Design of a Backstepping Controller based on an Adaptive Elman Neural Network for a Two-Link Robot System

Aliaa M. Sadek
Computers and Control Engineering Department,
Faculty of Engineering, Tanta University
Tanta, Egypt
aliaasadek608@yahoo.com

Wael M. Elawady
Computers and Control Engineering Department,
Faculty of Engineering, Tanta University
Now, Lecturer at the Higher Institute of Engineering and Technology in Kafrelshiekh
Tanta, Egypt
Wmeana2007@yahoo.com

Amany M. Sarhan
Head of and professor at Computers and Control Engineering Department
Faculty of Engineering, Tanta University
Tanta, Egypt
amany_sarhan@f-eng.tanta.edu.eg

***Abstract*—This paper presents a backstepping controller based on an adaptive Elman neural network (BSAENN) to solve the mismatched uncertainty problem of underactuated robotic systems. First, the nonlinear dynamical equations of the underactuated robotic system are transformed to a cascade form. Second, an adaptive backstepping controller has been established. This controller is adopted using the combination of the adaptive Elman neural network (AENN), which is a recurrent multilayered neural network, and the traditional backstepping control (TBS) approach. The AENN with adaptive parameters is exploited to approximate the unknown system parameters and enhance the control behavior against uncertainties. The adaptation laws of the AENN parameters and the stability of the closed-loop system are proved using Lyapunov stability. Numerical simulations with dynamical model of the two link robot, compared to conventional controllers (PID and TBS), show that the proposed controller provides robustness for trajectory tracking performance under the occurrence of uncertainties**

Keywords—Elman neural network, backstepping controller, uncertainty, robot, PID

I. Introduction

Control of underactuated mechanical systems became an essential research approach because of its applications, like cars, mobile robots, snake-type robots, aircraft, and underwater vehicles [16]. To overcome the problems of underactuated mechanical systems, many control techniques have been developed such as sliding mode control [15], backstepping control [8], and neural network–based adaptive control [9], [12]. The major problem in nonlinear dynamics of the under actuated mechanical systems is that the system has an input and two subsystems. Only one degrees-of- freedom (DOF) can be controlled. Therefore, it is impossible to use smooth feedback to stabilize the system around equilibrium even locally. To overcome this drawback, a transformation from underactuated form to a classic cascade form should be made [10], [20].

Approximation of unknown nonlinear system dynamics and uncertainties can be achieved using wavelets neural networks [8] or combination between neural networks and fuzzy control [5], [7] [18]. In [10], the authors presented a control algorithm using the TBS and fuzzy logic control in which, the nonlinear dynamics have been approximated using fuzzy controller. The disadvantage of this controller is that it requires a large number of fuzzy rules. To overcome this drawback, in this paper, the fuzzy logic approximator is supplanted with adaptive Elman neural network (AENN)–based approximator. The main merit of the AENN is that it has faster convergence rate and can handle dynamic response faster. The backstepping control [4], [6], [7], [14], [19] is a recursive control method which has the ability to prevent useful nonlinearities from being canceled.

In this paper, a backstepping controller based on an adaptive Elman neural network (BSAENN) is presented to control the nonlinear underactuated robot system. The adopted BSAENN control algorithm is motivated from the published authors work in [2] and [10] to overcome the disadvantages of the fuzzy logic approximator of the work in [10]. First, the concept of TSB controller in [10] is augmented with the concept of the AENN approximator in [2] to combine the advantages of the both techniques in [2] and [10].

In designing the BSAENN controller, first, the nonlinear dynamical equations of the underactuated robotic system are transformed to a cascade form, second, the proposed backstepping controller based on adaptive Elman neural network (BSAENN).

The AENN approximator has been designed to estimate the unknown system dynamics and uncertainties. The adaptation laws of the AENN parameters and the stability of the closed-loop system are proved using Lyapunov stability. Numerical simulations with dynamical model of the two link robot, compared to conventional controllers (PID and TBS), show that the proposed controller provides robustness for trajectory tracking performance under the occurrence of uncertainties.

This paper is organized as follows: in Section II, the description of the two link robot system is presented. Section III presents the BSAENC controller design. Section IV presents the simulation results. Section V presents the conclusion of the presented work.

