

AHMED S. AL-ARAJI et al: DESIGN OF AN ADAPTIVE NEURAL PREDICTIVE NONLINEAR . . .

Design of an Adaptive Neural Predictive Nonlinear Controller for Nonholonomic Mobile Robot System Based on Posture Identifier in the Presence of Disturbance

Ahmed S. Al-Araji, Maysam F. Abbod and Hamed S. Al-Raweshidy
Wireless Networks and Communication Centre, School of Engineering and Design
Brunel University
London - UK
ahmedsas2040@yahoo.com

Abstract —This paper proposes an adaptive neural predictive nonlinear controller to guide a nonholonomic wheeled mobile robot during continuous and non-continuous gradients trajectory tracking. The structure of the controller consists of two models that describe the kinematics and dynamics of the mobile robot system and a feedforward neural controller. The models are modified Elman neural network and feedforward multi-layer perceptron respectively. The modified Elman neural network model is trained off-line and on-line stages to guarantee the outputs of the model accurately represent the actual outputs of the mobile robot system. The trained neural model acts as the position and orientation identifier. The feedforward neural controller is trained off-line and adaptive weights are adapted on-line to find the reference torques, which controls the steady-state outputs of the mobile robot system. The feedback neural controller is based on the posture neural identifier and quadratic performance index optimization algorithm to find the optimal torque action in the transient state for N-step-ahead prediction. General back propagation algorithm is used to learn the feedforward neural controller and the posture neural identifier. Simulation results show the effectiveness of the proposed adaptive neural predictive control algorithm; this is demonstrated by the minimised tracking error and the smoothness of the torque control signal obtained with bounded external disturbances.

Keywords - *Nonholonomic Mobile Robots, Adaptive Predictive Nonlinear Controller, Neural Networks, Trajectory Tracking.*

I. INTRODUCTION

In recent years, wheel-based mobile robots have attracted considerable attention in various industrial and service applications. For example, room cleaning, factory automation, transportation, etc. These applications require mobile robots to have the ability to track specified path stably [1]. In general, nonholonomic behaviour in robotic systems is particularly interesting because this mechanism can completely be controlled with reduced number of actuators. Several controllers were proposed for trajectory tracking of mobile robots with nonholonomic constraints. The traditional control methods for mobile robot path tracking have used linear or non-linear feedback control while artificial intelligent controllers were carried out using neural networks or fuzzy inference [2].

There are other techniques for trajectory tracking controllers such as predictive control technique. Predictive approaches to path tracking seem to be very promising because the reference trajectory is known beforehand. Model predictive trajectory tracking control was applied to a mobile robot where linearised tracking error dynamics was used to predict future system behaviour and a control law was derived from a quadratic cost function penalizing the system tracking error and the control effort [3].

In addition, an adaptive trajectory-tracking controller based on the robot dynamics was proposed in [4 and 5] and its stability property was proved using the Lyapunov theory.

An adaptive controller of nonlinear PID-based neural networks was developed for the velocity and orientation tracking control of a nonholonomic mobile robot [6].

A trajectory tracking control for a nonholonomic mobile robot by the integration of a kinematics controller and neural dynamic controller based on the sliding mode theory was presented in [7]. The adaptive feedforward and feedback neural controllers with predictive optimization algorithm have minimised the tracking error of the nonholonomic wheeled mobile robot as presented in [8].

Two novel dual adaptive neural control schemes were proposed for dynamic control of nonholonomic mobile robots [9]. The first scheme was based on Gaussian radial basis function ANNs and the second on sigmoidal multilayer perceptron (MLP) ANNs. ANNs were employed for real-time approximation of the robot's nonlinear dynamic functions which were assumed to be unknown. Integrating the neural networks into back-stepping technique has improved learning algorithm of analogue compound orthogonal networks and novel tracking control approach for nonholonomic mobile robots [10]. A variable structure control algorithm was proposed to study the trajectory tracking control based on the kinematics model of a 2-wheel differentially driven mobile robot by using of the back stepping method and virtual feedback parameter with the sigmoid function [11]. The trajectory-tracking controllers designed by pole-assignment approach for mobile robot model were presented in [12].

The contribution of the presented approach is the analytically derived control law which has significantly high

computational accuracy with predictive optimization technique to obtain the optimal torques control action and lead to minimum tracking error of the mobile robot for different types of trajectories with continuous gradients such as (lemniscates) or non-continuous gradients (square) with bounded external disturbances.

The predictive optimization algorithm for N step ahead can generate excellent feedback control action in order to reduce the effect of external disturbances.

Simulation results show that the proposed controller is robust and effective in terms of fast response and minimum tracking error and in generating an optimal torque control action despite of the presence of bounded external disturbances.

The remainder of the paper is organized as follows. Section two is a description of the kinematics and dynamics model of the nonholonomic wheeled mobile robot. In section three, the proposed adaptive neural predictive controller is derived. The simulation results of the proposed controller are presented in section four and the conclusions are drawn in section five.

II. THE KINEMATICS AND DYNAMICS MODEL OF NONHOLONOMIC WHEELED MOBILE ROBOT

The schematic of the nonholonomic mobile robot, shown in figure 1, consists of a cart with two driving wheels mounted on the same axis and an omni-directional castor in the front of cart. The castor carries the mechanical structure and keeps the platform more stable [6 and 8]. Two independent analogous DC motors are the actuators of left and right wheels for motion and orientation. The two wheels have the same radius denoted by r , and L is the distance between the two wheels. The centre of mass of the mobile robot is located at point C , centre of axis of wheels.

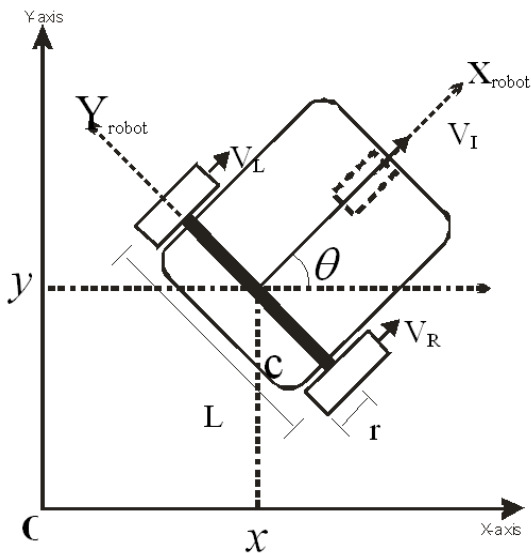


Figure 1. Schematic of the nonholonomic mobile robot.

The pose of mobile robot in the global coordinate frame $[0, x, y]$ and the pose vector in the surface is defined as:

$$q = (x, y, \theta)^T \quad (1)$$

where $q(t) \in \mathbb{R}^{3 \times 1}$,

x and y are coordinates of point C and θ is the robotic orientation angle measured with respect to the X-axis. These three generalized coordinates can describe the configuration of the mobile robot. The mobile robot is subjected to an independent velocity constraint that can be expressed in matrix form [13]:

$$A^T(q)\dot{q} = 0 \quad (2)$$

where

$$A^T(q) = [-\sin \theta(t) \quad \cos \theta(t) \quad 0] \quad (3)$$

$$A(q) \in \mathbb{R}^{3 \times 1}$$

It is assumed that the mobile robot wheels are ideally installed in such a way that they have ideal rolling without skidding [14].

Therefore, the kinematics of the robot can be described as

$$\dot{q} = \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} \cos \theta(t) & 0 \\ \sin \theta(t) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V_l(t) \\ V_w(t) \end{bmatrix} \quad (4)$$

where $S(q)$ is defining a full rank matrix as

$$S(q) = \begin{bmatrix} \cos \theta(t) & 0 \\ \sin \theta(t) & 0 \\ 0 & 1 \end{bmatrix} \quad (5)$$

where V_l and V_w , the linear and angular velocities.

Forces must be applied to the mobile robot to produce motion. These forces are modeled by studying the motion of the dynamic model of the differential wheeled mobile robot shown in figure 1. Mass, forces and speed are associated with this motion. The dynamic model can be described by the following form of dynamic equations based on Euler Lagrange formulation [5, 6, 8 and 9].

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + \tau d = B(q)\tau - A^T(q)\lambda \quad (6)$$

$M(q) \in \mathbb{R}^{3 \times 3}$ is a symmetric positive definite inertia matrix,

$C(q, \dot{q}) \in \mathbb{R}^{3 \times 3}$ is the centripetal and coriolis matrix,

$G(q) \in \mathbb{R}^{3 \times 1}$ is the gravitational torques vector, $\tau d \in \mathbb{R}^{3 \times 1}$ denotes bounded unknown disturbances including unstructured and unmodeled dynamics, $B(q) \in \mathbb{R}^{3 \times 2}$ is the input transformation matrix, $\tau \in \mathbb{R}^{2 \times 1}$ is input torque vector, and $\lambda \in \mathbb{R}^{1 \times 1}$ is the vector of constraint forces.

