

Design and Implementation of a Real-Time Intelligent Controller for a Differential Drive Mobile Robot

Zeyad Yousif Abdoon Al-Shibaany (IEEE Member), Nadia Abd Alzhra Jassar

Abstract— The paper proposes an intelligent kinematic controller to guide a differential drive mobile robot system, which is a nonholonomic differential drive mobile robot, to follow a predefined path. The controller has been designed and implemented by using a Multi-Layer Perceptron MLP neural network. The error back propagation algorithm has been used in order to train the neural network to identify the kinematic model of the differential drive mobile robot. The neural network based controller has been trained offline and then online in order to calculate the required linear and angular velocities to guide the robot throughout the predefined path. The controller has been implemented by using a real-time and FPGA based input/output board, which is fully compatible with LabVIEW2012 software, and then the controller is being deployed to the mobile robot platform to allow for autonomous navigation. The simulation and experimental results explained and proved that the proposed controller can successfully guide the robot along the required path.

Index Terms— Nonholonomic, Mobile Robot, Neural Network, Real Time, Adaptive Controller, LabVIEW, National Instrument.

1 INTRODUCTION

In general, a mobile robot can be defined as a moveable system which has the ability to navigate and interact with its environment by using its sensors and actuators [1]. Wheeled mobile robots have become an interesting area of research over the last few decades. There are many applications for the mobile robots such as materials handling, and household services.

An intelligent autonomous mobile robot has the ability to detect and react to its surroundings when it is moving from a start position to its final position [1]. The detection and reaction abilities can be achieved by using the sensors and actuators of the mobile robot. The two essential components required to read data from the sensors and make decisions based on that sensory data are the processor and control system. As the requirements of each mission can vary; those four elements must be presented for any mobile robot to be autonomous and to function correctly.

By using its sensors, the mobile robot is capable of observing the environment and providing useable data to the processor which in turn uses this data via the control algorithm, to guide the mobile robot throughout the environment [2]. The control system provides the mobile robot with the algorithm required to achieve various missions. The motion control of mobile robot systems has gained considerable attention in the last few years as the applications of mobile robots have in-

creased significantly [3].

Many control strategies have been developed in order to provide a robust feedback control rule for mobile robot systems. The control algorithm of any mobile robot depends either on the dynamic model or the kinematic model of the mobile robot.

The dynamic controller, sometimes called the torque controller depends on the dynamic model of the mobile robot such that the actual velocity of the mobile robot converges to the required velocity, while the main aim of the kinematic controller is to output a velocity for the mobile robot in order to reduce the error between the required path and the actual path of the mobile robot system [3].

This paper considers the use of the neural networks because of their learning and adaptation capabilities. Neural network techniques demonstrate very good performance when they have been used in the motion control of mobile robots [3]. Other advantages of a neural network are its ability to provide a very good model of any nonlinear system, and its parallel structure makes the neural network performance faster than the performance of other classical artificial intelligence techniques such as a Genetic algorithm. In addition to these, the simple implementation of neural network has made the use of it much easier than other techniques.

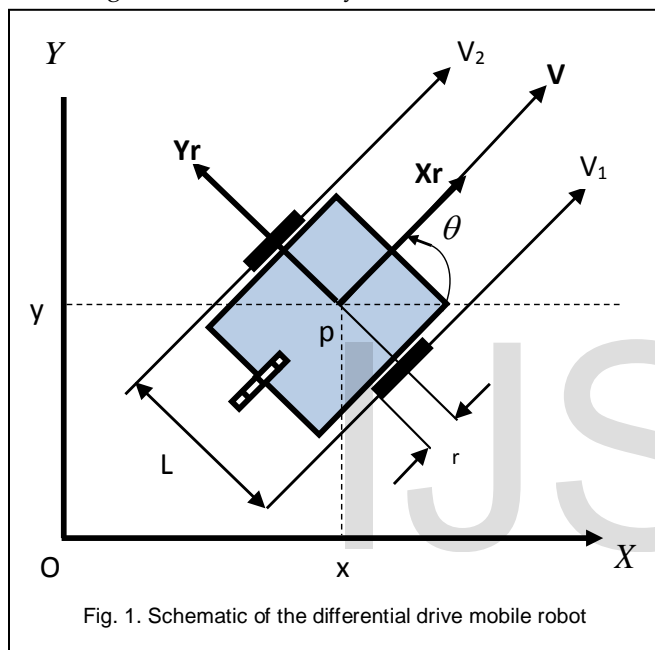
A neural network based controller is designed to guide a nonholonomic mobile robot to follow its reference path in [4]. Lyapunov theorem is used to test the stability of the controller and the simulation results show that the controller functions well. A Q-learning neural network based controller is presented in [5]. The main purpose of this controller is to enable the mobile robot to determine the optimal path from its initial position to its final goal. The validation and performance of the controller is proved by experimental results as well as by

