Neural network simulation for obstacle avoidance and wall follower robot as a helping tool for teaching-learning process in classroom

T Dewi, P Risma, Y Oktarina and M Nawawi

Department of Electrical Engineering, Politeknik Negeri Sriwijaya
Jalan Srijaya Negara Palembang, Indonesia, 30139

tresna_dewi@polsri.ac.id

Abstract. One of the most applied technologies is robotics, and one the most applied type of robots is a mobile robot. To provide a tool for the interactive teaching-learning process in the classroom, a simulation program is needed. Low price and easy to use software is preferable considering not every polytechnic can provide expensive simulation software for its students. Robotics related curriculum should include artificial intelligence since it is functioning as a brain to the robot. One type of artificial intelligence is neural networks. This study presents the application of neural networks by using low price simulation software. The simulation results show the feasibility of utilizing the software for the students to learn neural networks by variating the scenarios of mobile robots application, adding more robots, more sensors and writing the programs to control the robots. The contribution of this paper is to show and inspire teachers to create a more interactive teaching-learning process in the classroom by using low price, user-friendly software, and to encourage them to search more alternatives of low price or even free software to be the teaching tool in the classroom.

1. Introduction
Polytechnic as a vocational education has to focus its curriculum and educational process to the application of current technology rather than theoretical analysis [1]. The vocational education insists the teachers to leave the conventional learning method into a more interactive and applicative learning method. The curriculum has to be always updated with the technology in the industry [2]. Robotics is the rapidly increasing technology in the industry, therefore polytechnic also has to provide the basic knowledge for its students to understand robots and how to design it [3-6]. To provide robotics course means to integrate several subjects that support it, such as mathematical analysis, computing, electronics, mechanics, etc. The integration of this several subjects can be a subject discussing the main controller or the brain of the robot.

In order to function well, a robot has to be equipped with a brain, therefore one of the classes should be provided in robotics curriculum is the one that discussed the brain of the robot. This brain is called artificial intelligence [7]. The installed artificial intelligence in a robot will provide a robot a way to think and decide the best options to finish the assigned task. Artificial intelligence class should be included in the polytechnic curriculum if that polytechnic decides to include robot’s related topics [3-6, 8].
Before applying to the real robot, simulation is the best method to get the feel of the environment and avoid dealing with the uncertainties for the beginner when interacting with the robot for the first time [9-11]. Simulation is bringing the textbook to live (animation) with much less risk of damaging real system with trial and error. The method of applying the right simulation software in vocational education has made the class more interesting rather than focus within textbook theory. Students can observe the effects of creating a certain system justification and prepare them to work with the real system.

Neural networks are a method for artificial intelligence [7]. It imitates the function of neural in a human brain. This type of artificial intelligence is very suitable for a system with a series of sensors and not easy to be modelled into certain mathematical equations. Data and information received from sensors are utilized to be the input for training the neurons or synaptic weights. The learning process will create output that will be fed back to the controller system. The basics applications of neural networks in robotics are obstacle avoidance and wall follower robot [12-22]. From this basic, students can improve and develop the examples to a more complicated system.

The current method of teaching-learning in the classroom is still in a conventional way, where the teacher teaches and students only listen. This conventional way can be improved by creating an interactive class and involving students more, this way not only can improve students understanding but also their interest to participate more in class [3, 4, 6]. Providing simulation program and involving students actively in working on simulation to show the theoretical and application of the taught subject is one of the excellent ways to create an interactive class. The problem with providing simulation is to find the right software program. In many cases of universities and polytechnics in developing countries cannot provide expensive software program such as MATLAB [9-11]. Therefore, choosing the right simulation software is a crucial decision. Product price becomes one of the considerations in purchasing the software.

Mobotsim is low price software and offers 1-month trial to simulate a robot. Obstacle avoidance and wall follower robot can be easily simulated in this software [21]. Students can easily develop the robot system from the examples to any systems. This paper presents the application of low price software to create teaching module for learning neural networks in the classroom. The procedure to develop the program and program examples of obstacle avoidance and wall follower robots will be provided in simulation results to show the feasibility of using this software (and any software alike) in creating a helping tool for teaching-learning process of artificial intelligence class. The contribution of this paper is to show and inspire teachers to create more interactive teaching-learning process in the classroom by using low price, user-friendly software, and to encourage them to search more alternatives of low price or even free software to be the teaching tool in the classroom.