Remark 1: The plane of each wheel is perpendicular to the ground and the contact between the wheels and the ground is pure rolling and non-slipping, and hence the velocity of the centre of the mass of the mobile robot is orthogonal to the rear wheels' axis.

Remark 2: The trajectory of mobile robot base is constrained to the horizontal plane, therefore, $G(q)$ is equal to zero.

Remark 3: In this dynamic model, the passive self-adjusted supporting wheel influence is not taken into

consideration as it is a free wheel. This significantly reduces the complexity of the model for the feedback controller design. However, the free wheel may be a source of substantial distortion, particularly in the case of changing its movement direction. This effect is reduced if the small velocity of the robot is considered [5 and 6]. Remark 4: The centre of mass for mobile robot is located in the middle of axis connecting the rear wheels in C point as shown in figure 1, therefore, $C(q, \dot{q})$ is equal to zero.

The dynamical equation of the differential wheeled mobile robot can be expressed as

$$\begin{bmatrix} M & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\theta} \end{bmatrix} + \tau d = \frac{1}{r} \begin{bmatrix} \cos \theta & \cos \theta \\ \sin \theta & \sin \theta \\ \frac{L}{2} & -\frac{L}{2} \end{bmatrix} \begin{bmatrix} \tau_L \\ \tau_R \end{bmatrix} + \begin{bmatrix} -\sin \theta \\ \cos \theta \\ 0 \end{bmatrix} \lambda \quad (7)$$

where τ_L and τ_R are the torques of left and right motors respectively. M and I present the mass and inertia of the mobile robot respectively.

By solving equation (4 and 7) then we can reach the normal form,

$$\dot{V}_I = \frac{\tau_L + \tau_R}{Mr} + \tau d \quad (8)$$

$$\dot{V}_W = \frac{L(\tau_L - \tau_R)}{2rI} + \tau d \quad (9)$$

where \dot{V}_I and \dot{V}_W are the linear and angular acceleration of the differential wheeled mobile robot.

The dynamics and the kinematics model structure of the differential wheeled mobile robot can be shown in figure 2.

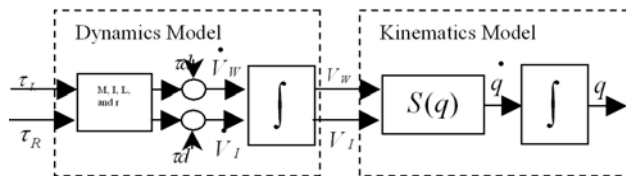


Figure 2. Dynamics and kinematics model structure of the mobile robot.

III. ADAPTIVE NEURAL PREDICTIVE CONTROL METHODOLOGY

The control of nonlinear MIMO mobile robot system is considered in this section. The approach to control the mobile robot depends on the available information of the unknown nonlinear system can be known by the input-output data only and the control objectives. The first step in the procedure of the control structure is the identification of the kinematics and dynamics mobile robot from the input-output data. Then an adaptive feedforward neural controller is designed to find reference torques that control the steady-state outputs of the mobile robot trajectory.

The feedback neural controller is based on the minimisation of a quadratic performance index function of

the error between the desired trajectory input and the posture neural identifier output, i.e. position and orientation of mobile robot trajectory, and the feedback neural controller itself. The predictive optimization algorithm is used to determine the torque control signal for N-steps-ahead and to use minimum torque effort. The torque control signal will minimise the cost function in order to minimise the tracking error as well as reduce the torque control effort in the presence of external disturbance. The integrated adaptive control structure, which consists of an adaptive feedforward neural controller and feedback neural controller with an optimization algorithm, brings together the advantages of the adaptive neural method with the robustness of feedback for N-step-ahead prediction.

The proposed structure of the adaptive neural predictive controller can be given in the form of block diagram as shown in figure 3. It consists of:

- Position and Orientation Neural Networks Identifier.
- Feedforward Neural Controller.
- Feedback Neural Controller.

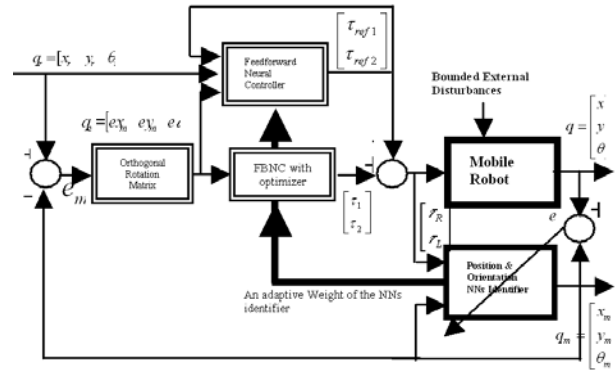


Figure 3. The proposed structure of the adaptive neural predictive controller for the nonholonomic wheeled mobile robot.

A. Position and Orientation Neural Networks Identifier

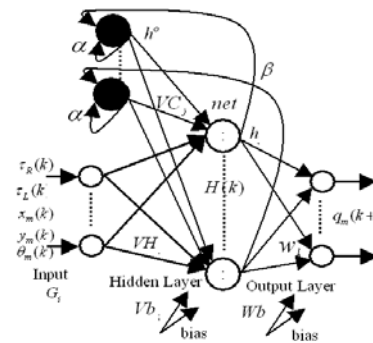


Figure 4 . Elman neural networks acts as the posture identifier.

Nonlinear MIMO system identification of kinematics and dynamics mobile robot, position and orientation, will be introduced in this section. The modified Elman recurrent neural network model is applied to construct the position and orientation neural network identifier as shown in figure

4. The nodes of input, context, hidden and output layers are highlighted. The network uses two configuration models, series-parallel and parallel identification structures, which are trained using dynamic back-propagation algorithm.

The structure shown in figure 4 is based on the following equations [15]:

$$h(k) = F\{VH\overline{G}(k), VC\overline{h}^o(k), bias\overline{Vb}\} \quad (10)$$

$$O(k) = (Wh(k), bias\overline{Wb}) \quad (11)$$

where VH, VC and W are weight matrices, \overline{Vb} and \overline{Wb} are weight vectors and F is a non-linear vector function. The multi-layered modified Elman neural network, shown in figure 4, is composed of many interconnected processing units called neurons or nodes.

The output of the context unit in the modified Elman network is given by [15]:

$$h_c^o(k) = \alpha h_c^o(k-1) + \beta h_c(k-1) \quad (12)$$

where $h_c^o(k)$ and $h_c(k)$ are the outputs of the context and hidden units respectively. α is the feedback gain of the self-connections and β is the connection weight from the hidden units (j^{th}) to the context units (c^{th}) at the context layer. The value of α and β are selected randomly between (0 and 1) [15].

The outputs of the identifier are the modelling pose vector in the surface and are defined as:

$q_m = (x_m, y_m, \theta_m)^T$, where x_m and y_m are the modelling coordinates and θ_m is the modelling orientation angle.

The learning algorithm will be used to adjust the weights of dynamical recurrent neural network. Dynamic back propagation algorithm is used to train the Elman network. The sum of the square of the differences between the desired outputs $q = (x, y, \theta)^T$ and neural network identifier outputs $q_m = (x_m, y_m, \theta_m)^T$ is given by equation (13).

$$E = \frac{1}{2} \sum_{i=1}^{np} ((x - x_m)^2 + (y - y_m)^2 + (\theta - \theta_m)^2) \quad (13)$$

where np is the number of patterns.