- Zeyad Yousif Abdoon Al-Shibaany is currently an assistant lecturer in the Control and Systems Engineering Department - University of Technology / Iraq. He obtained his Master degree in Mechatronics engineering from Newcastle University - United Kingdom with a Distinction and his rank was the FIRST among all the postgraduate students in 2012. He also had been awarded a prize for the BEST performing Master student in the School of Mechanical and Systems Engineering - Newcastle University for the academic year 2011/2012. E-mail: zeyad.al-shibaany@ieee.org.
- Nadia Abd Alzhra Jassar is currently an engineer in the Laser and Optoelectronics Engineering Department - University of Technology / Iraq. She obtained a B.Sc. degree in Control and Systems Engineering in 2007 from the University of Technology / Iraq. E-mail: nadia_abdalzhra@yahoo.com

simulation results. A nonlinear controller is designed and implemented based on neural network technique to enable a one wheel mobile robot to be balanced in [6]. The practical results proved that the neural network based controller is much better than classical controllers as the neural network has a very good adaptation feature.

2 THE KINEMATIC MODEL OF THE DIFFERENTIAL DRIVE MOBILE ROBOT

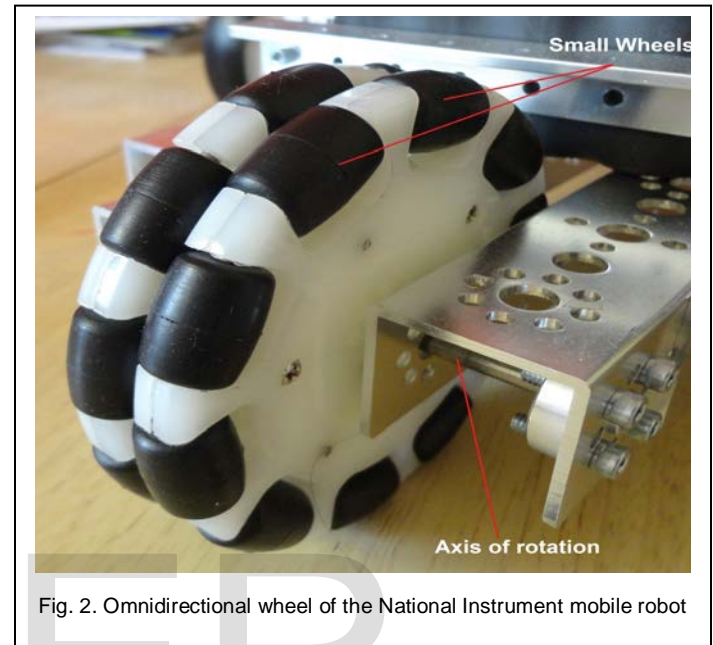
The mobile robot platform shown in fig. 1 has two driving wheels, and one passive wheel which is added for the purpose of stability as the robot requires at least three points of contact with the ground to be statically stable.



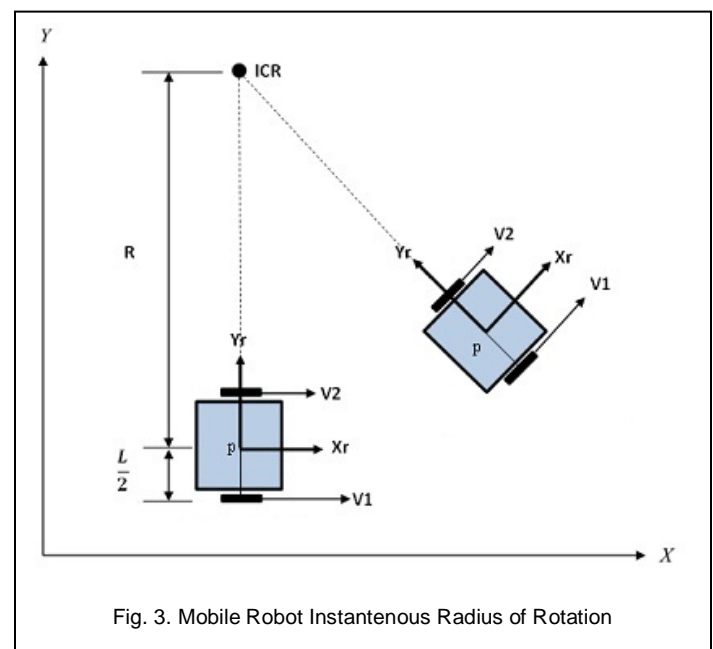
The driving wheels are parallel to each other. The rear passive wheel in this case is an omnidirectional wheel that has small wheels mounted around its circumference. The axis of rotation of these small wheels is perpendicular to the axis of rotation of the omnidirectional wheel in order to eliminate the friction effect when the mobile robot is turning. The small wheels give the omnidirectional wheel an additional degree of freedom. This means the omnidirectional wheel can move forward or backward by rotating around its axis of rotation, and can move laterally when the small wheels are rotating around their axis of rotation. Figure 2 below shows the omnidirectional wheel of the differential drive mobile robot (National Instrument starter kit) which is used in this paper.

