Applying posture identifier in designing an adaptive nonlinear predictive controller for nonholonomic mobile robot

Ahmed S. Al-Araji *, Maysam F. Abbod, Hamed S. Al-Raweshidy

Wireless Networks and Communication Centre, School of Engineering and Design, Brunel University, London, UK

**Abstract**

This paper presents a trajectory tracking controller for a nonholonomic mobile robot using an optimization algorithm based predictive feedback control and an adaptive posture identifier model while following a continuous and a non-continuous path. The posture identifier model is a modified Elman neural network that describes the kinematics and dynamics of the mobile robot model. The feedforward neural controller is trained off-line and its 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 prediction 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 and experimental work show the effectiveness of the proposed adaptive nonlinear predictive control algorithm; this is demonstrated by the minimized tracking error and the smoothness of the torque control signal obtained, especially with regards to the external disturbance attenuation problem.

**1. Introduction**

During the past few 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 most mobile robots are nonholonomic wheeled mechanical systems. Control problems of mobile robot caused by the motion of wheels that has three degrees of freedom, while control of the mobile robot is done using only two control signals under nonholonomic kinematics constraints.

There are three major reasons for increasing tracking error for mobile robot. First reason for tracking error is the discontinuity of the rotation radius on the path of the differential driving mobile robot. The rotation radius changes at the connecting point of the straight line route and curved route, or at a point of inflection. At these points it can be easy for differential driving mobile robot to secede from its determined orbit due to the rapid change of direction [2]. Therefore, in order to decrease tracking error, the trajectory of the mobile robot must be planned so that the rotation radius is maintained at a constant value, if possible. Second reason for increasing of tracking error is due to the small rotation radius interferes with the accurate driving of the mobile robot. The path of the mobile robot can be divided into curved and straight-line segments. While tracking error is not generated in the straight-line segment, significant error is produced in the curved segment due to centrifugal and centripetal forces, which cause the robot to slide over the surface [2]. Third reason for increasing of tracking error due to the rotation radius is not constant such as the complex curvature or randomly curvature, that is, the points of inflection exist at several locations lead to the mobile robot wheel velocities need to be changed whenever the rotation radius and travelling direction are changed [3,4].

The traditional control methods for trajectory tracking are based on the use of linear or non-linear feedback control while the artificial Intelligent controllers were carried out using neural networks or fuzzy inference systems [5–7] which aimed at tracking a desired mobile robot trajectory with minimum error. The contributions of the presented approach can be understood considering the following points. (1) Overcome the challenge in identifying the position and orientation of the mobile robot for N-step-ahead prediction. (2) The analytically derived control law which has significantly high computational accuracy with predictive optimization technique which are aimed at obtaining the best count of N-step-ahead prediction to find the optimal torques.
control action and lead to minimum tracking error of the mobile robot. (3) Investigation of the controller robustness performance through adding boundary unknown disturbances. (4) Verification of the controller adaptation performance through change the initial pose state. (5) Validation of the controller capability of tracking any trajectories with continuous (lemniscates) and non-continuous (square) gradients.

Simulations and experimental results show that the proposed neural predictive controller is robust and effective in terms of the mobile robot following the trajectory with minimum tracking error and in generating an optimal torque control action despite the presence of bounded external disturbances.

The remainder of this paper is organized as follows: Section 2 is a description of the kinematics and dynamics model of the nonholonomic wheeled mobile robot. In Section 3, the proposed adaptive neural predictive controller is derived. Simulations and experimental results of the proposed controller are presented in Section 4 and the conclusions are drawn in Section 5.

2. Nonholonomic wheeled mobile robot modelling

The schematic of the nonholonomic mobile robot, shown in Fig. 1, consists of a cart with two driving wheels mounted on the same axis and an omni-directional caster in the front of cart. The caster carries the mechanical structure and keeps the platform more stable [8,9]. 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.

The pose of mobile robot in the global coordinate frame [O,X,Y] and the pose vector in the surface are defined as \( q = (x, y, \theta)^T \), where x and y are the coordinates of point c and \( \theta \) is the robot 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 [10,11]:

\[
A_l^T (q) \ddot{q} = \begin{bmatrix} -\sin \theta(t) & \cos \theta(t) & 0 \end{bmatrix}, \quad \ddot{q} = 0
\]  

(1)

Generally, nonholonomic mobile robot systems have an n-dimensional configuration space with n generalization configuration variable \( \{q_1, ..., q_n\} \) and subject to m constraints. Where \( q(t) \in \mathbb{R}^n \times 1 \), \( A(q) \in \mathbb{R}^{n \times m} \).