The connection matrix between hidden layer and output layer is W_{kj}

$$\Delta W_{kj}(k+1) = -\eta \frac{\partial E}{\partial W_{kj}} \quad (14)$$

where η is learning rate.

$$\frac{\partial E}{\partial W_{kj}} = \frac{\partial E}{\partial q_m(k+1)} \frac{\partial q_m(k+1)}{\partial o_k} \frac{\partial o_k}{\partial net_k} \frac{\partial net_k}{\partial W_{kj}} \quad (15)$$

$$W_{kj}(k+1) = W_{kj}(k) + \Delta W_{kj}(k+1) \quad (16)$$

The connection matrix between input layer and hidden layer is VH_{ji}

$$\Delta VH_{ji}(k+1) = -\eta \frac{\partial E}{\partial VH_{ji}} \quad (17)$$

$$\frac{\partial E}{\partial VH_{ji}} = \frac{\partial E}{\partial q_m(k+1)} \frac{\partial q_m(k+1)}{\partial o_k} \frac{\partial o_k}{\partial net_k} \frac{\partial net_k}{\partial h_j} \frac{\partial h_j}{\partial net_j} \frac{\partial net_j}{\partial VH_{ji}} \quad (18)$$

$$VH_{ji}(k+1) = VH_{ji}(k) + \Delta VH_{ji}(k+1) \quad (19)$$

The connection matrix between context layer and hidden layer is VC_{ji}

$$\Delta VC_{jc}(k+1) = -\eta \frac{\partial E}{\partial VC_{jc}} \quad (20)$$

$$\frac{\partial E}{\partial VC_{jc}} = \frac{\partial E}{\partial q_m(k+1)} \frac{\partial q_m(k+1)}{\partial o_k} \frac{\partial o_k}{\partial net_k} \frac{\partial net_k}{\partial h_j} \frac{\partial h_j}{\partial net_c} \frac{\partial net_c}{\partial VC_{jc}} \quad (21)$$

$$VC_{jc}(k+1) = VC_{jc}(k) + \Delta VC_{jc}(k+1) \quad (22)$$

B. Feedforward Neural Controller

The feedforward neural controller (FFNC) is of prime importance in the structure of the controller due to its necessity in keeping the steady-state tracking error at zero. This means that the actions of the FFNC, $\tau_{ref1}(k)$ and $\tau_{ref2}(k)$ are used as the reference torques of the steady state outputs of the mobile robot. Hence, the FFNC is supposed to learn the adaptive inverse model of the mobile robot system with off-line and on-line stages to calculate mobile robot's reference input torques drive. Reference input torques will keep the robot on a desired trajectory in the presence of any disturbances or initial state errors. To achieve FFNC, a multi-layer perceptron model is used as shown in figure 5 [16].

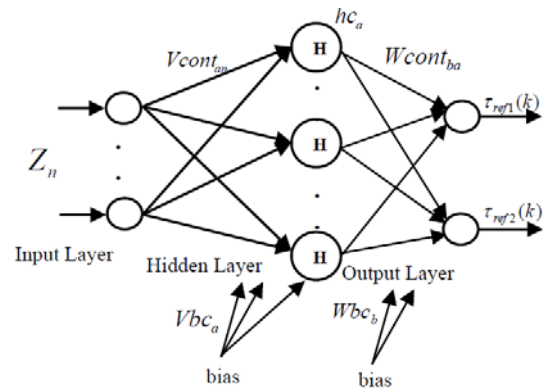


Figure 5. MLP Neural network acts as the feed forward neural controller

The training of the feedforward neural controller is performed off-line as shown in figure 6, which the weights adapted on-line. It depends on the posture neural network identifier to find the mobile robot Jacobian through the neural identifier model.

This approach is currently considered as one of the better approaches that can be followed to overcome the lack of initial knowledge. The dynamic back propagation algorithm is employed to realize the training the weights of the feedforward neural controller.

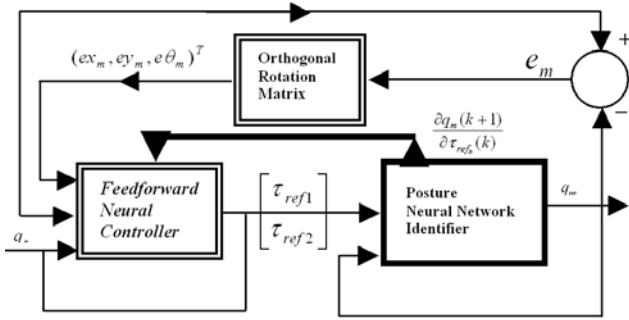


Figure 6. The feedforward neural controller structure for mobile robot model.

The sum of the square of the differences between the desired posture $q_r = (x_r, y_r, \theta_r)^T$ and neural network posture $q_m = (x_m, y_m, \theta_m)^T$ is:

$$Ec = \frac{1}{2} \sum_{i=1}^{npc} ((x_r - x_m)^2 + (y_r - y_m)^2 + (\theta_r - \theta_m)^2) \quad (23)$$

where npc is number of patterns.

The connection matrix between hidden layer and output layer is $Wcont_{ba}$

$$\Delta Wcont_{ba}(k+1) = -\eta \frac{\partial Ec}{\partial Wcont_{ba}} \quad (24)$$

$$\frac{\partial Ec}{\partial Wcont_{ba}} = \frac{\partial Ec}{\partial q_m(k+1)} \frac{\partial q_m(k+1)}{\partial \tau_{ref1}(k)} \frac{\partial \tau_{ref1}(k)}{\partial oc_b} \frac{\partial oc_b}{\partial netc_b} \frac{\partial netc_b}{\partial Wcont_{ba}} \quad (25)$$

$$\frac{\partial Ec}{\partial q_m(k+1)} = \frac{\frac{1}{2} \sum ((x_r - x_m)^2 + (y_r - y_m)^2 + (\theta_r - \theta_m)^2)}{\partial q_m(k+1)} \quad (26)$$

This is achieved in the local coordinates with respect to the body of the mobile robot that is the same outputs of the position and orientation neural networks identifier. The configuration error can be represented by using a transformation matrix as:

$$\begin{bmatrix} ex_m \\ ey_m \\ e\theta_m \end{bmatrix} = \begin{bmatrix} \cos \theta_m & \sin \theta_m & 0 \\ -\sin \theta_m & \cos \theta_m & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x_m \\ y_r - y_m \\ \theta_r - \theta_m \end{bmatrix} \quad (27)$$

where x_r , y_r and θ_r are the reference posture of the mobile robot.

$$Jacobian = \frac{\partial q_m(k+1)}{\partial \tau_{ref1}(k)} \quad (28)$$

where the outputs of the identifier are $q_m = (x_m, y_m, \theta_m)^T$.

$$\frac{\partial q_m(k+1)}{\partial \tau_{ref1}(k)} = \frac{\partial q_m(k+1)}{\partial oc_k(k)} \frac{\partial oc_k(k)}{\partial netc_k} \frac{\partial h_j}{\partial netc_j} \frac{\partial netc_j}{\partial \tau_{ref1}(k)} \quad (29)$$

Substituting equations (26 and 29) into equation (25), to find $\Delta Wcont_{ba}(k+1)$, then

$$Wcont_{ba}(k+1) = Wcont_{ba}(k) + \Delta Wcont_{ba}(k+1) \quad (30)$$

The connection matrix between input layer and hidden layer is $Vcont_{an}$

$$\Delta Vcont_{an}(k+1) = -\eta \frac{\partial Ec}{\partial Vcont_{an}} \quad (31)$$

$$\frac{\partial Ec}{\partial Vcont_{an}} = \frac{\partial Ec}{\partial q_m(k+1)} \frac{\partial q_m(k+1)}{\partial \tau_{ref1}(k)} \frac{\partial \tau_{ref1}(k)}{\partial oc_b} \times$$

$$\frac{\partial oc_b}{\partial netc_b} \times \frac{\partial netc_b}{\partial hc_a} \times \frac{\partial hc_a}{\partial netc_a} \times \frac{\partial netc_a}{\partial Vcont_{an}} \quad (32)$$

Substituting equations (26 and 29) into equation (32), to find $\Delta Vcont_{an}(k+1)$, then

$$Vcont_{an}(k+1) = Vcont_{an}(k) + \Delta Vcont_{an}(k+1) \quad (33)$$

Once the feedforward neural controller has learned, it generates the torque control action to keep the output of the mobile robot at the steady state reference value and to overcome any external disturbances during trajectory.

The torques will be known equivalently as τ_{ref1} and τ_{ref2} , the reference torques of the right and left wheels respectively.

C. Feedback Neural Controller

The feedback neural controller is essential to stabilise the tracking error of the mobile robot system when the trajectory of the robot is drifted from the reference trajectory during transient state. The feedback neural controller generates an optimal torque control action that minimises the cumulative error between the reference input trajectory and the output trajectory of the mobile robot. The weighted sum of the torque control signal can be obtained by minimising a quadratic performance index. The feedback neural controller consists of the adaptive weights of the position and orientation neural networks identifier and an optimization algorithm. The quadratic performance index for multi input /multi output system can be expressed as:

$$J = \frac{1}{2} \sum_{k=1}^N Q(q_r(k+1) - q(k+1))^2 + R((\tau_{ref1}(k) - \tau_R(k))^2 + (\tau_{ref2}(k) - \tau_L(k))^2) \quad (34)$$

Hence

$$q_r(k+1) = [x_r(k+1), y_r(k+1), \theta_r(k+1)]^T \quad (35)$$

$$q(k+1) = [x(k+1), y(k+1), \theta(k+1)]^T \quad (36)$$

$$\tau_R(k) = \tau_{ref1}(k) + \tau_1(k) \quad (37)$$

$$\tau_L(k) = \tau_{ref2}(k) + \tau_2(k) \quad (38)$$

(Q, R) are positive weighting factors.

N is the number of steps ahead.