The two driving wheels have the same radius which is denoted by r in fig. 1. The distance between the two wheels is denoted by L . The mobile robot (NI starter kit) is designed and manufactured so that its center of mass is located at point p which is located at the middle of the distance between the two wheels. There are two frames associated with the mobile robot. The first one is the robot frame which is rigidly attached to the mobile robot and denoted by X_r and Y_r . The second

frame is the global (reference) frame which is fixed and denoted by X and Y as shown in fig. 1. The posture of the mobile robot is represented by a vector $P = [x \ y \ \theta]^T$, where x and y are the coordinates of the point p according to the reference frame, and θ is the mobile robot orientation which is the angle between the x axis of the global frame (X) and the x axis of the robot frame (X_r).



The two driving wheels are free to rotate about their axes of rotation which are coinciding with the y axis of the robot frame (Y_r). The linear velocity of the robot (V) is orthogonal to the axis of rotation of the two driving wheels. According to the nonholonomic constraints, the robot is assumed to have no lateral velocity which means the wheels are purely rotating with no slipping [7].



The two driving wheels of the differential drive mobile robot are moving around the instantaneous center of rotation (ICR) with the same angular velocity (W) at each time instant [8]. The instantaneous center of rotation can be found by drawing a line, which is normal to the driving wheels plane, at a specific time instant then to draw the same line but at different time instant and then the ICR is the point where those two lines meet [9]. This means that the robot is following a circular path with a radius of R and the center point is the ICR [9]. Figure 3 clarifies the process of finding the instantaneous center of rotation ICR.

The kinematic model of the differential drive mobile robot system, which is shown in fig. 1, can be derived as follow:

$$W(t) = \frac{d\theta(t)}{dt} \quad (1)$$

$$V(t) = W(t)R(t) \quad (2)$$

Now it is easy to calculate the linear velocities of both the right and the left wheels:

$$V_1(t) = W(t) (R(t) + 0.5L) \quad (3)$$

$$V_2(t) = W(t) (R(t) - 0.5L) \quad (4)$$

By solving Eq. 3 and Eq. 4, the ICR can be calculated as:

$$R(t) = 0.5L \left(\frac{V_2(t) + V_1(t)}{V_2(t) - V_1(t)} \right) \quad (5)$$

The linear and angular velocities of the mobile robot can be represented in terms of the linear velocities of the two driving wheels as follow:

$$V(t) = \frac{V_1(t) + V_2(t)}{2} \quad (6)$$

$$W(t) = \frac{V_1(t) - V_2(t)}{L} \quad (7)$$

In term of the global frame, the kinematic equations of the mobile robot can be represented as follow:

$$\dot{x}(t) = V(t) \cos(\theta(t)) \quad (8)$$

$$\dot{y}(t) = V(t) \sin(\theta(t)) \quad (9)$$

$$\dot{\theta}(t) = W(t) \quad (10)$$

In order to obtain the position and orientation of the mobile robot, it is required to integrate Eq. 8, 9 and 10 as illustrated below:

$$x(t) = x_0 + \int_0^t V(t) \cos(\theta(t)) dt \quad (11)$$

$$y(t) = y_0 + \int_0^t V(t) \sin(\theta(t)) dt \quad (12)$$

$$\theta(t) = \theta_0 + \int_0^t W(t) dt \quad (13)$$

Where (x_0, y_0) is the initial position of the mobile robot, and θ_0 is the initial orientation of the mobile robot.

3 STRUCTURE OF THE ADAPTIVE KINEMATIC CONTROLLER

The structure of the adaptive kinematic controller is shown in fig. 4 in the form of a block diagram. The navigation of the mobile robot starts with generating the required trajectory where $P_r = [x_r \ y_r \ \theta_r]^T$ is the required pose of the mobile robot. This required pose is used by the inverse kinematic of the mobile robot to generate the required linear and angular velocities for the specified trajectory with the required orientation. The error vector is then calculated by comparing the required pose of the mobile robot with the actual pose of the robot. This error is calculated and needs to be transformed into the mobile robot frame by using the transformation matrix R . This error is then used by the controller to produce a control action to force the mobile robot to follow the required path.