II. PROPLEM FORMULATION

In this paper, the proposed BSAENN is applied to control the 2nd order nonlinear underactuated robot system, shown in Fig. (1) that is described by the following equation, [3], [11], [13], and [17]:

$$B(q)\ddot{q}+C(q,\dot{q})\dot{q}+G(q)+F(\dot{q})=\tau \tag{1}$$

where $q=\begin{pmatrix} q_1 \\ q_2 \end{pmatrix}$ represents joint position ,

$B(q)=\begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$ represents the inertia matrix,

$$C(q,\dot{q})=\begin{bmatrix} c_{11}(q,\dot{q}) & c_{12}(q,\dot{q}) \\ c_{21}(q,\dot{q}) & c_{22}(q,\dot{q}) \end{bmatrix}$$

represents the coriolis and centrifugal forces matrix,

$$G(q)=\begin{bmatrix} g_1(q) \\ g_2(q) \end{bmatrix}$$

represents the gravity force vector.

$$F(\dot{q})=\begin{bmatrix} F_1(\dot{q}) \\ F_2(\dot{q}) \end{bmatrix}$$

represents the friction forces vector, τ represents the applied torque exerted on the joints.

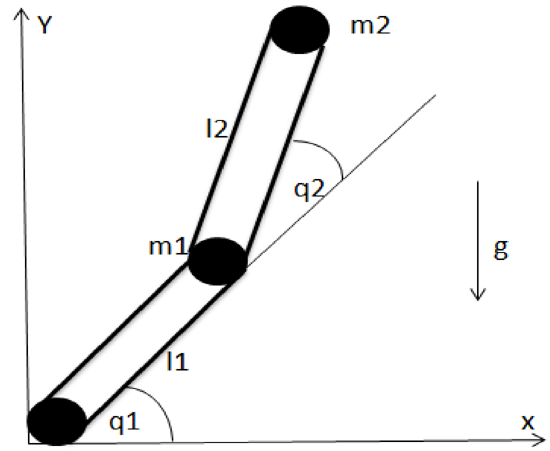

Fig. 1 Model of two-link robot manipulator

To design backstepping controller, the state space model of Eq. (1) is required.

define the state variables as

$$x_1=q_1,\quad x_2=\dot{q}_1,\quad x_3=q_2,\quad x_4=\dot{q}_2 \tag{2}$$

From Eq. (2) the state space model is given by:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= f_1(x)+N_1(x)u \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= f_2(x)+N_2(x)u \end{aligned} \tag{3}$$

where $x=[x_1 \quad x_2 \quad x_3 \quad x_4]^T$ is the state vector, u is the control input, f_1, f_2, N_1 and N_2 are nonlinear functions which are given by :

$$f_1(x)=\frac{(q_2q_3(x_2+x_4)^2\sin x_3+q_3{}^2x_2{}^2\sin x_3\cos x_3)}{q_1q_2-q_3{}^2\cos^2 x_3} -\frac{(-q_2q_4\cos x_1+q_3q_5 g\cos x_3\cos(x_1+x_3)}{q_1q_2-q_3{}^2\cos^2 x_3} \tag{4}$$

$$f2(x)=\frac{(q_3{}^2(x_2+x_4)^2\sin x_3\cos x_3-q_3q_4\cos x_1\cos x_3)}{q_1q_2-q_3{}^2\cos^2 x_3} +\frac{(q_1q_3x_2{}^2\sin x_3+q_1q_5 g\cos(x_1+x_3))}{q_1q_2-q_3{}^2\cos^2 x_3}-f_1(x) \tag{5}$$

$$N_1(x) = \frac{q_2}{q_1 q_2 - q_3^2 \cos^2 x_3}$$
$$N_2(x) = \frac{-(q_2 + q_3 \cos x_3)}{q_1 q_2 - q_3^2 \cos^2 x_3} \tag{6}$$

The system equations in presence of uncertainties and external disturbance can be written as:

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= f_1(x) + \Delta f_1(x) + [N_1(x) + \Delta N_1(x)]u + \eta_1 \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= f_2(x) + \Delta f_2(x) + [N_2(x) + \Delta N_2(x)]u + \eta_2
\end{aligned} \tag{7}$$

where η_1, and η_2 represents the external disturbance and friction of joints, $\Delta f_1(x)$, and $\Delta f_2(x)$ are model uncertainties and friction terms [11], [13].