2. Introduction to Neural Networks
The Neural Networks was first introduced by McCulloch and Pitts in 1943, as the result of a brain study, and in 1949 Hebb presented the concept of synaptic weight [6]. Neural networks are the type of artificial intelligence that use a large number of nonlinear processors called neurons. This neuron imitates human brain and has a number of internal parameters called synaptic weights. The neurons are characterized by a state, weighted inputs defined by a learning process and a state equation for their dynamic operation. Neural networks get the input from the sensory data information, process it and learn from the data information.

The most popular type of neural networks application is multilayer perceptron where the feedback control can approximate any functions such as continuous and integrated function. The structure of layers design consists of input layers, hidden layers, and output layers. Hidden layers are the part where the learning process is happening and adjusting online.

In robotics, neural networks get input data from sensory information and/or robot modelling; kinematics and/or dynamics modelling. The inputs are processed in hidden layers, and the results are the output of the system. The processes are updated online based on data received by the sensors.
and/or the updated mathematical equations in Jacobian derived from kinematics and dynamics of the system. The generic output equations for neural networks is given by equation (1).

\[ y_i = \sum_{j=1}^{l} [v_{ij}\sigma(\sum_{k=0}^{n} w_{jk}x_k)], \quad i = 1, 2, \ldots, m, \]

where \( y_i \) is the output, \( w_{jk} \) are weight function as the input to hidden layers, \( v_{ij} \) are the output from hidden layers, \( x_k \) \((k = 0,1,2, \ldots, n)\) is the control input, \( l \) is numbers of neurons in the hidden layers, \( m \) is the output neurons, \( l \) is a number of neurons in the hidden layers, \( m \) is the output neurons, and \( \sigma \) is the nonlinear hyperbolic tangent activation function, given by equation (2).

\[ y = V^T\sigma(W^Tx), \]

where \( N_1, N_2, N_3, \ldots, N_n \) is the vector function; related to neurons, and \( W \) and \( V \) are weight of neurons.

3. Simulating mobile robot with neural networks

For students learning in the classroom, the most applied type of mobile robots is considered, the two-differential driven mobile robot, shown in the figure 1, where \( v \) and \( \omega \) are the translational and rotational velocity of the robot, \( \phi \) is the orientation of the robot, \( L \) is the distance from driving wheels to the robot axis in \( y \)-direction, \( \theta_L \) and \( \theta_R \) are left and right tires’ angles, and \( S_1 \) to \( S_4 \) are the installed proximity sensors. This type of robot is suitable to show the application of neural networks in robotics.

In order to control the robot, modelling of the robot is necessary. There are two types of modelling, kinematics modelling; describing robot movement without considering the input to the robot, and dynamics modelling where the input forces are considered.

![Figure 1. Two-differential driven mobile robot.](image)

3.1. Kinematics modeling

In kinematics modelling, the robot movement in its coordinate frame is derived relative to world coordinate frame. In this modelling, the pose, position and orientation, of the robot is defined as \( q = [x\ y\ \phi]^T \) where \( x \) and \( y \) are robot’s position in \( x \) and \( y \)-axis, and \( \phi \) is the orientation of the robot. From this pose, the translational and rotational velocities are derived as in equation (3).

\[ \dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = f(\theta_R, \theta_L, L, r, \phi), \]

where \( \dot{x} \) and \( \dot{y} \) are the translational velocities in \( x \) and \( y \) axis respectively, and \( \omega = \dot{\phi} \) is the rotational velocity.