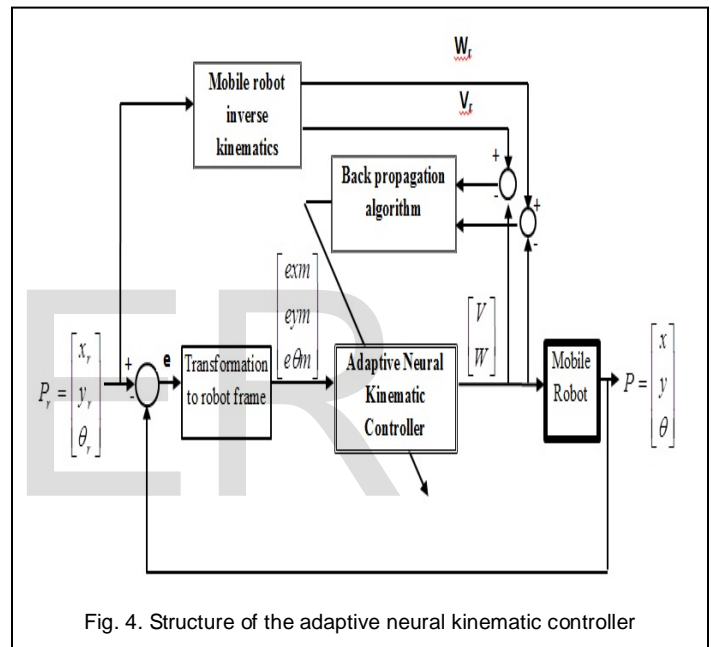


Fig. 4. Structure of the adaptive neural kinematic controller

The kinematic model of the mobile robot can be written in the form:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} \cos(\theta(t)) & 0 \\ \sin(\theta(t)) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V(t) \\ W(t) \end{bmatrix} \quad (14)$$

Let (x_r, y_r) is the required position of the mobile robot, and θ_r is the required orientation of the mobile robot, so that the required posture vector is $P_r = [x_r \ y_r \ \theta_r]^T$. The actual position of the mobile robot is assumed to be (x, y) and the actual orientation of the mobile robot is assumed to be θ , so that the actual posture of the mobile robot is $P = [x \ y \ \theta]^T$. Then the error $e = [e_x \ e_y \ e_\theta]^T$ can be calculated in the global frame:

$$\begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} \quad (16)$$

The error vector e needs to be transferred into the robot frame by using the orthogonal rotation matrix R [10]:

$$em = R * e \tag{17}$$

$$\begin{bmatrix} exm \\ eym \\ e\theta m \end{bmatrix} = \begin{bmatrix} \cos(\theta(t)) & \sin(\theta(t)) & 0 \\ -\sin(\theta(t)) & \cos(\theta(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} ex \\ ey \\ e\theta \end{bmatrix} \tag{18}$$

Figure 5 illustrates the error transformation from the global frame into the robot frame. The error vector em is fed into the controller and then it is the responsibility of the controller to generate the required linear and angular velocities to reduce the tracking error.

The controller is designed based on the MLP neural network topology. The adaptive neural kinematic controller learns the behavior of the differential drive mobile robot inverse kinematics by making use of the neural network's ability to learn the behavior of nonlinear systems. The identification process includes two stages: offline and online

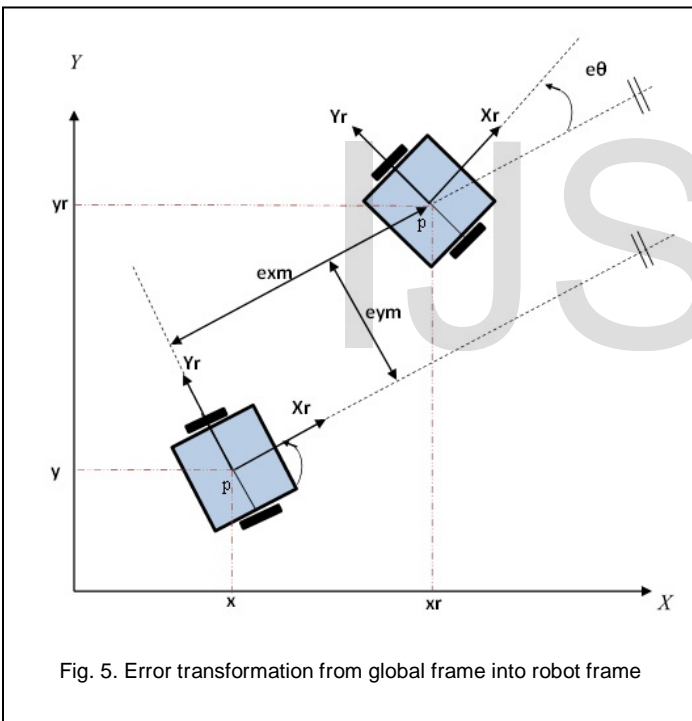


Fig. 5. Error transformation from global frame into robot frame

Figure 6 shows the MLP neural network which consists of three layers: input layer, hidden layer and output layer. The input layer has only three nodes because the input to the controller is the error vector which includes the error in robot x coordinate, the error in robot y coordinate and the robot orientation error. In the case of the input layer in fig. 6, the input nodes are drawn as solid nodes which mean that those nodes do nothing more than just collect the input of the neural network.