It is assumed that the mobile robot wheels are ideally installed in such a way that they have ideal rolling without skidding [11,12]. Therefore, the kinematics of the robot can be described as

\[
\dot{q} = \begin{bmatrix} x(t) \\ y(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} \cos \theta(t) & \sin \theta(t) & 0 \\ -\sin \theta(t) & \cos \theta(t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} V_l(t) \\ V_w(t) \end{bmatrix}
\]  

(2)

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

\[
S(q) = \begin{bmatrix} \cos \theta(t) & 0 \\ \sin \theta(t) & 0 \\ 0 & 1 \end{bmatrix}
\]  

(3)

where \( V_l \) and \( V_w \), the linear and angular velocities respectively. Forces must be applied to the mobile robot to produce motion. These forces are modelled by studying the motion of the dynamic model of the differential wheeled mobile robot as shown in Fig. 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 [6–9,13]:

\[
M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + \tau d = B(q, \tau) - A(q) \lambda
\]  

(4)

where \( M(q) \in \mathbb{R}^{n \times n} \) is a symmetric positive definite inertia matrix, \( C(q, \dot{q}) \in \mathbb{R}^{n \times n} \) is the centripetal and cariolic matrix, \( G(q) \in \mathbb{R}^n \) is the gravitational torques vector, \( \tau d \in \mathbb{R}^{n \times 1} \) denotes bounded unknown disturbances including unstructured and unmodelled dynamics, \( B(q) \in \mathbb{R}^{n \times r} \) is the input transformation matrix, \( \tau \in \mathbb{R}^{r \times 1} \) is input torque vector, and \( \lambda \in \mathbb{R}^{m \times 1} \) is the vector of constraint forces.

Remark 1. The plane of each wheel is perpendicular to the ground while the contact between the wheels and the ground is pure rolling and non-slipping, and hence the velocity of the centre of mass of the mobile robot is orthogonal to the rear wheels’ axis and the trajectory of mobile robot base is constrained to the horizontal plane, therefore, \( C(q, \dot{q}) \) is equal to zero.

Remark 2. 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 [8,9].

Remark 3. The centre of mass for mobile robot is located in the middle of axis connecting the rear wheels in c point as shown in Fig. 1, therefore, \( C(q, \dot{q}) \) is equal to zero.

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

\[
\begin{bmatrix} M & 0 \\ 0 & M \\ 0 & 0 \end{bmatrix} \ddot{x} + \tau d = \frac{1}{I} \begin{bmatrix} \cos \theta & \cos \theta & \tau_l \\ \sin \theta & \sin \theta & \tau_r \\ \frac{1}{2} & \frac{1}{2} & \cos \theta \end{bmatrix} + \begin{bmatrix} -\sin \theta \\ \cos \theta \\ 0 \end{bmatrix} \lambda
\]  

(5)

where \( \tau_l \) and \( \tau_r \) are the torques of left and right motors respectively. \( M \) and \( I \) present the mass and inertia of the mobile robot respectively.

By solving (2) and (5) the following normal form can be obtained:

\[
\dot{x} = \frac{\tau_l + \tau_r}{Mr} + \tau d
\]  

(6)

\[
\dot{V}_l = \frac{\tau_l + \tau_r}{2rl} + \tau d
\]  

(7)
where \( \ddot{V}_1 \) and \( \ddot{V}_W \) are the linear and angular acceleration of the differential wheeled mobile robot respectively. The dynamics and the kinematics model structure of the differential wheeled mobile robot can be shown in Fig. 2.

## 3. Adaptive neural predictive controller

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 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 minimize the cost function in order to minimize the tracking error as well as reduce the torque control effort in the presence of external disturbance.

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

(a) Position and orientation neural network identifier;
(b) Feedforward neural controller;
(c) Feedback neural controller.

To apply the idea of predictive optimization algorithm that minimizes the difference between the predicted and the desired robot trajectory in a certain interval. The first step in the procedure of the control structure is the identification of the kinematics and dynamics mobile robot model from the input–output data, second step a feedforward neural controller is designed using feedforward multi-layer perceptron (MLP) neural networks to find reference torques that control the steady-state outputs of the mobile robot trajectory and final step, a robust feedback neural predictive controller is used to stabilize the tracking error of the mobile robot during transient state.