In order to get the value of tires orientation and velocity, the inverse kinematics of the robot is derived as in equation (4).
\[
\begin{bmatrix}
\dot{\theta}_R \\
\dot{\theta}_L
\end{bmatrix} = f(\dot{x}, \dot{y}, \dot{\phi}).
\]  \hspace{1cm} (4)

From equation (4), the relation between robot’s translational \(v\) and rotational \(\omega\) velocities and both tires velocities are shown in equation (5).

\[v = r \frac{\dot{\theta}_R - \dot{\theta}_L}{2}, \quad \text{and} \quad \omega = \frac{r}{2l} (\dot{\theta}_R - \dot{\theta}_L).
\]  \hspace{1cm} (5)

3.2. Dynamics modelling

Dynamics modelling considers the total forces applied to the system, including disturbance. The velocities and accelerations are derived from the pose (positions and orientations) given in Kinematics. The complete generic dynamics modelling of two differential driven mobile robots that suffers the non-holonomic constraints is shown in equation (6).

\[D(q)\ddot{q} + C(q, \dot{q}) + G(q) + M^T \lambda = \mathbf{E}\tau,
\]  \hspace{1cm} (6)

where \(D(q)\) is the inertia matrix, \(C(q, \dot{q})\) is the centripetal and coriolis matrix, \(G(q)\) is the gravitational vector, \(M^T\) is the \(m \times n\) matrix of the \(m\) non-holonomic constraints, \(E\) is a non-singular transformation matrix, and \(\tau\) is total force and/or torque applied to the system, including frictions occurs due to actuators motions.

The constraints matrix \(M^T\) can be eliminated by a constraint-free model of \(n \times (n - m)\) matrix \(B(q)\). By eliminating \(M^T\), the dynamics equations of two differential driven mobile robot can be written using equation (7).

\[\bar{D}(q)\ddot{q} + \bar{C}(q, \dot{q}) + \bar{G}(q) = \bar{E}\tau,
\]  \hspace{1cm} (7)

where \(\bar{D}, \bar{C}, \bar{G}\) and \(\bar{E}\) show the justification of the original representation in equation (6).

Jacobian matrix is derived from kinematics and dynamics of the robot and defined as in equation (8).

\[J = \frac{\partial q}{\partial v_c} = \frac{\partial q}{\partial v} \frac{\partial v}{\partial \tau} \frac{\partial \tau}{\partial e_c} \frac{\partial e_c}{\partial v_c}
\]  \hspace{1cm} (8)

where \(v_c\) is the control law and \(e_c\) is the error dynamics. The Jacobian is needed to be the input for hidden layers of neural networks and updated online.

The application of neural networks to control a mobile robot can be applied by deriving kinematics and dynamics model, as shown in figure 2 or directly using the sensory data information as the input, shown in figure 3. Figure 2(a) and 3(a) show the diagram block representation of both methods, and figure 2(b) and 3(b) show the layer presentation, where \(v\) and \(\omega\) are the translational and rotational velocity, \(S_1\) to \(S_4\) are sensor 1 to 4, and \(m_1\) and \(m_2\) are motor left and right velocity.
Figure 2. Neural networks controller design without considering robot’s kinematics and dynamics.

Figure 3. Kinematics and dynamics modelling application in neural networks controlled design.
4. Simulation results
A simulation of a system is operating the model of the system by reconfiguring and experimenting using the software. Teaching-learning process in vocational education insists the real application to understand the topic more. Learning and understanding neural networks will be easier and deeper by simulating its application using mobile robot simulation software, such as Mobotsim from Mobotsoft.

Mobotsoft is low price software that is possible to purchase by students. The students can use the trial version for 30 days or for 100 times usage, which one comes first. The simulation is written with Basic programming language and the environment can be customized according to the scenarios assigned by the programmer. Mobotsim is already giving examples of a mobile robot applying neural networks controller that can be developed for more sensors, more complicated environment and/or for more robots. This study shows four cases of neural networks application for two-wheel differential driven mobile robot. Robot(s) has (have) 4 to 8 proximity sensors for obstacle avoidance and/or wall follower.

4.1. Case 1: a wall follower robot with 4 stopping points
Case 1 in figure 4 has the scenario of a patrol robot. It comes to 3 checking points for 3 rounds, and in the 4th round, the robot goes to its base. Robot follows the wall and stops at the stopping points. One stopping point for one round, after four rounds, the robot will stop at the final position.