The number of nodes in the hidden layer can vary depending on the required performance of the neural network. The suitable number of nodes in the hidden layer one can start with is equal to $(2n+1)$ where n is the number of nodes in the input

layer [11]. The activation function of the hidden layer nodes in this case is a bipolar sigmoid function. Table 1 explains the abbreviations that have been used in fig. 6. The output of the neural network is calculating as follow:

$$O_k = \sum_{j=1}^{nj} hnet_j \times W_{kj} + bias \times P_{k2} \tag{19}$$

where $hnet_j$ is the output of the hidden layer node j and is being calculated as:

$$hnet_j = H(h_j) \tag{20}$$

where:

$$h_j = \sum_{i=1}^{ni} z_i \times V_{ji} + bias \times P_{j1} \tag{21}$$

$$H(h_j) = \frac{2}{1 + e^{-h_j}} - 1 \tag{22}$$

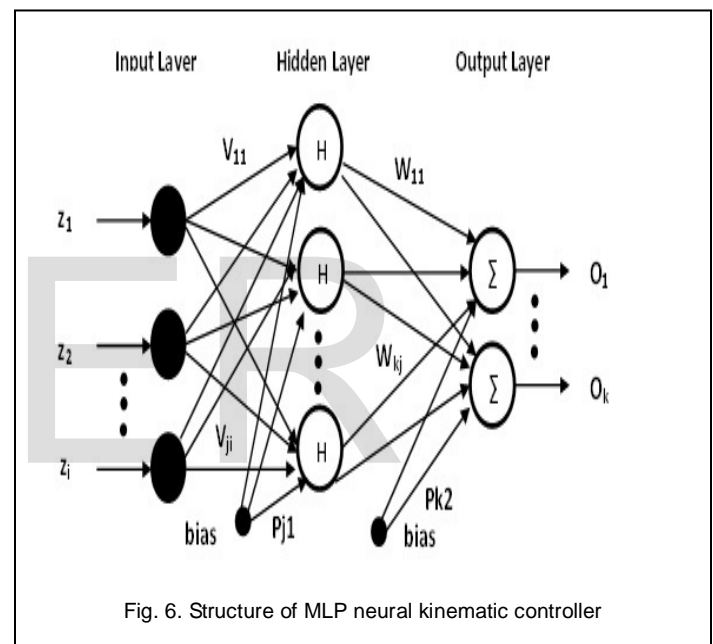


Fig. 6. Structure of MLP neural kinematic controller

TABLE 1
 NEURAL NETWORK PARAMETERS

Abbreviation	Meaning
V_{ji}	The weight between input node (i) and hidden node (j)
W_{kj}	The weight between hidden node (j) and output node (k)
P_{j1}	The weight between bias and hidden node (j)
P_{k2}	The weight between bias and output node (k)
H	Activation function of hidden layer node
z_i	The input to the input node i
O_k	The output of the output node k
n_i	Number of nodes in the input layer
n_j	Number of nodes in the hidden layer
n_k	Number of nodes in the output layer

The error back propagation algorithm technique was used to adjust the weights of the neural network by calculating the error in the output layer and then propagating back this error in order to adjust the weights of the neural network. The weights are adjusted offline and then online.

The neural network learns the exact behavior of the inverse kinematic model of the mobile robot and can then serve as a controller to guide the robot during the trajectory tracking. The weights of the neural kinematic controller can be adjusted during the navigation in case the robot is subjected to an external disturbance. Those weights are adjusted by the back propagation algorithm as shown in fig. 4.

4 SIMULATION RESULTS

The proposed controller in this paper was designed and tested using MATLAB program before being applied in LabVIEW in order to test the controller functionality before applying it to the real mobile robot platform. In this simulation test, the robot is required to follow a predefined path which is a lemniscate. The reason for choosing this path is because it examines all the possible postures of the mobile robot where at each sample the ICR is different. The robot is also required to follow the path with a predefined orientation.

A training set of 100 samples and a learning rate of 0.1 were used to train the neural kinematic controller. The neural kinematic controller learnt the behavior of the mobile robot model after 2000 epochs with a mean square error equal to 3×10^{-6} as shown in fig.7.

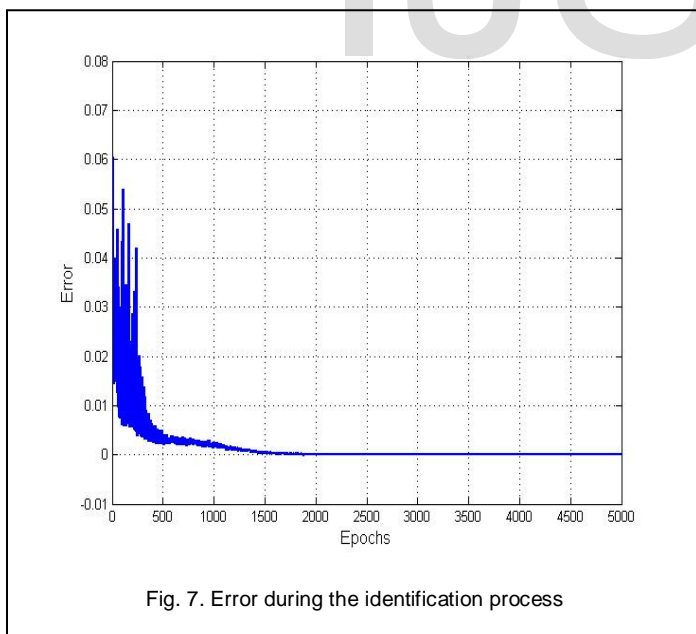


Fig. 7. Error during the identification process

The sampling time in this case is 0.5 second, and the physical parameters of the differential drive mobile robot are: $r=5$ cm and $L=36$ cm. Figure 8 shows the required trajectory as well as the simulated path of the mobile robot.