### 3.1. Position and orientation neural network identifier

Nonlinear multi-input multi-output (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 Fig. 4. The nodes of input, context, hidden and output layers are highlighted. The network uses two configuration models, prediction for one-step (series-parallel) and simulation for N-step prediction (parallel) identification structures, which are trained using dynamic back-propagation algorithm (BPA). The parallel structure model is employed using series–parallel structure model. It can be guaranteed that the learning neural networks model of the weights will converge or the error between the output of the system model and that of the neural networks model will lend to zero [14].
The output of the context unit in the modified Elman network is given by [15]:

$$h^c_k = z h^c_{k-1} + \beta h_j(k-1)$$  \hspace{1cm} (10)

where $h^c_k$ and $h_j(k)$ are the outputs of the context and hidden units respectively. $z$ is the feedback gain of the self-connections and $\beta$ is the connection weight from the hidden units (jth) to the context units (cth) at the context layer. The value of $z$ and $\beta$ are selected randomly between (0 and 1) [16,17]. 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 $\theta_m$ is the 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 (11):

$$E = \frac{1}{2} \sum_{i=1}^{np} (x-x_m)^2 + (y-y_m)^2 + (\theta-\theta_m)^2$$  \hspace{1cm} (11)

where np is the number of patterns.

The connexion matrix between the hidden layer and the output layer is $W_{kj}$

$$\Delta W_{kj}(k+1) = -\eta \frac{\partial E}{\partial W_{kj}}$$  \hspace{1cm} (12)

where $\eta$ is the learning rate.

$$\frac{\partial E}{\partial W_{kj}} = \frac{\partial E}{\partial q_m(k+1)} \frac{\partial q_m(k+1)}{\partial \theta_k} \frac{\partial \theta_k}{\partial \eta_k} \frac{\partial \eta_k}{\partial W_{kj}}$$  \hspace{1cm} (13)

$W_{kj}(k+1) = W_{kj}(k) + \Delta W_{kj}(k+1)$  \hspace{1cm} (14)

The connexion matrix between their input layer and the hidden layer is $V_{Hji}$

$$\Delta V_{Hji}(k+1) = -\eta \frac{\partial E}{\partial V_{Hji}}$$  \hspace{1cm} (15)

$$\frac{\partial E}{\partial V_{Hji}} = \frac{\partial E}{\partial q_m(k+1)} \frac{\partial q_m(k+1)}{\partial \theta_k} \frac{\partial \theta_k}{\partial \eta_k} \frac{\partial \eta_k}{\partial V_{Hji}}$$  \hspace{1cm} (16)

$V_{Hji}(k+1) = V_{Hji}(k) + \Delta V_{Hji}(k+1)$  \hspace{1cm} (17)

The connexion matrix between the context layer and the hidden layer is $V_{Cji}$

$$\Delta V_{Cji}(k+1) = -\eta \frac{\partial E}{\partial V_{Cji}}$$  \hspace{1cm} (18)

$$\frac{\partial E}{\partial V_{Cji}} = \frac{\partial E}{\partial q_m(k+1)} \frac{\partial q_m(k+1)}{\partial \theta_k} \frac{\partial \theta_k}{\partial \eta_k} \frac{\partial \eta_k}{\partial V_{Cji}}$$  \hspace{1cm} (19)

$V_{Cji}(k+1) = V_{Cji}(k) + \Delta V_{Cji}(k+1)$  \hspace{1cm} (20)

### 3.2. 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, $\tau_{ref}(k)$ 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 weights adaptation is used to modify the reference control action of the FFNC when there is any disturbance effect. This approach is currently considered as one of the better approaches that can be followed to overcome the lack of initial knowledge because it depends on the posture neural network identifier to find the mobile robot Jacobian through the neural identifier model.

The dynamic back propagation algorithm (BPA) is employed to realize the training the weights of the feedforward neural controller. The sum of the square of the differences between the desired posture $q_i=(x_i,y_i,\theta_i)^T$ and neural network posture $q_m=(x_m,y_m,\theta_m)^T$ is given by:

$$E_c = \frac{1}{2 \sum_{i=1}^{np}} \left( (x_i-x_m)^2 + (y_i-y_m)^2 + (\theta_i-\theta_m)^2 \right)$$  \hspace{1cm} (21)

where npc is number of patterns.

The connexion matrix between hidden layer and output layer is $W_{Contba}$

$$\Delta W_{Contba}(k+1) = -\eta \frac{\partial E_c}{\partial W_{Contba}}$$  \hspace{1cm} (22)

$$\frac{\partial E_c}{\partial W_{Contba}} = \frac{\partial E_c}{\partial q_m(k+1)} \frac{\partial q_m(k+1)}{\partial \theta_k} \frac{\partial \theta_k}{\partial \eta_k} \frac{\partial \eta_k}{\partial W_{Contba}}$$  \hspace{1cm} (23)

$$\frac{\partial E_c}{\partial q_m(k+1)} = \frac{1}{2 \sum_{i=1}^{np}} \left( (x_i-x_m)^2 + (y_i-y_m)^2 + (\theta_i-\theta_m)^2 \right)$$  \hspace{1cm} (24)

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.