Assuming that: $\left|\Delta N_1(x)\right| \le N_1(x)$, *and* $\left|\Delta N_2(x)\right| \le N_2(x)$

Now transforming the system representation to another form by defining new states $z = [z_1 \quad z_2 \quad x_{z3} \quad z_4]^T$ [11].

$$\begin{aligned}
z_1 &= x_1 - \int_0^{x3} \frac{N_1(x) + \Delta N_1(x)}{N_2(x) + \Delta N_2(x)} ds \\
z_2 &= x_2 - \frac{N_1(x) + \Delta N_1(x)}{N_2(x) + \Delta N_2(x)} x_4 \\
z_3 &= x_3 \\
z_4 &= x_4
\end{aligned} \tag{8}$$

use Eq. (7), and Eq. (8) to obtain

$$\begin{aligned}
\dot{z}_1 &= z_2 + \frac{N_1}{N_2} z_4 - \frac{d}{dt} \int_0^{x3} \frac{N_1(s)}{N_2(s)} ds \\
\dot{z}_2 &= f_1(z) - \frac{N_1}{N_2} f_2(z) + \Delta f_1 - \frac{N_1}{N_2} \Delta f_2 - \frac{d}{dt}(\frac{N_1}{N_2}) z_4 + \eta_1 - \frac{N_1}{N_2} \eta_2 \\
\dot{z}_3 &= z_4 \\
\dot{z}_4 &= f_2(z) + \Delta f_2(z) + [N_2(x) + \Delta N_2(x)]u + \eta_2
\end{aligned} \tag{9}$$

The system equations can be written as:

$$\begin{aligned}
\dot{z}_i &= \bar{f}_i(z) + d_i(z) z_{i+1} + \gamma_i \\
\dot{z}_4 &= \bar{f}_4(z) + d_4(z) u + \gamma_4
\end{aligned} \tag{10}$$

where i=1, 2, 3

$$\begin{aligned}
\bar{f}_1(z) &= \frac{N_1}{N_2} z_4 - \frac{d}{dt} \int_0^{x3} \frac{N_1(s)}{N_2(s)} ds \\
d_1(z) &= 1 \\
\gamma_1 &= 0,
\end{aligned} \tag{11}$$

$$\begin{aligned}
\bar{f}_2(z) &= f_1(z) - \frac{N_1}{N_2} f_2(z) - \frac{d}{dt}(\frac{N_1}{N_2}) z_4 - z_3 \\
d_2(z) &= 1, \qquad \gamma_2 = \Delta f_1 - \frac{N_1}{N_2} \Delta f_2 + \eta_1 - \frac{N_1}{N_2} \eta_2,
\end{aligned} \tag{12}$$

$$\begin{aligned}
\bar{f}_3(z) &= 0 \\
d_3(z) &= 1, \qquad \gamma_3 = 0
\end{aligned} \tag{13}$$

$$\begin{aligned}
\bar{f}_4(z) &= f_2(z) \\
d_4(z) &= N_2(z) \qquad \gamma_4 = \Delta f_2(z) + \Delta N_2(z) u + \eta_2
\end{aligned} \tag{14}$$

To design a straightforward controller where mismatched uncertainties are exist the unknown terms γ_2, γ_4, have been estimated using a proposed adaptive Elman neural network Controller (AENN), the objective is to design controller such that z_1, z_3 track the desired trajectories z_{1d}, z_{3d}.

III. CONTROLLER DESIGN

1) *Description of Backstepping controller*

The system described in Eq. (10) has an output y = $(z_1, z3)^T$, and the desired output $y_d(t) = (z_{1d}, z_{3d})^T$.

define error functions:

$$e_{z1} = z_1 - z_{1d}$$

$$e_{zi} = z_i - z_{id} - \beta_{i-1} \qquad , \quad i = 1,2,3 \tag{15}$$

where β_i is a virtual control, and z_{id} is the desired trajectory for z_i

use Eq. (10) and Eq. (15) to derive $\dot{e}_{z1}$

$$\begin{aligned}
\dot{e}_{z1} &= \dot{z}_1 - \dot{z}_{1d} = \bar{f}_1 + d_1 z_2 - \dot{z}_{1d} \\
\dot{e}_{z1} &= \bar{f}_1 + d_2(e_{z2} + z_{2d} + \beta_1) - \dot{z}_{1d}
\end{aligned} \tag{16}$$