Mobile robot applied in figure 4 is equipped with 8 sensors assigned to follow the wall. The neural network controller is used making the robot follows the wall, and this wall-following scheme is combined with the virtual force application. The stopping points considered to have an attractive force for one time after the robot stops one time at a point, the point has no attractive force anymore, therefore for the next round the robot will skip it and stop at the next stopping point. Figure 4(a) shows the robot at the starting point and figure 4(b) shows robot at the finish point.

Figure 4. Case 1: a robot follows the wall, and stop at one point at one round, and on the 4th round, the robot reaches the final point.

4.2. Case 2: two wall follower robots with many stopping points for one robot
Figure 5 shows the environment of the simulation program. The programmer can develop the program according to any scenarios, sensors applied to the robot can be reduced, robots and obstacles can be added, and the syntax can be developed as required.

Case 2 shows in figure 5 is two robots in one environment. The first robot (the green robot) is a robot that applied in a cookies manufacture company. The robot will work from supply and ingredient (suing) post to stir post, to bake post and finally to packing post. After the first robot reaches the packing post, robot 2 will move bringing the cookies to the market.

The robot is also equipped with 8 proximity sensors to follow the wall. The points are applying virtual attractive force to make the robot stop at some time, using a time delay. Figure 5(a) shows
robot 1 on starting point, figure 5(b) and 5(c) show the robot is on the move from point to another. Figure 5(d) shows robot 1 reaches the final position and robot 2 starts to move. Figure 5(e) shows robot 2 reaches its final position. Figure 6 shows the Basic programming language for setting the simulation.

Figure 5. Case 2: 2 robots, one robot stops in many points and robot 2 moves after robot 1 reach the final position.
4.3. Case 3: a delivery robot

Case 3 shown in figure 7 is a delivery robot. The robot delivers food to the final position. The robot will reach the target by choosing the path automatically and avoid obstacles. This scenario is not applying the optimum path; it chooses the path with fewer obstacles. The robot is also equipped with 8 proximity sensors and attractive virtual forces on points.

![Diagram of delivery robot](image)

(a) Robot at starting point  (b) Robot at final point

**Figure 7.** Case 3: a food delivery robot, considering obstacles avoidance

4.4. Case 4: 2 robots in two parallel environments

Case 4 in figure 8 is the same with case 2 in figure 5. Case 4 just to show that more robots can be added and more environments can be set up. Case 4 is to show the flexibility of simulating neural networks application in a two-wheel differential driven mobile robot and more.
Figure 8. Case 4: 2 parallel environment setting.

Figure 4 to 8 show the examples of simulating neural networks application in a mobile robot. The flexibility of the simulation software program helps the students to understand the topic more and deeper by improving and developing it. The environments and robots addition can be done in a fun way and the controller can be set in Basic programming language. This simulation helps the teacher to reach the objective of vocational education where more applications are better rather than theoretical study.

5. Conclusion
Simulation for learning artificial intelligence is necessary to show the application of robotics without having to deal with the uncertainties of the real system. This paper has presented the utilization of low price software namely Mobotsim to learn neural networks interactively. The basic applications of neural networks in mobile robot are chosen, the obstacle avoidance and wall follower. Students can learn the application of Neural networks in mobile robot by setting the scenario, choosing the robot or robots, and programming it (them) with Basic programming language. The simulation results of 4 scenarios show that teaching-learning process in the classroom for understanding neural networks application for obstacle avoidance and wall follower robots can be conducted by using a simulation software as a helping tool. This paper has shown that application of low price software can be an alternative to create an interactive teaching-learning process in the classroom, and authors would like to inspire and encourage teachers to search more alternatives of low price or free software (open source) for teaching tool in the classroom.
References

[1] Jianyu W, Xi L and Chen X 2012 Vocational Ability Oriented Modularized Curriculum for Advanced Vocational School 2012 International Conference on Future Computer Supported Education, IERI Procedia vol 2 pp 897-900