![Fig. 5. MLP neural networks act as the feedforward neural controller.](image)

Fig. 6. The feedforward neural controller structure.
The configuration error can be represented by using a transformation matrix as
\[
\begin{bmatrix}
ex_m \\
ey_m \\
et_m
\end{bmatrix}
= \begin{bmatrix}
cos\theta_m & \sin\theta_m & 0 \\-
\sin\theta_m & \cos\theta_m & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x_r-x_m \\
y_r-y_m \\
\theta_r-\theta_m
\end{bmatrix}
\]
(25)
where \(x_r, y_r\) and \(\theta_r\) are the reference posture of the mobile robot.
\[
Jacobian = \frac{\partial q_m(k+1)}{\partial \tau_{ref}(k)}
\]  
(26)
where the outputs of the identifier are \(q_m\).
\[
\frac{\partial q_m(k+1)}{\partial \tau_{ref}(k)} = \frac{\partial q_m(k+1)}{\partial \tau_{net}(k)} \times \frac{\partial \tau_{net}(k)}{\partial \tau_{ref}(k)}/C_0
\]
(27)
Substituting (24) and (27) into (23), to find \(\Delta W_{contan}(k+1)\), then
\[
W_{contan}(k+1) = W_{contan}(k) + \Delta W_{contan}(k+1)
\]
(28)
The connection matrix between input layer and hidden layer is \(V_{contan}\)
\[
\Delta V_{contan}(k+1) = -\eta \frac{\partial Ec}{\partial V_{contan}}
\]
(29)
\[
\frac{\partial Ec}{\partial V_{contan}} = \frac{\partial Ec}{\partial q_m(k+1)} \times \frac{\partial q_m(k+1)}{\partial \tau_{ref}(k)}/C_0 \times \frac{\partial \tau_{ref}(k)}{\partial \tau_{net}(k)}/C_0 \times \frac{\partial \tau_{net}(k)}{\partial \tau_{contan}}/C_0
\]
(30)
Substituting (24) and (27) into (30), to find \(\Delta V_{contan}(k+1)\), then
\[
V_{contan}(k+1) = V_{contan}(k) + \Delta V_{contan}(k+1)
\]
(31)
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 \(\tau_{ref}\) and \(\tau_{ref2}\), the reference torques of the right and left wheels respectively.

### 3.3. Feedback neural controller

The feedback neural controller (FNBC) is essential to stabilize the tracking error of the mobile robot system when the trajectory of the robot is drifted from the desired trajectory during transient state. The feedback neural controller generates an optimal torque control action that minimizes the cumulative error between the reference input trajectory and the output trajectory of the mobile robot.

These can be achieved by using quadratic performance index with optimization algorithm for \(N\) step ahead prediction and on-line adapted the posture identifier weights in order to overcome the disturbances and parameter variations in the mobile robot model.

A quadratic cost functions for the neural controller can be expressed as in Eq. (32):
\[
J = \frac{1}{2} \sum_{k=1}^{N} Q[(x_r(k+1)-x_m(k+1))^2+(y_r(k+1)-y_m(k+1))^2 \\
+(\theta_r(k+1)-\theta_m(k+1))^2] + \frac{1}{2} \sum_{k=1}^{N} R[(\tau_{1r}(k))^2+(\tau_{2r}(k))^2]
\]
(32)
where \(Q\) 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.