As shown in fig. 8, the robot follows the required path with minimum trajectory tracking error. Figure 9 shows the error in

both x and y coordinates.

The required orientation of the mobile robot and the simulated orientation are given in fig. 10. It can be clearly seen that the robot tracks the reference trajectory.

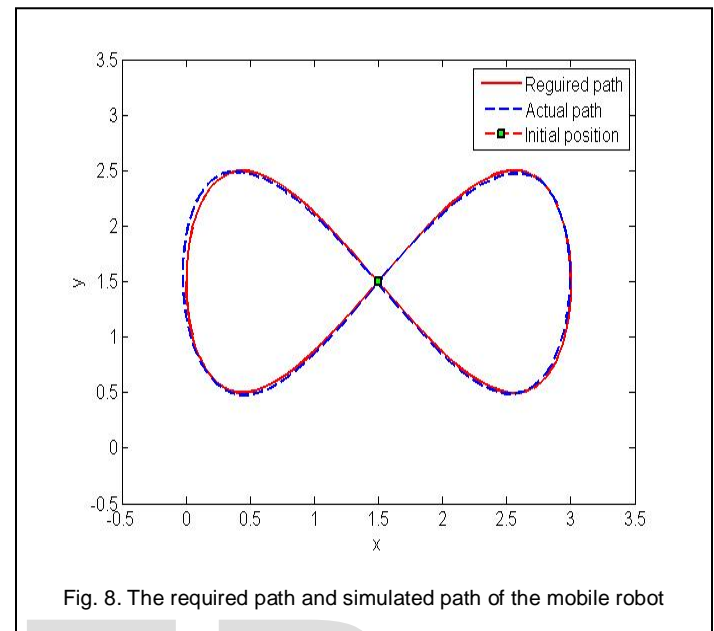


Fig. 8. The required path and simulated path of the mobile robot

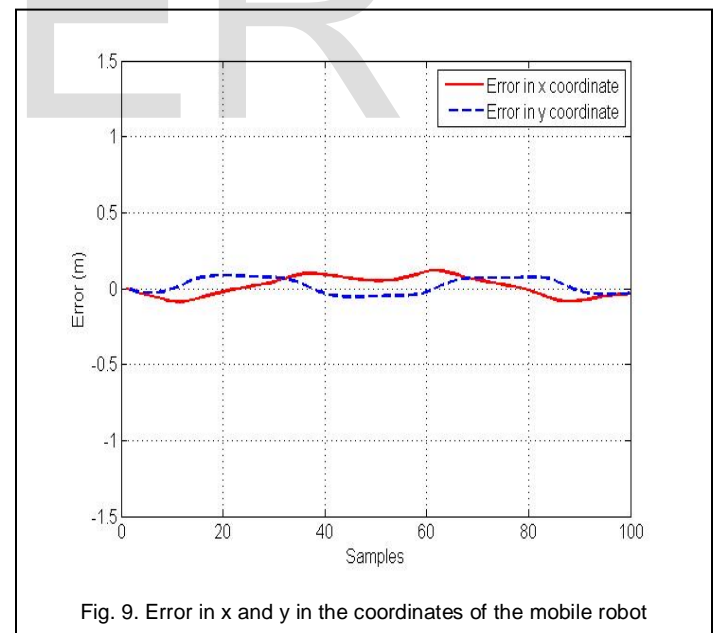


Fig. 9. Error in x and y in the coordinates of the mobile robot

The simulation results show that the controller functions correctly and prove its ability to guide the robot to follow this complex path with minimum error in the robot position and orientation.

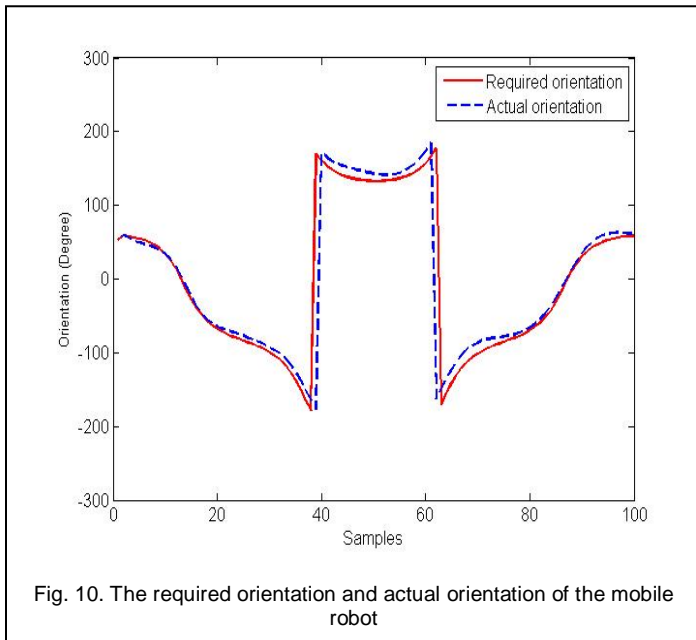


Fig. 10. The required orientation and actual orientation of the mobile robot

5 EXPERIMENTAL RESULTS

The experimental results were obtained by using the National Instrument mobile robot which is shown in fig. 11. This robot is compatible with LabVIEW2011 software and the controller is implemented in this software and then deployed to the mobile robot. The robot has two independent driving wheels, and one omnidirectional wheel which is added for the purpose of stability. The robot also has a powerful FPGA based processor with Real-time Input-Output (RIO) board.

