Trends in Applied Sciences Research

ISSN 1819-3579

Academic
Journals Inc.

www.academicjournals.com

Trends in Applied Sciences Research 9 (6): 312-318, 2014
ISSN 1819-3579 / DOI: 10.3923/tasr.2014.312.318

Underwater Mobile Robot Global Localization by using Feedforward Backpropagation Neural Network

[1,2]Mohammed Waadalla, [1]C.K. Yong, [1]David F.W. Yap and [1]R. Abd. Rahim
[1]Faculty of Electronics and Computer Engineering, Universiti Teknikal Malaysia Melaka (UTeM), Hang Tuah Jaya, Durian Tunggal, Melaka, 76109, Malaysia
[2]University of Mosul, Al-Majmoa'a Street, Mosul, Iraq

Corresponding Author: Mohammed Waadalla, Faculty of Electronics and Computer Engineering, Universiti Teknikal Malaysia Melaka (UTeM), Hang Tuah Jaya, Durian Tunggal, Melaka, 76109, Malaysia

ABSTRACT

Underwater global localization is an essential tool for underwater researchers. In this study global gocalization for underwater mobile robot has been developed using Feedforward Backpropagation Neural Network (FBNN). Twelve sonar sensors have been recorded with the x and y location of the robot using MobotSim software. There are a total of 58081 points and 12 sonars that are used to record each point. These recordings have been used for supervised training by using MATLAB software. The results are determined by using four random points to calculate the location of the robot from the sonar sensor readings. The proposed method that is used in calculating x and y points has accuracy equal to 0.01 m. The result shows that in 10 layers network, the 0.000511 absolute error value with percentage error of 0.035% in x point and the 0.0028893 absolute error value with percentage error of 0.13% in y point are achieved. While, in 12 layers network, the 4.43×10^{-05} absolute error value with percentage error of 0.003% in x point and the 0.0001767 absolute error value with percentage error of 0.008% in y point are achieved. This study illustrates that feedforward backpropagation neural network can be used to determine the location of the robot with marginal percentage error. Moreover, the resulted percentage error is internationally accepted by electronic engineers.

Key words: Feedforward backpropagation neural network, MobotSim, underwater

INTRODUCTION

Robotic technology became more popular in recent years where the robots are used in industrial factories, home appliance and in other life fields (Tamimi *et al*., 2010). However one of the difficulties which the robot designer will face it is to enable the robot to determine its current location. One of the famous methods to find the position of the robot is global localization method (Liu *et al*., 2007). For indoor environment a modified Monte Carlo Localization (MCL) proved to save computational resources and enhances localization efficiency (Fang *et al*., 2011) while the navigation and robot localization algorithm method (Song and Liu, 2013) was used to trail and position robots in real-time to make the formulation of navigation and localization more convenient and reliable. While these two methods work conveniently for indoor environment, an underwater localization certainly is a more challenging task. In this study Feedforward Backpropagation

Neural Network (FBNN) was applied to determine the position of the underwater robot such as in swimming pools, industrial tanks, oil tanks and in any underwater environment (Yeo *et al*., 2011). In this technique, neural network was created and trained using the data recorded from the simulation of the robot locations using MobotSim software (Se *et al*., 2005). Once the training of the network is done, the robot can determine its current location in the underwater environment according to the readings of the sonar sensors (Kim and Kim, 2009). However, the depth of the underwater robot is omitted in the simulation study.

Artificial Neural Network (ANN) is considered as the main computational method used in artificial intelligence. It is widely used in non-linear system. Artificial neural network consists of input layer, hidden layer and output layer (Kim and Kim, 2011). Each layer has multiple neurons. The neurons are connected together using links. Each link has a specific number of weights where the information is saved. Before using the neural network system, supervised training was carried out using actual sonar sensor readings (input data) and actual x position and actual y position (target data) (Moreno *et al*., 2007). In this study, back propagation training method was used to train the neural network in order to find the position of the robot.