The quadratic cost function will not only force the mobile robot output to follow the reference trajectory by minimising the cumulative error for N steps ahead but also forces the torque control actions ($\tau_1(k)$ and $\tau_2(k)$) in the transient period to be as close as possible to the reference torque control signals ($\tau_{ref1}(k)$ and $\tau_{ref2}(k)$). In addition, J depends on Q & R factors and chooses a set of values of the weighting factors Q and R to determine the optimal control action by observing the system behavior [17]. The on-line position and orientation neural networks identifier is to be used to obtain the predicted values of the outputs of the mobile robot system $q_m(k+1)$ for N steps ahead instead of running the mobile robot system itself $q(k+1)$ for N steps. This is performed to find the optimal torque control actions by using the posture identifier weights and optimization algorithm depending on the quadratic cost function. Therefore, it can be said that:

$$q_m(k+1) \approx q(k+1) \quad (39)$$

The performance index of equation (34) can be put as:

$$J = \frac{1}{2} \sum_{k=1}^N Q((x_r(k+1) - x_m(k+1))^2 + (y_r(k+1) - y_m(k+1))^2 + (\theta_r(k+1) - \theta_m(k+1))^2) + R((\tau_1(k))^2 + (\tau_2(k))^2) \quad (40)$$

To achieve equation (39 and 40), the modified Elman neural network will be used as posture identifier. This task is carried out using an identification technique based on series-parallel and parallel configuration with two stages to learn the posture identifier. The first stage is an off-line identification, while the second stage is an on-line modification of the weights of the obtained position and orientation neural identifier. The on-line modifications are necessary to keep tracking any possible variation in the kinematics and dynamics parameters of the mobile robot system.

Back propagation algorithm (BPA) is used to adjust the weights of the posture neural identifier to learn the kinematics and dynamics of the mobile robot system, by applying a simple gradient decent rule.

For N steps estimation of the two feedback neural controller actions $\tau_1(k)$ & $\tau_2(k)$ the techniques of generalized predictive control theory will be used. The N steps estimation of $\tau_1(k)$ & $\tau_2(k)$ will be calculated for each sample. The position and orientation in the identifier model, shown in figure 4, represent the kinematics and dynamics model of the mobile robot system and will be controlled asymptotically. Therefore, they can be used to predict future values of the model outputs for the next N steps and can be used to find the optimal value of $\tau_1(k)$ & $\tau_2(k)$ using an optimization algorithm.

For this purpose, let N be a pre-specified positive integer that is denoted such that the future values of the set point are:

$$X_{r,t,N} = [x_r(t+1), x_r(t+2), x_r(t+3), \dots, x_r(t+N)] \quad (41)$$

$$Y_{r,t,N} = [y_r(t+1), y_r(t+2), y_r(t+3), \dots, y_r(t+N)] \quad (42)$$

$$\theta_{r,t,N} = [\theta_r(t+1), \theta_r(t+2), \theta_r(t+3), \dots, \theta_r(t+N)] \quad (43)$$

As the future values of set point and (t) represents the time instant, and the predicted outputs of the robot model used the neural identifier, shown in figure 4, are:

$$X_{m,t,N} = [x_m(t+1), x_m(t+2), x_m(t+3), \dots, x_m(t+N)] \quad (44)$$

$$Y_{m,t,N} = [y_m(t+1), y_m(t+2), y_m(t+3), \dots, y_m(t+N)] \quad (45)$$

$$\theta_{m,t,N} = [\theta_m(t+1), \theta_m(t+2), \theta_m(t+3), \dots, \theta_m(t+N)] \quad (46)$$

The error vector of position and orientation as equations (47, 48, and 49) can be calculated by using equation (27).

$$EX_{m,t,N} = [ex_m(t+1), ex_m(t+2), ex_m(t+3), \dots, ex_m(t+N)] \quad (47)$$

$$EY_{m,t,N} = [ey_m(t+1), ey_m(t+2), ey_m(t+3), \dots, ey_m(t+N)] \quad (48)$$

$$E\theta_{m,t,N} = [e\theta_m(t+1), e\theta_m(t+2), e\theta_m(t+3), \dots, e\theta_m(t+N)] \quad (49)$$

Two-feedback control signals can be determined by:

$$\tau'_{1,t,N} = [\tau'_1(t), \tau'_1(t+1), \tau'_1(t+2), \dots, \tau'_1(t+N-1)] \quad (50)$$

$$\tau'_{2,t,N} = [\tau'_2(t), \tau'_2(t+1), \tau'_2(t+2), \dots, \tau'_2(t+N-1)] \quad (51)$$

Assuming the following objective function:

$$J1 = \frac{1}{2} Q[(EX_{m,t,N} EX_{m,t,N}^T) + (EY_{m,t,N} EY_{m,t,N}^T) + (E\theta_{m,t,N} E\theta_{m,t,N}^T)] + \frac{1}{2} R[(\tau'_{1,t,N} \tau'_{1,t,N}^T) + (\tau'_{2,t,N} \tau'_{2,t,N}^T)] \quad (52)$$

then it is aimed to find τ'_1 and τ'_2 such that J1 is minimised using the gradient descent rule. The new control actions will be given by:

$$\tau'_{1,t,N}^{K+1} = \tau'_{1,t,N}^K + \Delta \tau'_{1,t,N}^K \quad (53)$$

$$\tau'_{2,t,N}^{K+1} = \tau'_{2,t,N}^K + \Delta \tau'_{2,t,N}^K \quad (54)$$

where k here indicates that calculations are performed at the k^{th} sample; and

$$\Delta \tau'_{1,t,N}^K = -\eta \frac{\partial J1}{\partial \tau'_{1,t,N}^K} = [\Delta \tau'_1(t), \Delta \tau'_1(t+1), \Delta \tau'_1(t+2), \dots, \Delta \tau'_1(t+N-1)] \quad (55)$$

$$\Delta \tau'_{2,t,N}^K = -\eta \frac{\partial J1}{\partial \tau'_{2,t,N}^K} = [\Delta \tau'_2(t), \Delta \tau'_2(t+1), \Delta \tau'_2(t+2), \dots, \Delta \tau'_2(t+N-1)] \quad (56)$$

$$-\eta \frac{\partial J1}{\partial \tau'_{1,t,N}^K} = \eta QEX_{m,t,N} \frac{\partial X_{m,t,N}}{\partial \tau'_{1,t,N}^K} + \eta QEY_{m,t,N} \frac{\partial Y_{m,t,N}}{\partial \tau'_{1,t,N}^K} + \eta QE\theta_{m,t,N} \frac{\partial \theta_{m,t,N}}{\partial \tau'_{1,t,N}^K} - \eta R \partial \tau'_{1,t,N}^K \quad (57)$$

$$-\eta \frac{\partial J1}{\partial \tau'_{2,t,N}^K} = \eta QEX_{m,t,N} \frac{\partial X_{m,t,N}}{\partial \tau'_{2,t,N}^K} + \eta QEY_{m,t,N} \frac{\partial Y_{m,t,N}}{\partial \tau'_{2,t,N}^K} + \eta QE\theta_{m,t,N} \frac{\partial \theta_{m,t,N}}{\partial \tau'_{2,t,N}^K} - \eta R \partial \tau'_{2,t,N}^K \quad (58)$$

Equations (59 to 64) are the well-known Jacobian vectors.

$$\frac{\partial X_{m,t,N}}{\partial \tau'_{1,t,N}^K} = \begin{bmatrix} \frac{\partial x_m(t+1)}{\partial \tau'_1(t)} & \frac{\partial x_m(t+2)}{\partial \tau'_1(t+1)} & \dots & \frac{\partial x_m(t+N)}{\partial \tau'_1(t+N-1)} \end{bmatrix} \quad (59)$$

$$\frac{\partial X_{m,t,N}}{\partial \tau'_{2,t,N}^K} = \begin{bmatrix} \frac{\partial x_m(t+1)}{\partial \tau'_2(t)} & \frac{\partial x_m(t+2)}{\partial \tau'_2(t+1)} & \dots & \frac{\partial x_m(t+N)}{\partial \tau'_2(t+N-1)} \end{bmatrix} \quad (60)$$

$$\frac{\partial Y_{m,t,N}}{\partial \tau'_{1,t,N}^K} = \begin{bmatrix} \frac{\partial y_m(t+1)}{\partial \tau'_1(t)} & \frac{\partial y_m(t+2)}{\partial \tau'_1(t+1)} & \dots & \frac{\partial y_m(t+N)}{\partial \tau'_1(t+N-1)} \end{bmatrix} \quad (61)$$

$$\frac{\partial Y_{m,t,N}}{\partial \tau'_{2,t,N}^K} = \begin{bmatrix} \frac{\partial y_m(t+1)}{\partial \tau'_2(t)} & \frac{\partial y_m(t+2)}{\partial \tau'_2(t+1)} & \dots & \frac{\partial y_m(t+N)}{\partial \tau'_2(t+N-1)} \end{bmatrix} \quad (62)$$

$$\frac{\partial \theta_{m,t,N}}{\partial \tau'_{1,t,N}^K} = \begin{bmatrix} \frac{\partial \theta_m(t+1)}{\partial \tau'_1(t)} & \frac{\partial \theta_m(t+2)}{\partial \tau'_1(t+1)} & \dots & \frac{\partial \theta_m(t+N)}{\partial \tau'_1(t+N-1)} \end{bmatrix} \quad (63)$$

$$\frac{\partial \theta_{m,t,N}}{\partial \tau'_{2,t,N}^K} = \begin{bmatrix} \frac{\partial \theta_m(t+1)}{\partial \tau'_2(t)} & \frac{\partial \theta_m(t+2)}{\partial \tau'_2(t+1)} & \dots & \frac{\partial \theta_m(t+N)}{\partial \tau'_2(t+N-1)} \end{bmatrix} \quad (64)$$