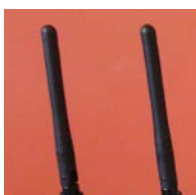


Fig. 11. National Instrument Mobile Robot

Figure 12 shows the required path and the actual (practical) path of the mobile robot. As shown in fig. 12 and fig. 13, the real mobile robot has followed the reference path with some

acceptable error in its position. The real orientation of the mobile robot is shown in fig. 14 and it can be seen that the robot is in the right orientation with some error because the robot is trying to correct the error in its position which causes an error in the orientation.

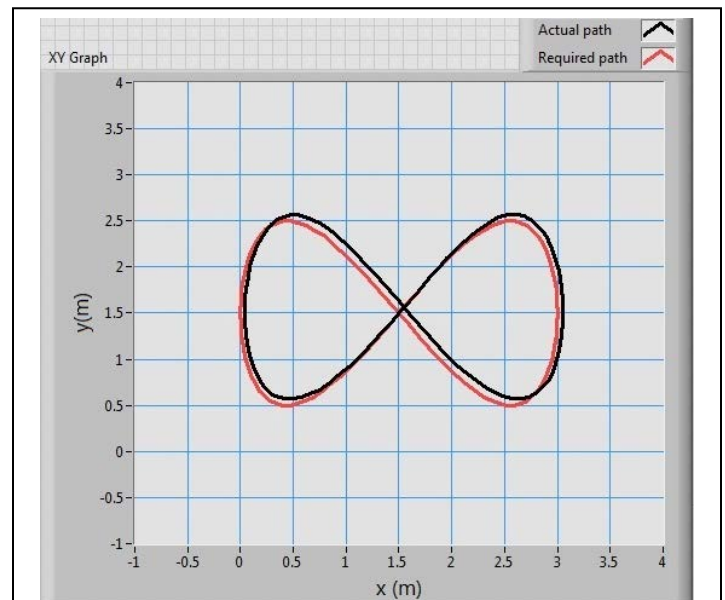


Fig. 12. The required path and actual path of the mobile robot (practical results)

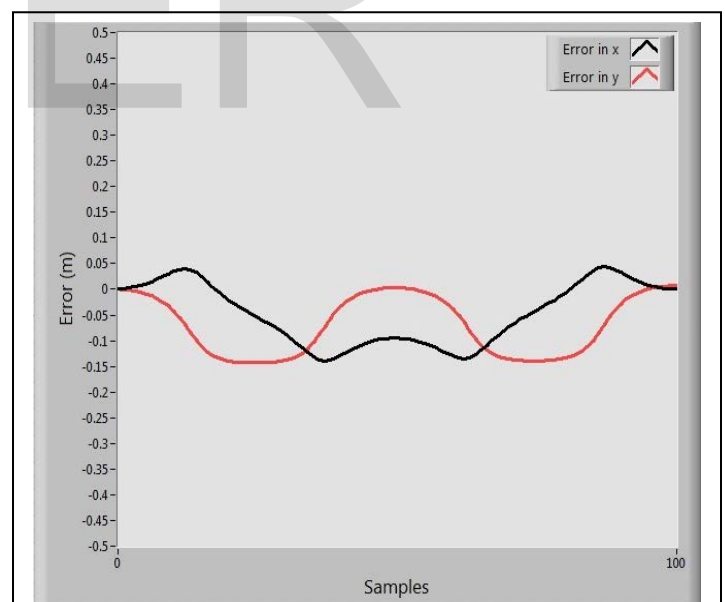


Fig. 13. Error in x and y coordinates of the mobile robot (practical results)

Figure 15 below shows the real mobile robot system while testing the functionality of the controller. The robots followed the predefined trajectory precisely and the controller was working successfully by guiding the robot with minimum error.

Iraq.

