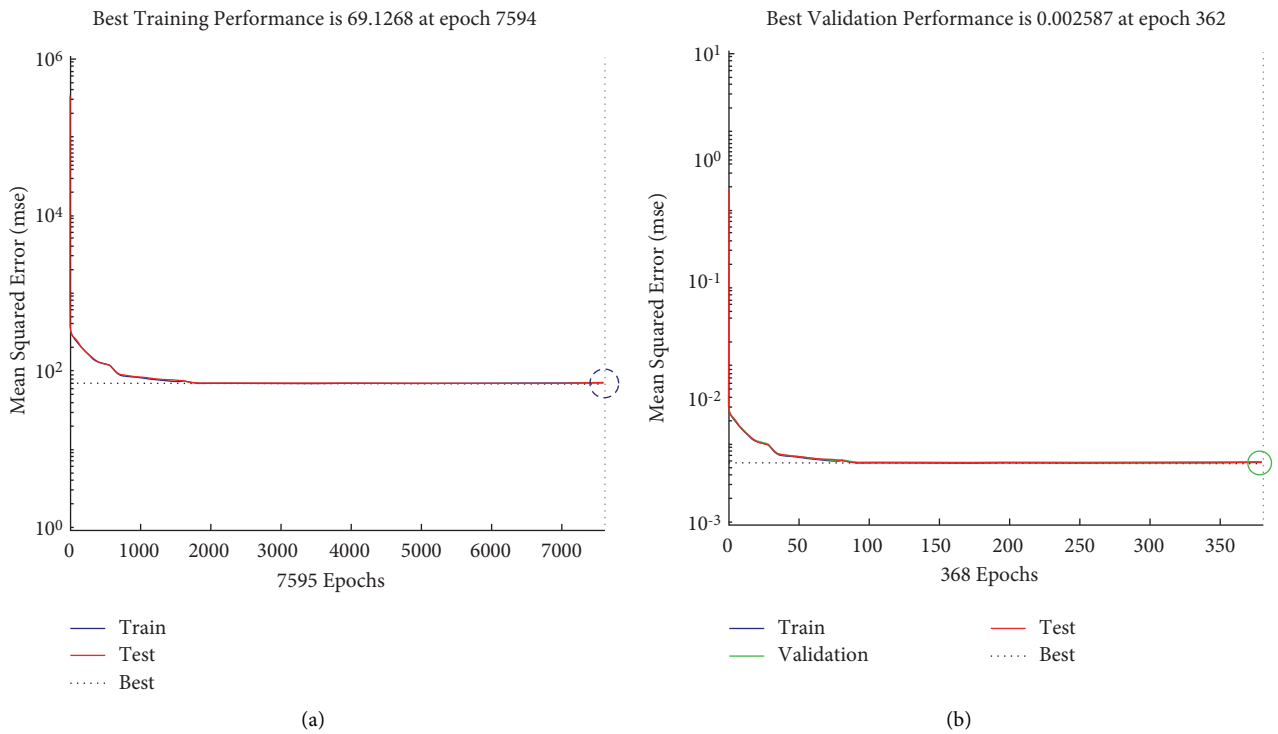


Figure 9: (a, b) High difference in training results of performance after using data standardization.

9. Conclusions and Future Work

This work illustrates the mobile robot obstacle avoidance technique using the ANN algorithm, but found big MSE when a big difference in values of trained datasets was seen. So, we used a data classification model based on the methods of the data standardization technique to rescaling all data and put all in specific range according to a standard formula which shows high improvement in the performance of the training results from (69.1268) to (0.00178) MSE with the shortest training time in two states with the same parameters, assumption, and activation function.

In future, this study will apply and prove the hardware experimentally with simulation in the real environment which is confirmed theoretically using MATLAB training in this paper.

Data Availability

The training dataset used to support this study's findings is included within the supplementary information files.

Conflicts of Interest

The authors declare no conflicts of interest.

Supplementary Materials

The supplementary files contain samples of the described datasets used in this work which contain case studies and their output, six inputs from Ultrasonic Sensors (USs) to determine the obstacles' distance, two inputs from the IP camera coordinate to determine the goal position "angle and distance," and one output to the servo motor to determine the mobile robot movement angle. (*Supplementary Materials*)

References

[1] A. Medina-Santiago, J. L. Camas-Anzueto, J. A. Vazquez-Feijoo, H. R. Hernández-de León, and R. Mota-Grajales, "Neural control system in obstacle avoidance in mobile robots using ultrasonic sensors," *Journal of Applied Research and Technology*, vol. 12, no. 1, pp. 104–110, 2014.

[2] M. Bucolo, A. Buscarino, C. Famoso, L. Fortuna, and M. Frasca, "Control of imperfect dynamical systems," *Nonlinear Dynamics*, vol. 98, no. 4, pp. 2989–2999, 2019.

[3] H. H. Shehata and J. Schlattmann, "Reactive algorithm for mobile robot path planning among moving target/obstacles by means of dynamic virtual obstacle concept,"in *proceedings of the 22nd International Conference on Flexible Automation and Intelligent Manufacturing (FAIM 2012)*, Helsinki, Finland, vol. 49, pp. 563–574, June 2012.

[4] M. K. Singh and D. R. Parhi, "Path optimisation of a mobile robot using an artificial neural network controller," *International Journal of Systems Science*, vol. 42, no. 1, pp. 107–120, 2011.

[5] D. R. Parhi and M. K. Singh, "Real-time navigational control of mobile robots using an artificial neural network," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 223, no. 7, pp. 1713–1725, 2009.

[6] D. Parhi and M. Singh, "Real-time navigational control of mobile robots using an artificial neural network," *Journal of Mechanical Engineering Science*, vol. 223, pp. 1713–1725, 2009.

[7] O. Azouaoui, Ouadah, Noureddine, Salem, Aouana, and D. Chabi, *Neural-Based Navigation Approach for a Bi-Steerable Mobile Robot*, Intech Open, London, UK, 2008.

[8] V. A. Zhmud, N. O. Kondratiev, K. A. Kuznetsov, V. G. Trubin, and L. V. Dimitrov, "Application of ultrasonic sensor for measuring distances in robotics," *Journal of Physics: Conference Series*, vol. 1015, 2018.

[9] T. Jayalakshmi and A. Santhakumaran, "Statistical normalization and back propagation for classification," *International Journal of Computer Theory and Engineering*, vol. 3, pp. 89–93, 2011.

[10] P. J. M. Ali and R. H. Faraj, "Data normalization and standardization: a technical report," *Machine Learning Technical Reports*, vol. 1, no. 1, pp. 1–6, 2014.

[11] F. Shamsfakhr and B. Sadeghibigham, "A neural network approach to navigation of a mobile robot and obstacle avoidance in dynamic and unknown environments," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 25, pp. 1629–1642, 2017.

[12] F. Schembri, F. Sapuppo, and M. Bucolo, "Experimental classification of nonlinear dynamics in microfluidic bubbles' flow," *Nonlinear Dynamics*, vol. 67, no. 4, pp. 2807–2819, 2012.