### For \(N\) steps estimation of the two feedback neural controller actions \(\tau_{1r}(k)\) and \(\tau_{2r}(k)\) the techniques of generalized predictive control theory will be used. The position and orientation in the identifier model, shown in Fig. 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 \(\tau_{1r}(k)\) and \(\tau_{2r}(k)\) 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:
\[
X_{r1,N} = [x_r(t+1),x_r(t+2),x_r(t+3),...,x_r(t+N)]
\]
(33)
\[
Y_{r1,N} = [y_r(t+1),y_r(t+2),y_r(t+3),...,y_r(t+N)]
\]
(34)
\[
\theta_{r1,N} = [\theta_r(t+1),\theta_r(t+2),\theta_r(t+3),...,\theta_r(t+N)]
\]
(35)
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 Fig. 4, are:
\[
X_{m1,N} = [x_m(t+1),x_m(t+2),x_m(t+3),...,x_m(t+N)]
\]
(36)
\[
Y_{m1,N} = [y_m(t+1),y_m(t+2),y_m(t+3),...,y_m(t+N)]
\]
(37)
\[
\theta_{m1,N} = [\theta_m(t+1),\theta_m(t+2),\theta_m(t+3),...,\theta_m(t+N)]
\]
(38)
The error vector of position and orientation as \((39)\) and \((40)\) can be calculated by using (25).
\[
E_{Xm1,N} = [e_{Xm1,N}(t+1),e_{Xm1,N}(t+2),e_{Xm1,N}(t+3),...,e_{Xm1,N}(t+N)]
\]
(39)
\[
E_{Ym1,N} = [e_{Ym1,N}(t+1),e_{Ym1,N}(t+2),e_{Ym1,N}(t+3),...,e_{Ym1,N}(t+N)]
\]
(40)
\[
E_{\theta m1,N} = [e_{\theta m1,N}(t+1),e_{\theta m1,N}(t+2),e_{\theta m1,N}(t+3),...,e_{\theta m1,N}(t+N)]
\]
(41)
Two-feedback control signals can be determined by:
\[
\tau_{1r,N} = [\tau_{1r}(t+1),\tau_{1r}(t+2),...,\tau_{1r}(t+N-1)]
\]
(42)
\[
\tau_{2r,N} = [\tau_{2r}(t+1),\tau_{2r}(t+2),...,\tau_{2r}(t+N-1)]
\]
(43)
Assuming the following objective function:
\[
J_1 = \frac{1}{2} \sum_{k=1}^{N} \left[(E_{Xm1,N}E_{Xm1,N})^T + (E_{Ym1,N}E_{Ym1,N})^T + (E_{\theta m1,N}E_{\theta m1,N})^T\right]
\]
(44)
Then our purpose is to find \(\tau_{1r}\) and \(\tau_{2r}\) two feedback control actions such that \(J_1\) is minimized using the gradient descent rule, The new control actions will be given by:
\[
\tau_{1r,N}^{+} = \tau_{1r,N}^{+} + \Delta \tau_{1r,N}^{+} K
\]
(45)
\[
\tau_{2r,N}^{+} = \tau_{2r,N}^{+} + \Delta \tau_{2r,N}^{+} K
\]
(46)
where \(K\) indicates that calculations are performed at the \(k\)th sample; and
\[
\Delta \tau_{1r,N}^{+} = -\eta \frac{\partial J_1}{\partial \tau_{1r,N}^{+}} = [\Delta \tau_{1r}(t),\Delta \tau_{1r}(t+1),\Delta \tau_{1r}(t+2),...,\Delta \tau_{1r}(t+N-1)]
\]
(47)
\[
\Delta \tau_{2r,N}^{+} = -\eta \frac{\partial J_1}{\partial \tau_{2r,N}^{+}} = [\Delta \tau_{2r}(t),\Delta \tau_{2r}(t+1),\Delta \tau_{2r}(t+2),...,\Delta \tau_{2r}(t+N-1)]
\]
(48)
\[
-\eta \frac{\partial J_1}{\partial \tau_{1r,N}^{+}} = \eta Q_{EXm1,N} E_{Xm1,N}^T + \eta Q_{EYm1,N} E_{Ym1,N}^T K + \eta Q_{E\theta m1,N} E_{\theta m1,N}^T K
\]
(49)
t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t
	n

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t

t
	n

t
	n

t

t

t

t

t

t

t

t

t

t
	n

t

t

t

t
\frac{\partial J_{2L,N}}{\partial \tau_{2L,N}} = \eta QEX_{m,N} \frac{\partial X_{m,N}}{\partial \tau_{2L,N}} + \eta QEY_{m,N} \frac{\partial Y_{m,N}}{\partial \tau_{2L,N}} + \eta QE\theta_{m,N} \frac{\partial \theta_{m,N}}{\partial \tau_{2L,N}} - \eta \tau_{2L,N} \frac{\partial \tau_{2L,N}}{\partial \tau_{2L,N}} \tag{50}

where

\frac{\partial X_{m,N}}{\partial \tau_{2L,N}} = \left[ \frac{\partial X_{m}(t+1)}{\partial \tau_{1}(t)} \frac{\partial X_{m}(t+2)}{\partial \tau_{1}(t)} \ldots \frac{\partial X_{m}(t+N)}{\partial \tau_{1}(t+N-1)} \right] \tag{51}