REFERENCES

- [1] M. T. Chew, S. Demidenko, L. Huang, C. Messom, G. SenGupta, and M. Watts, "Simple Mobile Robots for Introducing into Engineering," presented at the International Instrumentation and Measurement Technology Conference, Singapore, 2009.
- [2] Eva. B., J. A. Lopez-Orozco, P. Lanillos, and J. M. Cruz, "Localization of Non-Linearly Modeled Autonomous Mobile Robots Using Out-of-Sequence Measurements," *Sensors*, vol. 12, pp. 2487-2518, 23/02/2012.
- [3] O. Mohareri, Rached Dhaouadi, and Ahmad B. Rad, "Indirect adaptive tracking control of a nonholonomic mobile robot via neural networks," *Neurocomputing*, vol. 88, pp. 54 - 66 2012.
- [4] N. A. Martins, D. W. Bertol, W. C. Lombardi, and E. R. de Pieri, "Trajectory tracking of a nonholonomic mobile robot: A suggested neural torque controller based on the sliding mode theory," *International Conference on Variable Structure Systems*, pp. 384-389, 2008.
- [5] V. Ganapathy, S. C. Yun, and H. K. Joe, "Neural Q-Learning Controller for Mobile Robot," *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Vols 1-3, pp. 863-868, 2009.
- [6] P. K. Kim, J. H. Park, and S. Jung, "Experimental Studies of Balancing Control for a Disc-Typed Mobile Robot Using a Neural Controller: GYROBO," *IEEE International Symposium on Intelligent Control*, pp. 1499-1503, 2010.
- [7] Y.-T. Wang¹, Y.-C. Chen, and M.-C. Lin, "Dynamic Object Tracking Control for a Non-Holonomic Wheeled Autonomous Robot," *Tamkang Journal of Science and Engineering*, vol. 12, p. 339350, 2009.
- [8] S. Han, B. Choi, and J. Lee, "A precise curved motion planning for a differential driving mobile robot," *Mechatronics*, vol. 18, pp. 486-494, Nov 2008.
- [9] R. Siegwart and I. R. Nourbakhsh, "Introduction to autonomous mobile robots," Second ed.: MIT Press 2011.
- [10] J. Ye, "Tracking control for nonholonomic mobile robots: Integrating the analog neural network into the backstepping technique," *Neurocomputing*, vol. 71, pp. 3373-3378, 23 December 2008.
- [11] J. Zurada, "Introduction to Artificial Neural System": West Publishing Company, 1992.

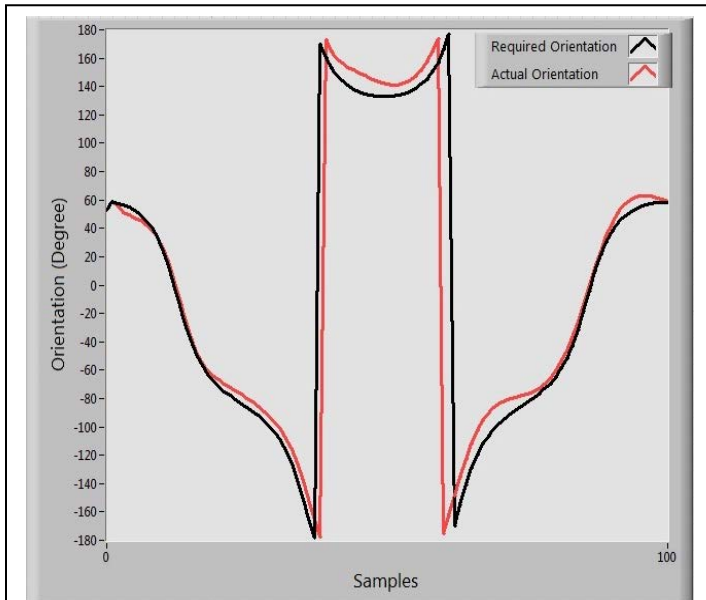


Fig. 14. The required path and actual orientation of the mobile robot (practical results)

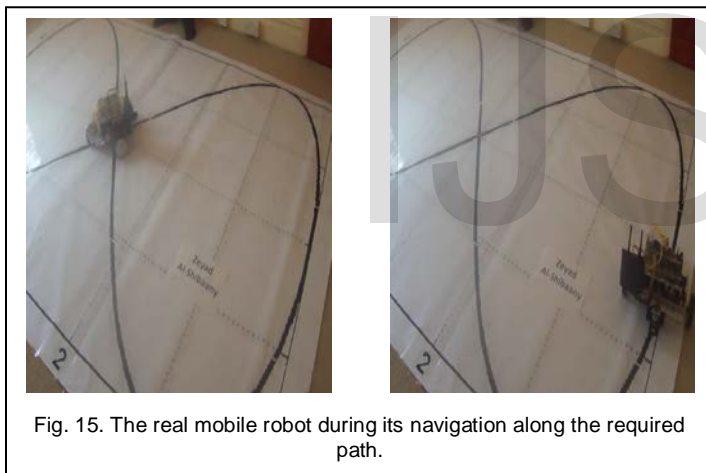


Fig. 15. The real mobile robot during its navigation along the required path.

6 CONCLUSION

The adaptive neural kinematic controller is proposed in this paper. The proposed controller has been designed and tested in MATLAB where the simulation results prove the functionality and robustness of the controller. The controller is implemented in LabVIEW software and applied to a real mobile robot where the practical results prove that the controller can successfully guide the robot during trajectory tracking.

ACKNOWLEDGMENT

This work would have not been possible without the great support of the Higher Committee for Education Development (HCED) in Iraq. The authors also would like to thank the Automation and Robotics Research Unit in the Control and Systems Engineering Department - University of Technology /