now chose $\beta_1 = \frac{1}{d_1}(-k_1 e_{z1} - g_1 + \dot{z}_{1d} - d_1 z_{2d})$ (17)

$$\dot{e}_{z1} = \bar{f}_1 + d_1 e_{z2} + z_{2d} d_1 - k_1 e z_1 - g_1 + \dot{z}_{1d} - d_1 z_{2d} - \dot{z}_{1d}$$

$$\dot{e}_{z1} = \bar{f}_1 + d_1 e_{z2} - k_1 e_{z1} - g_1 \tag{18}$$

similarly, choose

$$\beta_i = \frac{1}{d_i}(-k_i e_{zi} - g_i - d_{i-1} e_{zi-1} - \ddot{\gamma}_i + \dot{z}_{id} + \dot{\beta}_{i-1} - d_i z_{(i+1)d}), \qquad i = 2,3 \tag{19}$$

$$\beta_2 = \frac{1}{d_2}(-k_2 e_{z2} - g_2 - d_1 e_{z1} - \hat{\gamma}_2 + \dot{z}_{2d} + \dot{\beta}_1 - d_2 z_{3d})$$

$$\begin{aligned} \dot{e}_{z2} &= \bar{f}_2 + d_2 e_{z3} + d_2 z_{3d} - k_2 e_{z2} - g_2 - d_1 e_{z1} \\ &- \hat{\gamma}_2 + \dot{z}_{2d} + \dot{\beta}_1 - d_2 z_{3d} + \gamma_2 - \dot{z}_{2d} - \dot{\beta}_1 \end{aligned} \tag{20}$$

$$\dot{e}_{z2} = -k_2 e_{z2} + \bar{f}_2 - g_2 + d_2 e_{z3} - d_1 e_{z1} + (\gamma_2 - \hat{\gamma}_2) \tag{21}$$

by the same way

$$\dot{e}_{z3} = \dot{z}_3 - \dot{z}_{3d} - \dot{\beta}_2 = \bar{f}_3 + d_3 z_4 + \gamma_3 - \dot{z}_{3d} - \dot{\beta}_2 \tag{22}$$

use Eq .(10) and derivate of Eq. (15) to obtain $\dot{e}_{z4}$

$$\begin{aligned} \dot{e}_{z4} &= \dot{z}_4 - \dot{z}_{4d} - \dot{\beta}_3 \\ \dot{e}_{z4} &= \bar{f}_4 + d_4 u + \gamma_4 - \dot{z}_{4d} - \dot{\beta}_3 \end{aligned} \tag{23}$$

where g_i is a function to be defined using designer

2) Description of the proposed Elman Neural Network

Estimating the unknown system uncertainties is performed using AENN, The adaptive Elman Neural Network structure shown in fig .(2) shows that ENN is a multi-layered neural network with different layers of neurons containing input layer, hidden layer, context layer and output layer.

Layer1: Input layer

In this layer no function is being transformed ,input layer nodes only transmits input variables directly to hidden nodes , input variables here are $e_i = 1,2,3, and\ 4$

$$\text{where :} \quad \begin{aligned} e_1 &= q_1 - q_{1d}, \\ e_2 &= \dot{q}_1 - \dot{q}_{1d} + \lambda_1 e_1 \\ e_3 &= q_2 - q_{2d}, \\ e_4 &= \dot{q}_2 - \dot{q}_{2d} + \lambda_2 e_2 \end{aligned} \tag{24}$$

where λ_1 and λ_2 are positive constants.

Layer2: hidden layer

in this layer the hidden neurons have a Gaussian activation function as described in the following equation:

$$net_j = \sum_{i=1}^{n} \frac{(e_i - c_{ij})^2}{\sigma_{ij}{}^2} + \sum_{k=1}^{m} r_{kj} * L_k \tag{25}$$

$$R_j = \exp(-net_j), j = 1,2,......., m \tag{26}$$

where c_{ij}, σij are the adjustable parameters of the j[th] hidden neuron for the input variable, L_k is the output of the k[th] context neuron and r_{kj} is the connective weigh of the k[th] context neuron of the j[th] hidden neuron