\frac{\partial Y_{m,N}}{\partial \tau_{2L,N}} = \left[ \frac{\partial Y_{m}(t+1)}{\partial \tau_{2}(t)} \frac{\partial Y_{m}(t+2)}{\partial \tau_{2}(t)} \ldots \frac{\partial Y_{m}(t+N)}{\partial \tau_{2}(t+N-1)} \right] \tag{52}

\frac{\partial \theta_{m,N}}{\partial \tau_{2L,N}} = \left[ \frac{\partial \theta_{m}(t+1)}{\partial \tau_{3}(t)} \frac{\partial \theta_{m}(t+2)}{\partial \tau_{3}(t)} \ldots \frac{\partial \theta_{m}(t+N)}{\partial \tau_{3}(t+N-1)} \right] \tag{53}

\frac{\partial \theta_{m,N}}{\partial \tau_{2L,N}} = \left[ \frac{\partial \theta_{m}(t+1)}{\partial \tau_{3}(t)} \frac{\partial \theta_{m}(t+2)}{\partial \tau_{3}(t)} \ldots \frac{\partial \theta_{m}(t+N)}{\partial \tau_{3}(t+N-1)} \right] \tag{54}

After completing the procedure from \( n = 1 \) to \( N \) the new control actions for the next sample will be:

\tau_{k}(k+1) = \tau_{ref}(k+1) + \tau_{c}(k+1) \tag{57}

\tau_{l}(k+1) = \tau_{ref}(k+1) + \tau_{c}(k+1) \tag{58}

where \( \tau_{c}(k+1) \) and \( \tau_{c}(k+1) \) are the final values of the feedback-controlling signals calculated by the optimization algorithm. This is calculated at each sample time \( k \) so that \( \tau_{c}(k+1) \) and \( \tau_{c}(k+1) \) 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 amplification 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.

4. Simulations and experimental results

The proposed controller is verified by means of computer simulation using MATLAB/SIMULINK. The kinematics and dynamic model of the nonholonomic mobile robot described in Section 2 is used. The simulation is carried out by tracking a desired position \((x, y)\) and orientation angle \(\theta\) with a lemniscates and square trajectories in the tracking control of the robot. The parameter values of the robot model are taken from [19,20]: \(M = 0.65\) kg, \(I = 0.36\) kg m², \(L = 0.105\) m and \(r = 0.033\) m.

The proposed controller is implemented based on the structure shown in Fig. 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 is 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 cover the specified operating region. A hybrid excitation signal has been used for the robot model. Fig. 7 shows the input signals \(\tau_{k}(k)\) and \(\tau_{l}(k)\), right and left wheel torques respectively.

The training set is generated by feeding a pseudo random binary signal (PRBS), with sampling time of 0.5 s, to the model and measuring its corresponding outputs, position \(x\) and \(y\) and orientation \(\theta\) and the minimum number of the nodes in the hidden layer is equal to the number of nodes in the input layer [15,16]. The nodes in the context layer are equal to the node in the hidden layer; therefore, 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 Fig. 4.

A training set of 125 patterns has been used with a learning rate of 0.1. After 3244 epochs the mean square error has reached less than \(5.7 \times 10^{-6}\). The identifier outputs of the neural network position \(x, y\) and orientation \(\theta\), are approximated to the actual outputs of the model trajectory as shown in Fig. 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 epochs the same training set patterns has been achieved with a mean square error less than \(2.31 \times 10^{-6}\). The neural network identifier position and orientation outputs and the mobile robot model trajectory are shown in Fig. 9.

4.1. Case study-1

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

\[x(t) = 0.75 + 0.75 \times \sin(\frac{2\pi t}{50})\] \tag{59}

\[y(t) = \sin(\frac{4\pi t}{50})\] \tag{60}

\[\theta(t) = 2\tan^{-1}\left(\frac{\Delta y(t)}{\sqrt{(\Delta x(t))^2 - (\Delta y(t))^2 + \Delta x(t)}}\right)\] \tag{61}

The second stage of the proposed controller is feedforward neural controller. It uses multi-layer perceptron neural network (8–16–2) as shown in Fig. 5 where the maximum number of the nodes in the hidden layer can be expressed as \(2n+1\) where \(n\) is
the number of nodes in the input layer [14]. 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 Fig. 6 to find the suitable reference torque control action at steady state. Finally the case of tracking a lemniscates trajectory for robot model, as shows in Fig. 3, is demonstrated with optimization algorithm for $N$-step-ahead prediction.

For simulation purposes, the desired trajectory is chosen as described in (59) and (60) while the desired orientation angle is taken as expressed in (61).