It can be seen that each element in the above vectors can be calculated from equation (65 to 74) such that:

$$net_j = \sum_{i=1}^{nh} V H_{ji} \times G_i + \sum_{c=1}^C V C_{jc} \times h_c^o + bias \times V b_j \quad (65)$$

where $j=c$ and $nh=C$ are the number of the hidden and context nodes respectively and \bar{G} is the input vector such as

$$G = [\tau_R(t), \tau_L(t), x_m(t), y_m(t), \theta_m(t)]^T \quad (66)$$

$$h_j = \frac{2}{1 + e^{-net_j}} - 1 \quad (67)$$

$$f(net_j) = 0.5(1 - h_j^2) \quad (68)$$

From figure 4 shows that $\tau_R(k)$ is linked to the exciting nodes, VH_{j1} and $\tau_L(k)$ is linked to the exciting nodes VH_{j2} then can be calculated Jacobian vectors.

$$\frac{\partial x_m(t+1)}{\partial \tau'_1(t)} = \sum_{j=1}^{nh} W_{1j} f(net_j)' VH_{j1} \quad (69)$$

$$\frac{\partial y_m(t+1)}{\partial \tau'_1(t)} = \sum_{j=1}^{nh} W_{2j} f(net_j)' VH_{j1} \quad (70)$$

$$\frac{\partial \theta_m(t+1)}{\partial \tau'_1(t)} = \sum_{j=1}^{nh} W_{3j} f(net_j)' VH_{j1} \quad (71)$$

$$\frac{\partial x_m(t+1)}{\partial \tau'_2(t)} = \sum_{j=1}^{nh} W_{1j} f(net_j)' VH_{j2} \quad (72)$$

$$\frac{\partial y_m(t+1)}{\partial \tau'_2(t)} = \sum_{j=1}^{nh} W_{2j} f(net_j)' VH_{j2} \quad (73)$$

$$\frac{\partial \theta_m(t+1)}{\partial \tau'_2(t)} = \sum_{j=1}^{nh} W_{3j} f(net_j)' VH_{j2} \quad (74)$$

Therefore, recursive methods for calculating the Jacobian vectors are developed so that the algorithm can be applied to real-time systems. After completing the procedure from $n=1$ to N the new control actions for the next sample will be:

$$\tau_R(k+1) = \tau_{ref1}(k+1) + \tau'_1(t+N) \quad (75)$$

$$\tau_L(k+1) = \tau_{ref2}(k+1) + \tau'_2(t+N) \quad (76)$$

where $\tau'_1(t+N)$ & $\tau'_2(t+N)$ are the final values of the feedback-controlling signals calculated by the optimization algorithm. This is calculated at each sample time k so that $\tau_R(k+1)$ & $\tau_L(k+1)$ are torque control actions of the right and the left wheels respectively. These actions will be applied to the mobile robot system and the position and orientation identifier model at the next sampling time. The application of this procedure will continue at the next sampling time $(k+1)$ until the error between the desired input and the actual output becomes lower than a pre-specified value.

IV. SIMULATION RESULTS

The proposed controller is verified by means of computer simulation using MATLAB/SIMULINK. The kinematics and dynamics model of the nonholonomic mobile robot described in section 2 are used. The simulation is carried out by tracking a desired position (x, y) and orientation angle (θ) with a lemniscates and square trajectories in the tracking control of the robot. The parameter values of the robot model are taken from [18]: $M=0.65\text{kg}$, $I=0.36\text{kgm}^2$, $L=0.105\text{m}$ and $r=0.033\text{m}$.

A hybrid excitation signal has been used for the robot model. Figure 7 shows the input signals $\tau_R(k)$ and $\tau_L(k)$, right and left wheel torques respectively. The training set is generated by feeding a PRBS signals, with sampling time of 0.5 second, to the model and measuring its corresponding outputs, position x and y and orientation θ .

The proposed controller is implemented based on the structure shown in figure 3. The first stage of operation is to set the position and orientation neural network identifier. This task is performed using series-parallel and parallel identification technique configuration with modified Elman

recurrent neural networks model. The identification scheme of the nonlinear MIMO mobile robot system are needed to input-output training data pattern to provide enough information about the kinematics and dynamics mobile robot model to be modelled. This can be achieved by injecting a sufficiently rich input signal to excite all process modes of interest while also ensuring that the training patterns adequately covers the specified operating region. Back propagation learning algorithm is used with the modified Elman recurrent neural network of the structure (5-6-6-3). The number of nodes in the input, hidden, context and output layers are 5, 6, 6 and 3 respectively as shown in figure 4

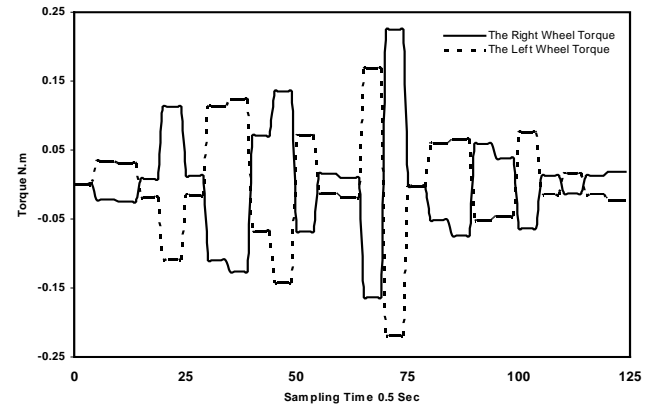


Figure 7. The PRBS input torque signals used to excite the mobile robot model.

A training set of 125 patterns has been used with a learning rate of 0.1. After 3244 epochs, the identifier outputs of the neural network, position x , y and orientation θ , are approximated to the actual outputs of the model trajectory as shown in figure 8.

Parallel configuration is used to guarantee the similarity between the outputs of the neural network identifier and the actual outputs of the mobile robot model trajectory. At 3538 the same training set patterns has been achieved with a mean square error less than 5.7×10^{-6} . The neural network identifier position and orientation outputs and the mobile robot model trajectory are shown in figure 9.

A. Case Study-1

The desired lemniscates trajectory which has explicitly continuous gradient with rotation radius changes, this trajectory can be described by the following equations:

$$x_r(t) = 0.75 + 0.75 \times \sin\left(\frac{2\pi t}{50}\right) \quad (77)$$

$$y_r(t) = \sin\left(\frac{4\pi t}{50}\right) \quad (78)$$

$$\theta_r(t) = 2 \tan^{-1}\left(\frac{\Delta y_r(t)}{\sqrt{(\Delta x_r(t))^2 - (\Delta y_r(t))^2 + \Delta x_r(t)}}\right) \quad (79)$$

The second stage of the proposed controller is feedforward neural controller. It uses multi-layer perceptron

neural network (8-11-2) as shown in figure 5. The trajectory has been learned by the feedforward neural controller with off-line and on-line adaptation stages using back propagation algorithm as shown in figure 6 to find the suitable reference torque control action at steady state.