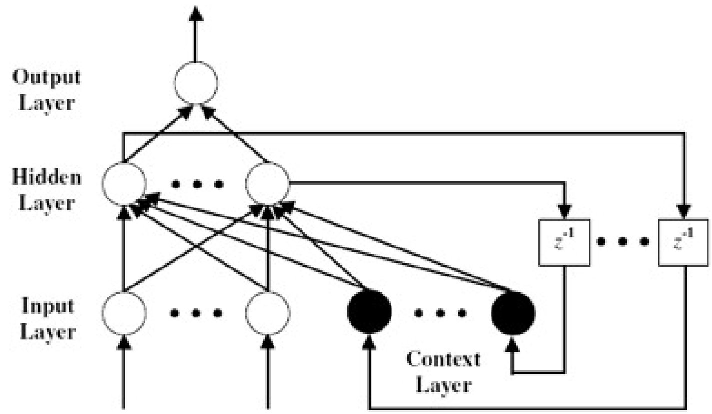

Fig. 2.Network structure of AENN

Layer3: context layer

context layer nodes as shown in fig. (2), receives its input from both the previous time hidden layer nodes and the previous time context layer nods which enable memorization as the following equation.

$$L_k = L_k^p + R_k^p, k = 1,2,......., m \tag{27}$$

Where L_k^p and R_k^p are the output signals of the K[th] context neuron and the K[th] hidden neuron in the previous time

Layer 4: Output layer

The output of the hidden layer nods is calculated as in Eq. (28):

$$\gamma nn = \sum_{j=1}^{m} \omega j Rj \tag{28}$$

The Elman Neural network (ENN) output can be represented in a vector form $\quad \gamma nn = \omega^T \phi(c, \sigma, r) \quad$ (29)

where $\omega = [\omega_1, \omega_2, ..., \omega_m]^T$, $L = [L_1, L_2, ..., L_m]^T$

, $c = [c_{11}, ..., c_{n1}, ..., c_{1m}, ..., c_{nm}]^T$

, $\sigma = [\sigma_{11}, ..., \sigma_{n1}, ..., \sigma_{1m}, ..., \sigma_{nm}]^T$

, and $\quad r = [r_{11}, r_{21}, ...r_{m1}, ...r_{1m}, r_{2m}, ..., r_{mm}]^T$

an ideal Enn γnn^* can be obtained as follows:

$$\gamma_4 = \gamma nn^* + \Delta = \omega^{*T} L(c^*, \sigma^*, r^*) + \Delta \tag{30}$$

Where Δ is the approximation error, ω^* and L^* are the optimal parameter vectors of ω and L, also c^*, σ^* and r^* are the optimal parameters of c, σ, and r.

For best approximation an estimated ENN $\hat{\gamma}_{nn}$ should be determined to overcome the problem not determining the optimal parameter vectors.

$$\hat{\gamma} nn = \hat{\omega}^T L(\hat{c}, \hat{\sigma}, \hat{r}) = \hat{\omega}^T \hat{L} \tag{31}$$

Where $\hat{\omega}$ and $\hat{L}$ are the estimated parameter vectors of ω, L and $\hat{c}, \hat{\sigma}, \hat{r}$ are estimated adaptive parameter vectors of c, σ *and* r. The estimation error is obtained as follows:

$$\tilde{\gamma} = \gamma_4 - \hat{\gamma} nn = \omega^{*T} L^* - \hat{\omega}^T \hat{L} + \Delta$$
$$\tilde{\gamma} = \tilde{\omega}^T \hat{L} + \hat{\omega}^T \tilde{L} + \tilde{\omega}^T \tilde{L} + \Delta \tag{32}$$

Where $\tilde{\omega} = \omega^* - \hat{\omega}$ and $\tilde{L} = L^* - \hat{L}$

Taking expansion of $\tilde{L}$ in a Taylor series obtaining

$$\tilde{L} = A^T \tilde{c} + Y^T \tilde{\sigma} + D^T \tilde{r} + O \tag{33}$$

where $\tilde{c} = c^* - \hat{c}$, $\tilde{\sigma} = \sigma^* - \hat{\sigma}$, $\tilde{r} = r^* - \hat{r}$, O is a vector of high order terms and

$$A = [(\frac{\partial L_1}{\partial c}), \ldots\ldots\ldots, (\frac{\partial L_n}{\partial c})] \mid c = \hat{c}$$

$$Y = [(\frac{\partial L_1}{\partial \sigma}), \ldots\ldots\ldots, (\frac{\partial L_n}{\partial \sigma})] \mid \sigma = \hat{\sigma}$$

$$D = [(\frac{\partial L_1}{\partial r}), \ldots\ldots\ldots, (\frac{\partial L_n}{\partial r})] \mid r = \hat{r}$$