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

A disturbance term $\tau_d = \begin{bmatrix} 0.01 \sin(2t) & 0.01 \sin(2t) \end{bmatrix}^T$ 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 $R$). Investigating the feedback control performance of the neural predictive controller can easily obtain by changing the ratio of the weighting matrices ($Q$ and $R$) as show in Fig. 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 Fig. 10.

After picking the best values of $Q$ and $R$ for $N$ is equal to one-step-ahead, it becomes necessary to choose the best count for $N$ step-ahead prediction. This is accomplished by carrying out the simulation of the desired lemniscates trajectory of the mobile robot with optimization algorithm for different $N$s (1–10) then calculating the position and orientation mean square error for each case in order to select the best count $N$ for smallest position and orientation MSE. Fig. 11 shows that the best count for $N$ step-ahead is equal to 5 although the position and orientation mean square error are the same in case $N=6$ and $N=7$ but it is necessary to take the execution time of the simulation when $N=6$ or $N=7$ it takes a long time to calculate the optimal control action, so the best count for $N$ step-ahead is equal to 5.

The robot trajectory tracking obtained by the proposed adaptive neural predictive controller is shown in Fig. 12. These figures demonstrate excellent position and orientation tracking performance for the five step-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.

The simulation results demonstrated the effectiveness of the proposed controller by showing its ability to generate small smooth valves of the control input torques for right and left wheels without sharp spikes. The actions described in Fig. 13 shows that smaller power is required to drive the DC motors of the mobile robot model.
The effectiveness of the proposed adaptive neural 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 by using mean-square error for each component of the state error is \( (q_r, e_r, e_0) \), for the one step-ahead prediction control is \( \text{MSE}(q_r - q) = (0.0021, 0.0028, 0.0577) \) while for five-step ahead prediction control is \( \text{MSE}(q_r - q) = (0.0012, 0.0017, 0.0387) \). Despite the presence of disturbances \( t_d = 0.01 \sin(2t) \), the mean square error has been minimized for five step-ahead prediction due to the capability of the controller robustness and adaptation performances.

### 4.2. 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 \( q(0) = [0, 0, 0] \) as its initial posture with the same external disturbance is used in case 1 and used the same stages of the proposed controller with best value of \( Q \) is equal to 0.01 and \( R \) is equal to 1.

Fig. 14 shows that the best count for \( N \) step ahead is equal to five because of minimum position mean square error and orientation mean square error.

Fig. 15a 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 Fig. 15b, therefore, the mobile robot takes a slow turn.

Fig. 16 shows 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.

Using mean-square error for each component of the state error for the five-step ahead predictive control is \( \text{MSE}(q_r - q) = (0.0007, 0.0018, 0.0277) \). While for one step ahead predictive control is \( \text{MSE}(q_r - q) = (0.0013, 0.002, 0.0367) \).

The main advantage 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 fast response with 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. Simulation results show that the five step-ahead prediction gives better control results, which is expected because of the more complex control structure, and also due to taking into account future values of the desired, not only the current value, as with one step ahead.

In order to validate the applicability of the proposed controller, experiments have executed by using mobile robot from PARALLAX Inc. The lab experiments have been conducted using a Boe-Bot robotics type nonholonomic wheeled mobile robot (V3) as shown in Fig. 17. The wheeled mobile robot is equipped with BASIC Stamp 2 programmable (BS2) microcontroller type (PIC16C57c) consisting of EEPROM 2KByte, a decoding logic unit, infrared sensors, PWM generator for differential control of the robot \[19,20\].

Velocities commands sent by the computer are coded messages which are recognized by microcontroller. Based on received characters, the microcontroller creates control actions for servo motors.

The output voltages of the two IR sensors are converted to coded messages by microcontroller and sent to the personal computer in order to calculate the tracking error of the mobile robot during motion. It is modified the data transmitting between the Boe-Bot robot and main computer from wire to wireless communication by using wireless USB Hub and adaptor that has radio speed up to 480 Mbps and forty times faster than wireless Internet (802.11b) protocol \[22\].

In the experiments, the best control data action of the simulations was the five step-ahead action of the control. These control data has transmitted to the Boe-Bot mobile robot model, which admits right wheel velocity and left wheel velocity as input signals by using wireless USB hub communication after has been converted the data format from MATLAB file of simulations to BASIC Stamp Editor Software version 2.5 format as a lookup table.