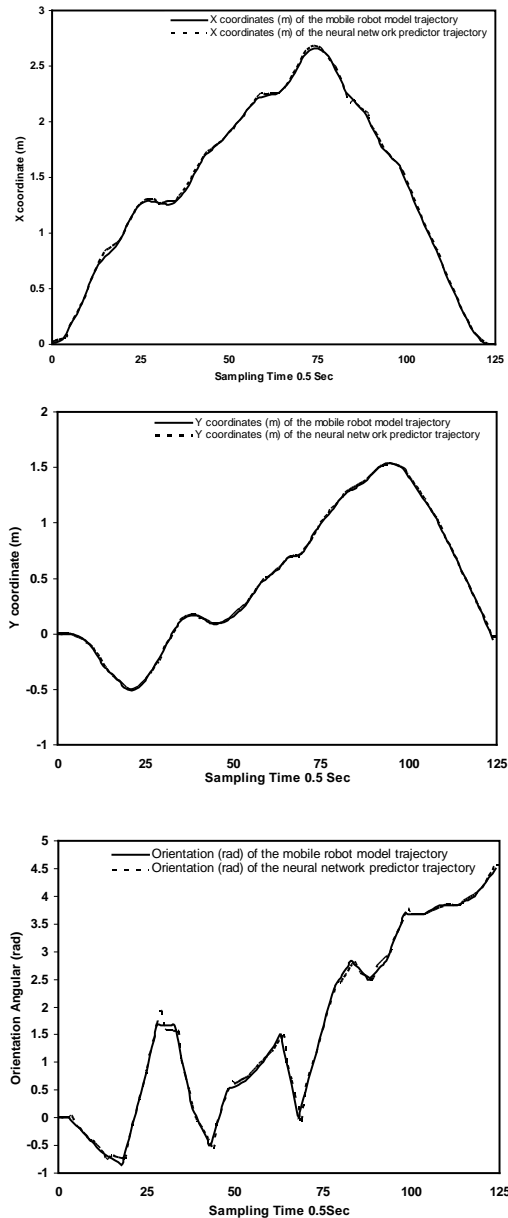


Figure 8. The response of the identifier with the actual mobile robot model output: (a) in the X-coordinate; (b) in the Y-coordinate; and (c) in the θ -orientation.

Finally the case of tracking a lemniscates trajectory for robot model, as shows in figure 3, is demonstrated with optimization algorithm for N-step-ahead prediction. For simulation purposes, the desired trajectory is chosen as described in equations 77 and 78 and the desired orientation angle is taken as expressed in equation 79. The robot model

starts from the initial posture $q(0) = [0.75, -0.25, \pi/2]$ as its initial conditions.

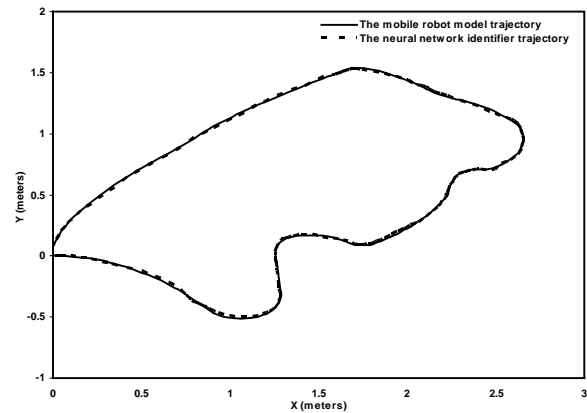


Figure 9. The response of posture identifier with the actual mobile robot model outputs for the training patterns.

A disturbance term $\bar{u}d = [0.01\sin(2t) \ 0.01\sin(2t)]^T$ [5, 6 and 8] is added to the robot system as unmodelled kinematics and dynamics disturbances in order to prove the adaptation and robustness ability of the proposed controller. The feedback neural controller seems to require more tuning effort of its two parameters (Q and R). Q is the sensitivity weighting matrix to the corresponding error between the desired trajectory and identifier trajectory, while the weighting matrix R defines the energy of the input torque signals of right and left wheels. Investigating the feedback control performance of the neural predictive controller can easily obtained by changing the ratio of the weighting matrices (Q and R) as show in figure 10. This also gives the designer the possibility of obtaining more optimized control action depending on the MSE of the position and orientation, which is more difficult to obtain in other controllers. Therefore, the best value of Q parameter is equal to 0.01 and best value of R parameter is equal to 1 for obtaining more optimized control action as shown in figure 10.

The robot trajectory tracking obtained by the proposed adaptive neural predictive controller is shown in figures 11a and 11b. These figures demonstrate excellent position and orientation tracking performance for five steps ahead prediction in comparison with one step ahead prediction. In spite of the existence of bounded disturbances the adaptive learning and robustness of neural controller with optimization algorithm show small effect of these disturbances.

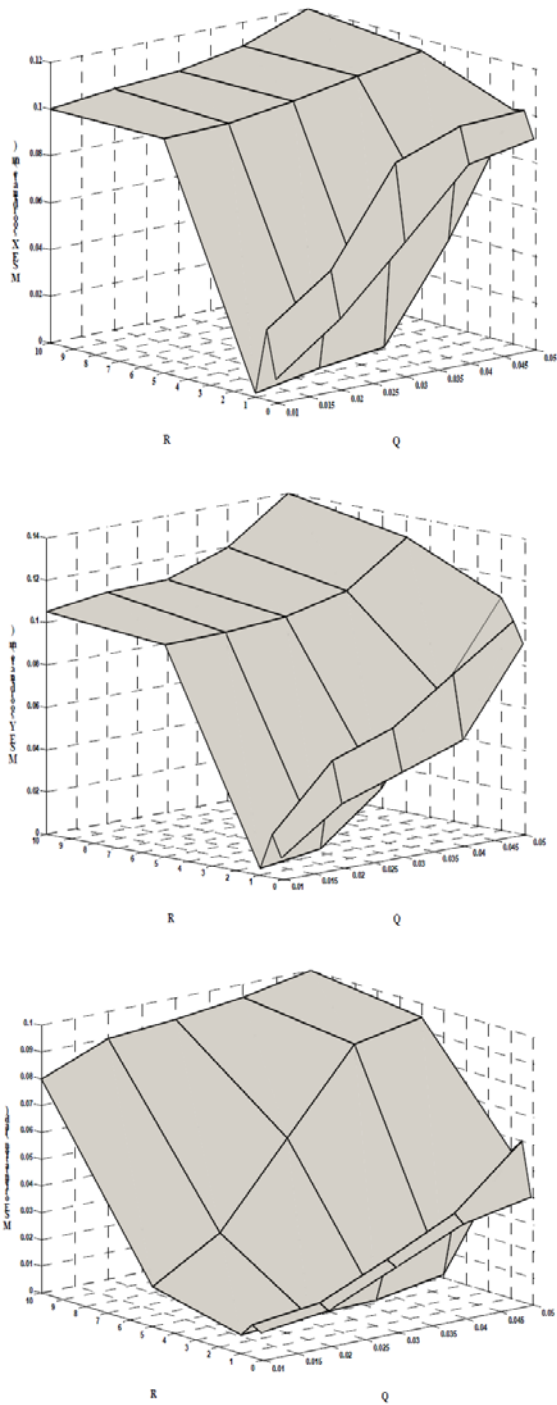


Figure 10. The MSE of position and orientation with (Q&R) parameters.

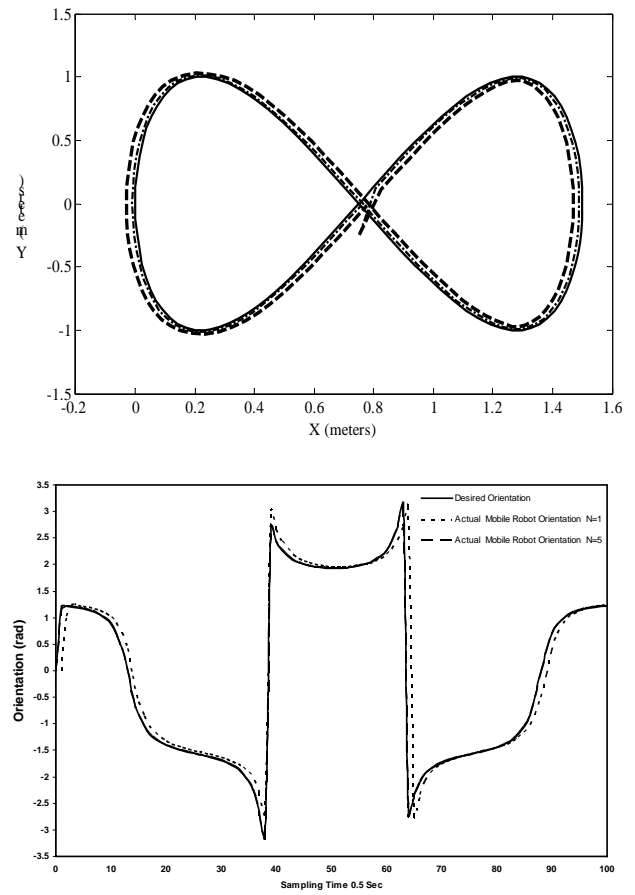


Figure 11. Simulation results for one and five steps ahead predictive: (a) actual and desired lemniscates trajectory; and (b) actual and desired orientation.