Underwater mobile robot is designed to perform tasks under the water where human has difficulty in diving. Underwater robot can be used in many fields such as in oil and gas industry in order to detail the seafloor map before they build well stations, in military application where it is used to locate and determine mines location and in research fields to study the lakes and ocean floors (Xie *et al*., 2010). In the next subsections the neural network will be explained in detail on the methods which were used in this study. The results will be shown and discussed in the results and discussion subsection.

Neural network: Neural network is based on neurons with each of them connected to other neurons through weighted links. The output value of each neuron can be found by means of a mathematical equation. The weight of the weighted link will adapt according to the network training. Figure 1 illustrate the single neuron with n weighted links (n inputs) and all of the mathematical function related to it (Janet *et al*., 1996).

In this study the basic design of a neuron will be used as in Fig. 1. However the inputs of each neuron consist of 12 links depending on the 12 sonar sensors in the robot. The transfer function which will be used in this neuron is tangent sigmoid transfer function (TANSIG) shown Fig. 2, while the equations are given as Eq. 1 and 2:

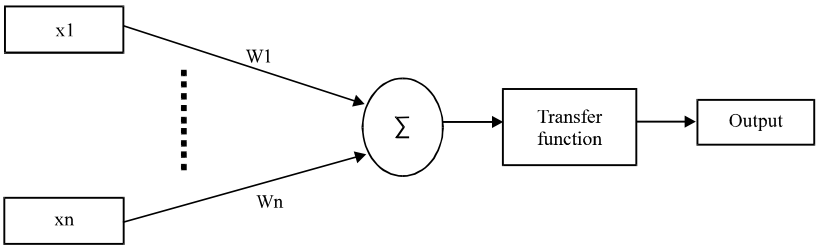

Fig. 1: Illustration of a neuron

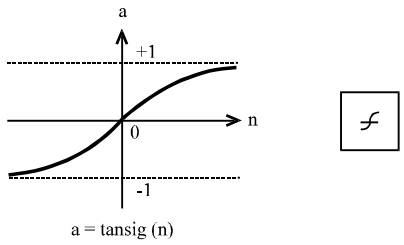

Fig. 2: Tan-sigmoid transfer function

$$\text{Tansig}(z) = \tanh(z) = \frac{e^{z} - e^{-2}}{e^{z} + e^{-2}} \tag{1}$$

$$\text{Tansig}(z) = \frac{2}{1 + e^{-2Z}} - 1 \tag{2}$$

The neural network training function which was used in this study is 'trainlm' function in Matlab. This training function would adapt the weight and bias of the neuron by means of Levenberg-Marquardt optimization. The weight and biasing learning function 'learngdm' was also used here. The performance of the network and the end of training will depend on the mean square error function (mse).

METHODOLOGY

The data which is needed is collected from the simulation of the robot movement in 3×3 m environment. The robot is equipped with 12 sonar sensors to measure the distance and each sensor can measure up to 2 m. If the distance of the particular sonar is more than 2 m then it will be recorded as 2.1 m. The simulation was done using MobotSim software. Figure 3 shows the simulation environment surface where each black point is recorded to use as the learning sample data sets.

Figure 3 illustrates the simulation of the surface environment. It can be observed that the distance between each two recorded position is 0.01 m towards x axes and 0.01 m towards y axes, while the dimension of the surface is 3×3 m. The recorded points should be at least 0.3 m from the edge of the surface. Figure 4 shows the snapshot of the simulation software, MobotSim.

After creating the simulation of the surface environment, the MobotSim software would start to move the robot to every point and record the sonar readings at that particular point and record it as a learning sample for the proposed neural network. After all of the needed data has been recorded and saved, a MATLAB m-file would be created with all of the recorded data before it is used to train the neural network. Table 1 shows the sample of the recorded data. Each column represents one sample point. The rows from 1 to 12 represent the distance in meter readied by sonar sensor. The last two rows represent the actual position of the robot with relation to the top left origin point. The minimum value of x and y actual point is 1.3 m, while the maximum value of x and y actual point is 3.7 m, relative to the limit of the sonar sensor capability which is from 1-4 m. Overall, the total number of test points is 58081.