The velocities of the simulation results for right and left wheels have downloaded to the memory of the Boe-Bot mobile robot as commands which have smooth values without sharp spikes and can be shown in Fig. 18a. The mean of linear velocity of
the mobile robot is equal to 0.135 m/sec, and maximum angular
velocity is equal to \(0.52\ \text{rad/sec}\), as shown in Fig. 18b.

The initial pose for the Boe-Bot mobile robot starts at position
0.75 and \(-0.25\) m and orientation 1.57 rad and should follow
desired lemniscates trajectory as show in Fig. 19. The desired
trajectory starts at position 0.75 and 0. After 50 s, the mobile robot
has finished the tracking of the desired path and the tracking was
reasonably accurate the mean-square error for each component of
the state error, was \(\text{MSE}(q_r-q) = (0.0014, 0.0019, 0.0417)\).

For the desired square trajectory, the Boe-Bot mobile robot
starts initial position 0 and \(-0.1\) m and initial orientation zero
radian, and should follow desired path as shown in Fig. 20. The
best control data action of the simulations has transmitted and
downloaded to the memory of Boe-Bot mobile robot model,
which admits right wheel velocity and left wheel velocity as

![Fig. 16. Torque action for \(N=5\): (a) the right and left wheel torque and (b) the linear and angular torque.](image)

![Fig. 17. Boe-Bot mobile robot for the experiments.](image)

![Fig. 18. Velocity action for \(N=5\): (a) the right and left wheel velocity and (b) the linear and angular velocity.](image)

![Fig. 19. Real set-up experiment of Boe-Bot robot for lemniscates trajectory tracking.](image)

![Fig. 20. Real set-up experiment of Boe-Bot robot for square trajectory tracking.](image)
An adaptive neural predictive trajectory tracking controller for nonholonomic wheeled mobile robot has been presented in this paper. The proposed controller consists of three parts: position and orientation neural network identifier, feedforward neural controller and nonlinear neural feedback controller with predictive optimization algorithm for N step ahead. The proposed control scheme minimizes the quadratic cost function which consists of the tracking errors and the control effort. Simulation results and experimental work show evidently that the proposed adaptive neural predictive controller model has the capability of generating smooth and suitable torque (\(T_R\) and \(T_L\)) and velocity (\(V_R\) and \(V_L\)) commands, without sharp spikes. The proposed controller has demonstrated the capability of tracking continuous and non-continuous gradients desired trajectories and effectively minimizing the tracking errors of the nonholonomic wheeled mobile robot model.

5. Conclusions

An adaptive neural predictive trajectory tracking controller for nonholonomic wheeled mobile robot has been presented in this paper. The proposed controller consists of three parts: position and orientation neural network identifier, feedforward neural controller and nonlinear neural feedback controller with predictive optimization algorithm for N step ahead. The proposed control scheme minimizes the quadratic cost function which consists of the tracking errors and the control effort. Simulation results and experimental work show evidently that the proposed adaptive neural predictive controller model has the capability of generating smooth and suitable torque (\(T_R\) and \(T_L\)) and velocity (\(V_R\) and \(V_L\)) commands, without sharp spikes. The proposed controller has demonstrated the capability of tracking continuous and non-continuous gradients desired trajectories and effectively minimizing the tracking errors of the nonholonomic wheeled mobile robot model.
Maysam Abbod was born in 1966. He has finished his BSc degree in electronics from the University of Technology, Baghdad in 1987 and PhD degree in Control Engineering from the University of Sheffield in 1992. His research work is of a multi-disciplinary nature ranging from biomedical applications to industrial applications. He has been a Research Fellow working for the Institute for Microstructural and Mechanical Process Engineering, The University of Sheffield (IMMPETUS) on modelling of material properties using hybrid modelling techniques. His research interests include intelligent systems for modelling and optimization. Currently he is a lecturer in Intelligent Systems in the School of Engineering and Design, Brunel University.

Dr Abbod is a member of the IET and Chartered Engineer. He has published over 40 papers in journals, 9 chapters in edited books, and over 50 papers in refereed conferences.

Hamed Al-Raweshidy is a research professor and the head of the Wireless Networks & Communications Research Group in the School of Engineering and Design at Brunel University. His research interests include network optimization, mesh and ad hoc networks, radio over fiber, PAN and PN, and IP mobility. He received his PhD for work in spread spectrum multiplexing for communication networks. He holds a number of projects with EPSRC and EU. He is a senior member of the IEEE, an IEE fellow, and a member of New York Academy of Sciences. Contact him at the Wireless Networks & Communications Group, School of Eng. and Design, Brunel Univ., Uxbridge, Middlesex UB8 3PH, UK.