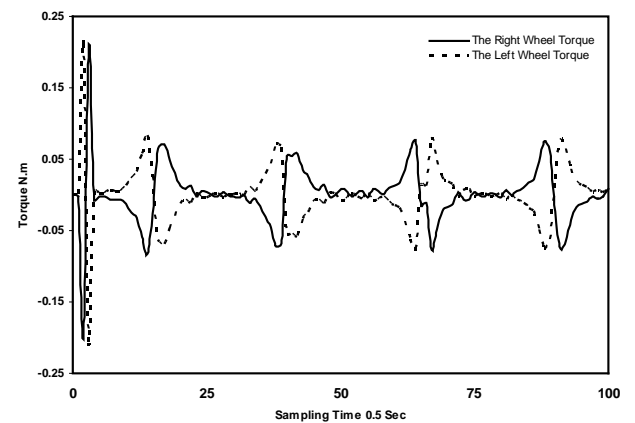
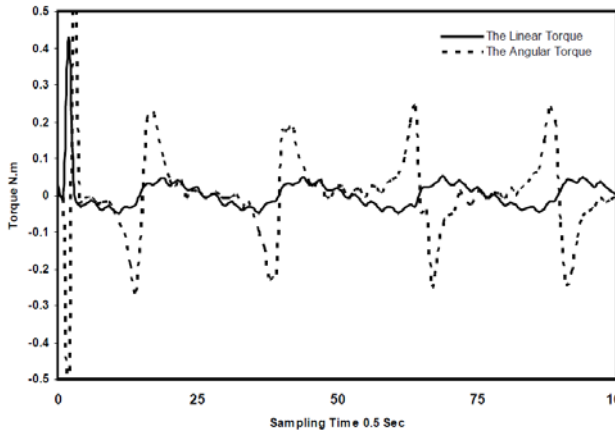
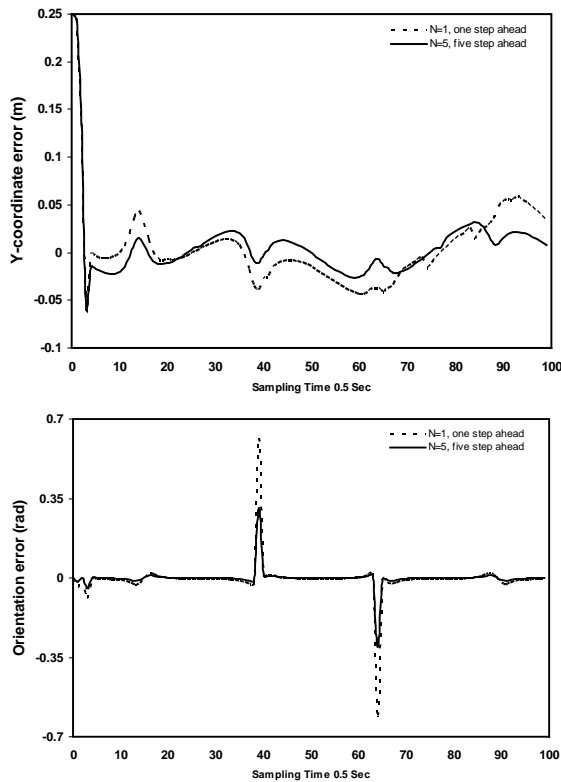


Figure 12a. The torque of the right and left wheel action for N=5.


 Figure 12b. The linear and angular torque action for $N=5$.

 Figure 13. Position and orientation tracking error for two cases $N=1, 5$.

The simulation results demonstrated the effectiveness of the proposed controller by showing its ability to generate small smooth values of the control input torques for right and left wheels without sharp spikes. The actions described in figures 12a and 12b show that smaller power is required to drive the DC motors of the mobile robot model. The effectiveness of the proposed adaptive neural predictive control with predictive optimization algorithm is clear by showing the convergence of the pose trajectory error for the

robot model motion for $N=1$ and 5 steps ahead as shown in figure 13.

The maximum tracking error in the X-coordinate trajectory is equal to $\pm 0.05\text{m}$ for one-step ahead while for the five steps ahead the X- coordinate error is equal to $\pm 0.01\text{m}$. For Y-coordinate tracking error is equal to $\pm 0.05\text{m}$ for one-step ahead and for the five steps ahead the error has declined to less than 0.01m . The maximum tracking error in the orientation of the trajectory is equal to ± 0.67 radian for one-step ahead but it is equal to ± 0.34 radian for five steps ahead.

The mean-square error for each component of the state error $(q_r - q) = (e_x, e_y, e_\theta)$, for the five step ahead predictive control is $MSE(q_r - q) = (0.0012, 0.0017, 0.0387)$, while for one step ahead predictive control is $MSE(q_r - q) = (0.0021, 0.0028, 0.0577)$.

B. Case Study-2

Simulation is also carried out for desired square trajectory which has explicitly non-continuous gradient for verification the capability of the proposed controller performance. The mobile robot model starts from the initial position and orientation $q(0) = [0, -0.1, 0]$ as its initial posture with the same external disturbance are used in case 1 and case 2, and used the same stages of the proposed controller.

Figure 14a shows that the mobile robot tracks the square desired trajectory quite accurately but at the end of one side of the square, there is a sudden increase in position errors of the mobile robot against the desired trajectory at the corners of the square because the desired orientation angle changes suddenly at each corner as shown in figure 14b, therefore, the mobile robot takes a slow turn.

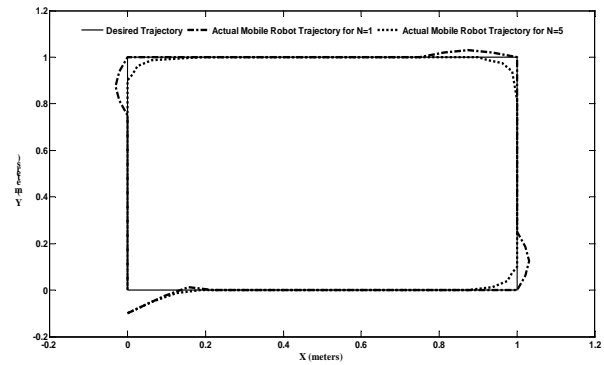


Figure 14a. Actual trajectory of mobile robot and desired trajectory for five steps ahead predictive.

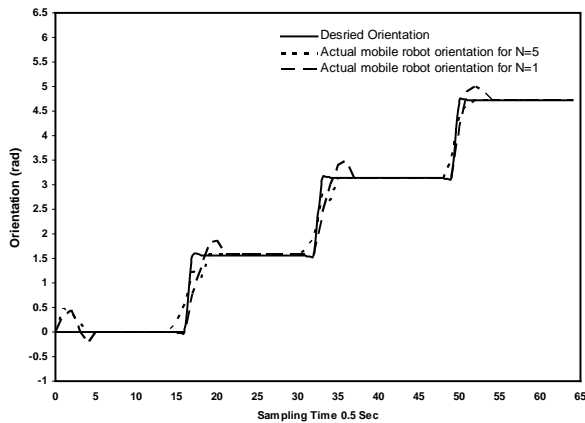


Figure 14b. Actual orientation of mobile robot and desired orientation for one and five steps ahead predictive.

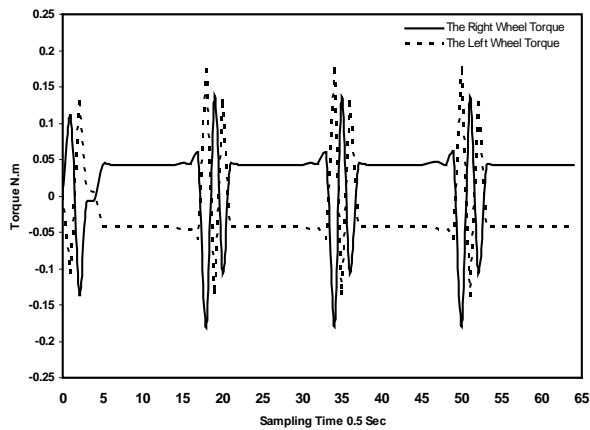


Figure 15a. The right and left wheels torque action for N=5.

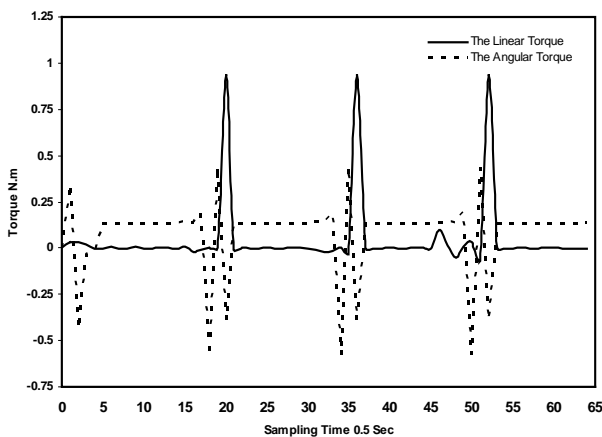


Figure 15b. The linear and angular torque action for N=5.

In figures 15a and 15b, the behaviour of the control action torques for right and left wheels is smooth values with small sharp spikes, when the desired orientation angle changes suddenly at each corner.