In this study two types of neural network was created. The first consists of 10 hidden layers and the second contains 12 hidden layers. The creation of the neural network was done by using the

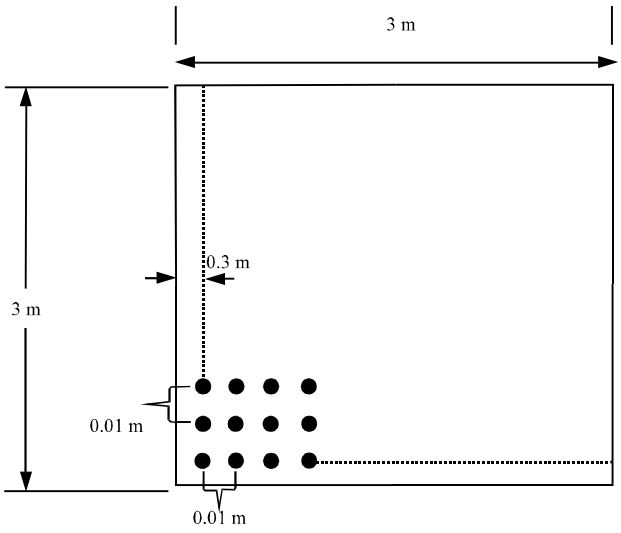

Fig. 3: Surface environment simulation

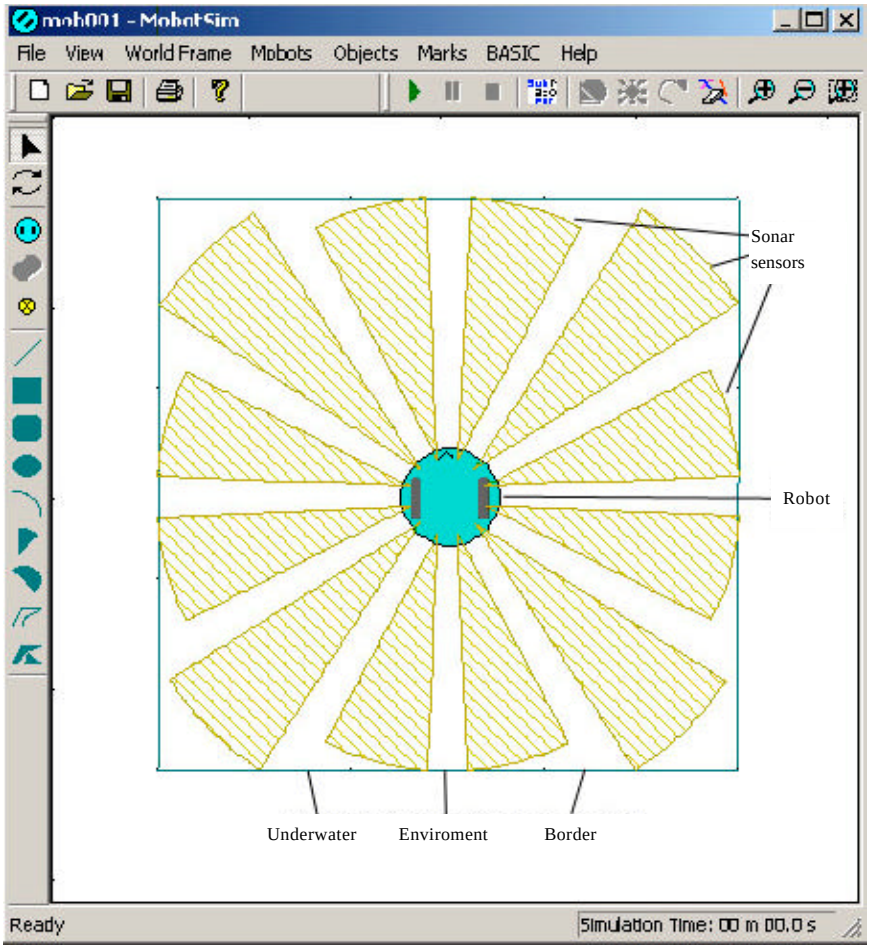

Fig. 4: MobotSim simulation software

Table 1: Sample of the recorded data

	Sample point			
Parameters	1	2	3	4
Sonar reading (m)				
1	0.6409	2.1000	1.8573	0.7299
2	0.8204	1.3994	0.9134	0.2941
3	1.6937	1.1313	0.7201	0.1970
4	2.1000	1.1305	0.7204	0.1971
5	2.1000	0.5343	0.3321	0.2956
6	1.9707	0.3993	0.2287	0.7373
7	1.4074	0.3990	0.2288	1.6302
8	0.6645	0.5342	0.3323	1.9923
9	0.5092	1.1694	0.8008	2.1000
10	0.5091	1.4800	1.8904	2.1000
11	0.6640	1.8141	2.1000	1.2261
12	0.6410	2.1000	2.1000	0.9815
Actual position				
X	1.7100	2.6800	3.0900	3.6100
Y	3.1700	1.6000	1.4300	2.8300

MATLAB function (newff) which is used to create a feedforward backpropagation network. The syntax of newff function is given below:

net = newff (Inputs, outputs, nhl)

where, net is a variable representing the neural network which is created by the function, Inputs is a matrix representing input vector, Outputs is a matrix representing target vector and nhl is number of hidden layers.

In order to train the network the author used the MATLAB function (train) and its syntax is shown below:

[net, moh] = train (Net, inputs, outputs)

where, net is the neural network that is used for training. Inputs is a matrix representing the input learning samples and Outputs representing the target learning samples.

RESULTS AND DISCUSSION