[2] Kuangdi X 2008 Engineering Education and Technology in a Fast-Developing China Technology in Society vol 30 pp 265-74

[3] Assaf D, Larsen J C and Reichardt M 2012 Extending Mechanical Construction Kits to Incorporate Passive and Compliant Elements for Educational Robotics 3rd International Conference on Robotics in Education pp 33-40

[4] Ospennikova E, Ershov M and Iljin I 2015 Educational Robotics as an Innovative Educational Technology Worldwide trends in the development of education and academic research, Procedia – Social and Behavioral Sciences vol 214 pp 18-26

[5] Baros T T T and Lages W F 2012 Development of a Firefighting Robot for Educational Competition 3rd International Conference on Robotics in Education pp 47-54

[6] Petrovic P Having Fun with Learning Robots 3rd International Conference on Robotics in Education pp 105-12

[7] Tzafestas S G 2014 Introduction to Mobile Robot Control Elsevier

[8] Alemany J and Cervera E 2012 Appealing Robots as a Means to Increase Enrolment Rate: a Case Study 3rd International Conference on Robotics in Education, pp 15-9

[9] Arlegui J, Moro M and Pina A 2012 Simulation of Robotic Sensors in BYOB 3rd International Conference on Robotics in Education pp 25-32

[10] Pandey A and Parhi D R 2014 MATLAB Simulation for Mobile Robot Navigation with Hurdles in Cluttered Environment Using Minimum Rule Based Fuzzy Logic Controller 2nd International Conference on Innovations in Automation and Mechatronics Engineering ICIAME 2014, Procedia Technology vol 14 pp 28-34

[11] Casini M and Garulli A 2016 MARS: a Matlab simulator for mobile robot experiments IFAC-PaperOnline vol 49 no 6 pp 069-074

[12] Farooq U, Amar M, Asad M U, Hanif A and Saleh S O 2014 Design and Implementation of Neural Network Based Controller for Mobile Robot Navigation in Unknown Environment International Journal of Computer and Electrical Engineering vol 6 no 2 pp 83-9

[13] Engedy and Horváth G 2009 Artificial Neural Network Based Mobile Robot Navigation 2009 IEEE International Symposium on Intelligent Signal Processing, Budapest pp 241-6

[14] Capi G, Kaneko S and Hua B 2015 Neural network based Guide Robot Navigation: An Evolutionary Approach 2015 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS2015) pp 74-79

[15] Chi K H and Lee M F R 2011 Obstacle avoidance in mobile robot using Neural Network 2011 International Conference on Consumer Electronics, Communications and Networks (CECNet) pp 5082-5

[16] Seyr M, Jakubek S and Novak G 2005 Neural Network Predictive Trajectory Tracking of an Autonomous Two-wheeled Mobile Robot 16 Triennial World Congress, Prague, Elsevier IFAC Publication pp 385-90

[17] Boukens M and Boukabou A 2017 Design of an Intelligent Optimal Neural Network-Based Tracking Controller for Non-holonomic Mobile Robot System Neurocomputing 226 p 46-57

[18] Janglová D 2004 Neural Network in Mobile Robot Motion International Journal of Advanced Robotics Systems vol 1 no 1 pp 15-22

[19] Shamsfakhr F and Bigham B S 2016 A Neural Network Approach to Navigation of a Mobile Robot and Obstacle Avoidance in Dynamic and Unknown Environments Turkish Journal of Electrical Engineering & Computer Science

[20] Larasati N, Dewi T and Oktarina Y 2017 Object Following Design for a Mobile Robot Using Neural Network Computer Engineering and Application Journal vol 6 no 1

[21] Fierro R and Lewis F L 1998 Control of a Nonholonomic Mobile Robot Using Neural Network
IEEE Transactions on Neural Networks vol 9 no 4 pp 589-600

[22] Mohareri M, Dhaouadi R and Rad A B 2012 Indirect Adaptive Tracking Control of a nonholonomic Mobile Robot Via Neural Network Neurocomputing vol 88 pp 54-66