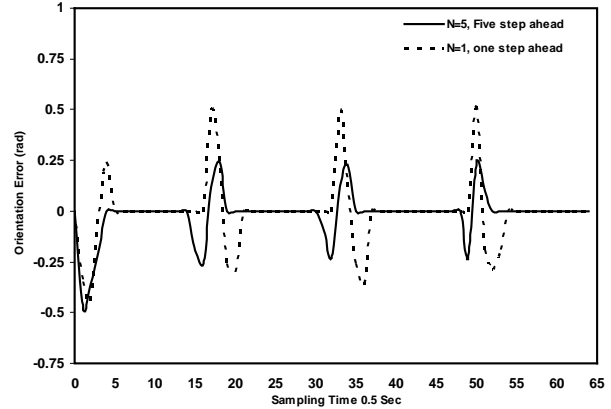
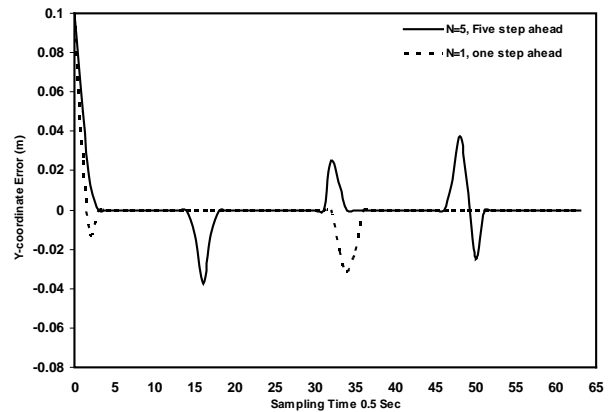
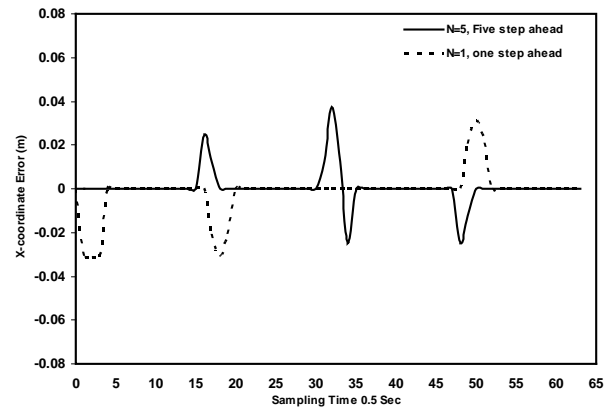


Figure 16. Position and orientation tracking error for N= 5.

In addition, the robot tracks the right side of the square desired trajectory and the tracking errors sharply drop to small values as shown in figure 16.

The maximum X coordinate error in the square trajectory is equal to $\pm 0.03\text{m}$ for one step ahead predication while for the five steps ahead prediction the maximum error in the X-coordinate is equal to $\pm 0.02\text{m}$. The maximum Y coordinate error in the square trajectory is equal to -0.04m for one step ahead predication while for the five steps ahead prediction the maximum error in the Y-coordinate is equal to $\pm 0.03\text{m}$.

Along any one side of the square, the desired orientation angle is constant, therefore the orientation error is equal to zero, but at the end of one side of the square trajectory, the desired orientation angle changes suddenly, therefore, the position and orientation errors of the mobile robot against the desired trajectory at the corners of the square are increasing as shown in figure 16.

The mean-square error for each component of the state error $(q_r - q) = (e_x, e_y, e_\theta)$, for five step ahead predictive control is $MSE(q_r - q) = (0.0007, 0.0018, 0.027)$.

While for one step ahead predictive control is $MSE(q_r - q) = (0.0013, 0.0020, 0.0367)$.

From the simulation results, the five step ahead predictive gives better control results, which is expected because of the more complex control structure, and taking into account future values of the desired, not only the current value, as with one step ahead.

The main advantage of the presented approach is the analytically derived control law which has significantly high computational accuracy with optimization technique to obtain the optimal control action and to minimise tracking error of the continuous and non-continuous gradients (lemniscates and square) trajectories respectively.

V. CONCLUSIONS

The adaptive neural predictive trajectory tracking control methodology for nonholonomic wheeled mobile robot is presented in this paper. The proposed controller consists of three parts: position and orientation neural network identifier, feedforward neural controller and feedback neural controller with optimization algorithm for N-step-ahead prediction. The proposed control scheme minimises the quadratic cost function consisting of tracking errors as well as control effort. It uses two models of neural networks in the structure of the controller, multi-layer perceptron and modified Elman neural network. They are trained off-line and adapted on-line using back propagation algorithm with series-parallel and parallel configurations. Simulation results illustrated evidently that the proposed adaptive neural predictive controller model has the capability of generating smooth and suitable torque commands, τ_R and τ_L without sharp spikes. The proposed controller has demonstrated the capability of tracking continuous and non-continuous gradients desired trajectories and minimises the tracking error approximately $\pm 0.01\text{m}$ for five steps-ahead prediction.

This was demonstrated when bounded external disturbances were added and achieved due to its adaptation ability and robustness behaviour.

REFERENCES

- [1] R-J Wai and C-M Liu, "Design of dynamic petri recurrent fuzzy neural network and its application to path-tracking control of nonholonomic mobile robot", IEEE Transactions on Industrial Electronics, 56(7), pp. 2667-2683, 2009.
- [2] F. Mnif and F. Touati, "An adaptive control scheme for nonholonomic mobile robot with parametric uncertainty", International Journal of Advanced Robotic Systems, 2(1), pp 59-63, 2005.
- [3] G. Klancar and I. Skrjanc, "Tracking error model-based predictive control for mobile robots in real time", Robotics and Autonomous System, 55, pp. 460-469, 2007.
- [4] F.N. Martins, W.C. Celesta, R. Carelli, M. Sarcinelli-Filho, and T.F. Bastos-Filho, "An adaptive dynamic controller for autonomous mobile robot trajectory tracking", Control Engineering Practice. 16, pp. 1354-1363, 2008
- [5] T. Das, I.N. Kar and S. Chaudhury, "Simple neuron-based adaptive controller for a nonholonomic mobile robot including actuator dynamics", Neurocomputing. 69, pp. 2140-2151, 2006.
- [6] J. Ye, "Adaptive control of nonlinear PID-based analogue neural network for a nonholonomic mobile robot", Neurocomputing. 71, pp. 1561-1565, 2008.
- [7] N.A. Martins, D.W. Bertol, E.R. De Pieri, E.B. Castelan and M.M. Dias, "Neural dynamic control of a nonholonomic mobile robot incorporating the actuator dynamics", CIMCA 2008, IAWTIC 2008, and ISA 2008, IEEE Computer Society 2008.
- [8] A. S. Al-Araji, M. F. Abbod and H. S. Al-Raweshidy, "Design of a neural predictive controller for nonholonomic mobile robot based on posture identifier", Proceedings of the IASTED International Conference Intelligent Systems and Control (ISC 2011). Cambridge, United Kingdom, July 11 - 13, 2011. pp. 198-207.
- [9] M.K. Bugeja, S.G. Fabri and L. Camilleri, "Dual adaptive dynamic control of mobile robots using neural networks", IEEE Transactions on Systems, Man, and Cybernetics-Part B: CYBERNETICS, 39(1), pp.129-141, 2009.
- [10] Jun Ye, "Tracking control for nonholonomic mobile robots: integrating the analog neural network into the backstepping technique", Neurocomputing. 71, pp. 3373-3378, 2008.
- [11] H-X Zhang, G-J Dai and H Zeng, "A trajectory tracking control method for nonholonomic mobile robot", Proceedings of the 2007 International Conference on Wavelet Analysis and Pattern Recognition Beijing, China, 2-4 Nov. 2007.
- [12] S. Sun, "Designing approach on trajectory tracking control of mobile robot", Robotics and Computer Integrated Manufacturing. 21, pp. 81-85, 2005.
- [13] R. Siegwart and I. R. Nourbakhah, Introduction to Autonomous Mobile Robots (MIT Press 2004).
- [14] S. X. Yang, H. Yang, Q. Max and H. Meng, "Neural dynamics based full-state tracking control of a mobile robot", IEEE International Conference on Robotics & Automation, New Orleans, LA, April 2004.
- [15] J. Xu, M. Zhang and J. Zhang, "Kinematic model identification of autonomous mobile robot using dynamical recurrent neural networks", IEEE International Conference on Mechatronics & Automation, Niagara Falls, Canada, July 2005.
- [16] J. M. Zurada, Introduction to Artificial Neural Systems (Jaico Publishing House, Pws Pub Co. 1992).
- [17] K. Ogata, Modern Control Engineering (4th Edition, by Addison-Wesley Publishing Company, Inc. 2003).
- [18] K-H. Su, Y-Y. Chen and S-F. Su "Design of neural-fuzzy-based controller for two autonomously driven wheeled robot", Neurocomputing. 73, pp. 2478-2488, 2010.