The outcome results represent the position of the robot which was calculated by the FBNN. The error can be evaluated by subtracting the neural network outcome from the actual position in that test point. Table 2 represents the sonar data and actual robot position of four test points in the underwater environment.

Table 3 and 4 show the x and y of actual location and the x and y of the location determined by the proposed 10 and 12 hidden layers neural network, respectively for four test points. Table 3 and 4 also show the x and y error which is calculated using Eq. 3 and the x and y percentage error which is calculated using Eq. 4. Each column represents one test point:

Table 2: Test points

	Test point			
Parameters	1	2	3	4
Sonar				
1	1.08130	0.53380	1.57180	0.95130
2	1.34220	0.18630	1.92310	1.18820
3	2.10000	0.10590	2.10000	2.10000
4	2.10000	0.10620	2.10000	2.10000
5	1.87490	0.18800	1.29300	2.10000
6	1.53080	0.54010	1.04010	1.66030
7	1.49400	2.10000	1.03960	0.84460
8	0.71240	2.10000	0.51040	0.35650
9	0.54920	2.10000	0.37910	0.24900
10	0.54920	1.52360	0.37900	0.24890
11	0.71130	0.72640	0.50990	0.35560
12	1.08150	0.56090	1.12530	0.84330
Actual position				
X	1.75000	3.70000	1.58000	1.45000
Y	2.73000	3.25000	2.24000	2.86000

Table 3: Results of 10 hidden layers network

	Test point			
Parameters	1	2	3	4
Actual X position (m)	1.75000000	3.7000000	1.58000000	1.45000000
Calculated X position (m)	1.74596383	3.6983343	1.58310250	1.44948900
Error in X position (m)	-0.00403617	-0.0016657	0.00310251	-0.00051100
Error (%)	0.23063830	0.0450190	0.19636140	0.03524140
Actual Y position (m)	2.73000000	3.2500000	2.24000000	2.86000000
Calculated Y position (m)	2.73127000	3.2617290	2.23711070	2.85380433
Error in Y position (m)	0.00127000	0.0117290	-0.00288930	-0.00619567
Error (%)	0.04652015	0.3608923	0.12898660	0.21663180

Table 4: Results of 12 hidden layers network

	Test point			
Parameters	1	2	3	4
Actual X position (m)	1.75000000	3.7000000	1.5800000	1.45000000
Calculated X position (m)	1.75028556	3.7071676	1.5799557	1.44992493
Error in X position (m)	0.00028556	0.0071676	-4.43E-050	-7.51E-0500
Error (%)	0.01631771	0.1937186	0.0028035	0.00517710
Actual Y position (m)	2.73000000	3.2500000	2.2400000	2.86000000
Calculated Y position (m)	2.73076340	3.2548810	2.2401767	2.86073160
Error in Y position (m)	0.00076340	0.0048810	0.0001767	0.00073160
Error (%)	0.02796337	0.1501846	0.0078888	0.02558042

$$\text{Error} = \text{Actual value-calculated value} \tag{3}$$

$$\text{Percentage error} = \frac{\text{Error}}{\text{Actual value}} * 100\% \tag{4}$$

CONCLUSION

In this study, the design and training of feedforward backpropagation neural network has been carried out to solve the underwater global localization problem. The result shows that the neural network can be used to determine the location of the robot with very small percentage of error see as shown in Table 3 and 4. Moreover, this resulted percentage of error is internationally accepted by electronic engineers.

ACKNOWLEDGEMENT

This study was supported by the internal grant of Universiti Teknikal Malaysia Melaka under grant No. PJP/2012/FKEKK(23B)/S01026. The authors would also like to thank Mr. Yeo Weng Kiong for his assistance in this project.

REFERENCES

Fang, F., M. Xudong and D. Xianzhong, 2011. An improved particle localization algorithm for mobile robot in indoor environment. Inform. Technol. J., 10: 1017-1023.

Janet, J.A., D.S. Schudel, M.W. White, A.G. England, R.C. Luo and W.E. Snyder, 1996. Global self-localization for actual mobile robots: Generating and sharing topographical knowledge using the region-feature neural network. Proceedings of the IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems, December 8-11, 1996, Washington, DC., pp: 619-626.

Kim, S.J. and B.K. Kim, 2009. A hybrid algorithm for global self-localization of indoor mobile robots with 2-D isotropic ultrasonic receivers. Proceedings of the IEEE International Symposium on Industrial Electronics, July 5-8, 2009, Seoul, South Korea, pp: 1446-1451.

Kim, S.J. and B.K. Kim, 2011. Accurate hybrid global self-localization algorithm for indoor mobile robots with two-dimensional isotropic ultrasonic receivers. IEEE Trans. Instrum. Meas., 60: 3391-3404.

Liu, Z., Z. Shi, M. Zhao and W. Xu, 2007. Mobile robots global localization using adaptive dynamic clustered particle filters. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, October 29-November 2, 2007, San Diego, CA., pp: 1059-1064.

Moreno, L., M.L. Murioz, S. Garrido and F. Martin, 2007. Evolutionary filter for mobile robot global localization. Proceedings of the IEEE International Symposium on Intelligent Signal Processing, October 3-5, 2007, Alcala de Henares, Spain, pp: 1-6.

Se, S., D.G. Lowe and J.J. Little, 2005. Vision-based global localization and mapping for mobile robots. IEEE Trans. Robotics, 21: 364 -375.

Song, Q. and L. Liu, 2013. Simulation design of robot localization and navigation system. J. Applied Sci., 13: 2213-2216.

Tamimi, A., H. Sadjadian and H. Omranpour, 2010. Mobile robot global localization using imperialist competitive algorithm. Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering, August 20-22, 2010, Chengdu, China, pp: V5-524-V5-529.

Xie, J.P., F. Nashashibi, M. Parent and O.G. Favrot, 2010. A real-time robust global localization for autonomous mobile robots in large environments. Proceedings of the 11th International Conference on Control Automation Robotics and Vision, December 7-10, 2010, Singapore, pp: 1397-1402.

Yeo, W.K., D.F.W. Yap, D.P. Andito and M.K. Suaidi, 2011. Grayscale MRI image compression using feedforward neural networks. Proceedings of the 6th International Conference on Broadband Communications and Biomedical Applications, November 21-24, 2011, Melbourne, VIC, Australia, pp: 39-42.