Substituting (33) into (32)

$$\tilde{\gamma} = \tilde{\omega}^T \hat{L} + \tilde{\omega} T(A^T \tilde{c} + Y^T \tilde{\sigma} + D^T \tilde{r} + O) + \tilde{\omega}^T \tilde{L} + \Delta$$
$$\tilde{\gamma} = \tilde{\omega}^T \hat{L} + \tilde{c}^T A \hat{\omega} + \tilde{\sigma}^T \gamma \hat{\omega} + \tilde{r}^T D \hat{\omega} + \varepsilon \tag{34}$$

3) Description of the adaptive backstepping Elman neural network-based controller (BSAENN)

The (BSAENN) shown in fig. (3) with a system control law as follow:

$$U = \frac{1}{d_4}(-k_4 e_{z4} - g_4 + \dot{\beta}_3 - d_3 e_{z3} + \dot{z}4d - \hat{\gamma}_4) \tag{35}$$

$$U = \frac{1}{d_4}(U_1 - \hat{\gamma}_4)$$

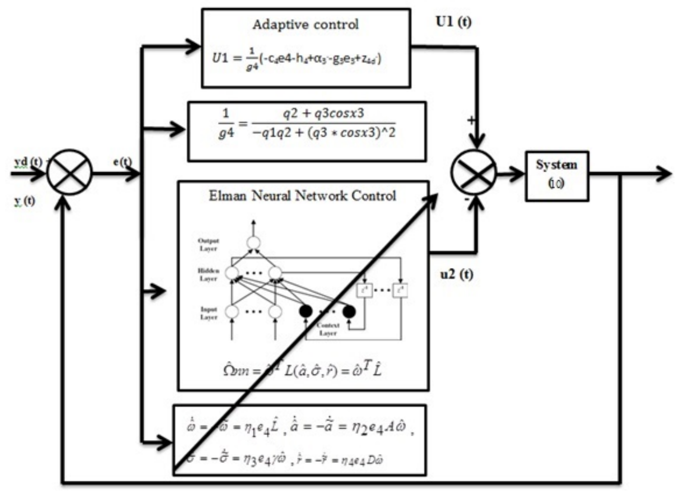

Fig. 3 .Block diagram of designed controller (BSAENN)

Where $\hat{\gamma}_4$ denotes the estimation of γ_4, The Elman neural network $\tilde{\gamma}_{nn}$ is designed to estimate the dynamics and uncertainties $\hat{\gamma}_4$

System stability can be guaranteed using the following Lyapunove function

$$V(t) = \frac{1}{2}\sum_{i=1}^{4} e_{zi}^2 + \frac{\tilde{w}^T \tilde{w}}{2b_1} + \frac{\tilde{c}^T \tilde{c}}{2b_2} + \frac{\tilde{\sigma}^T \tilde{\sigma}}{2b_3} + \frac{\tilde{r}^T \tilde{r}}{2b_4} + \frac{\tilde{E}^2}{2b_5} \tag{36}$$

Where: $\tilde{E} = E - \hat{E}$, is the bound estimated error, and b_1, b_2, b_3, b_4 *and* b_5 are positive learning constants.

$$\dot{V}(t) = e_{z1}\dot{e}_{z1} + e_{z2}\dot{e}_{z2} + e_{z3}\dot{e}_{z3} + e_{z4}\dot{e}_{z4} + \frac{\tilde{w}^T \dot{\tilde{w}}}{b_1} + \frac{\tilde{c}^T \dot{\tilde{c}}}{b_2} + \frac{\tilde{\sigma}^T \dot{\tilde{\sigma}}}{b_3} + \frac{\tilde{r}^T \dot{\tilde{r}}}{b_4} + \frac{\tilde{E}\dot{\tilde{E}}}{b_5} \tag{37}$$

Substitution in Eq. (37) from Eq. (16,21, 22, 23)

$$\begin{aligned} \dot{V}(t) &= e_{z1}(\bar{f}_1 + d_1 e_{z2} - k_1 e_{z1} - g_1) + \\ & e_{z2}(\bar{f}_2 + d_2 e_{z3} - k_2 e_{z2} - g_2 - d_1 e_{z1} + \gamma_2 - \hat{\gamma}_2) \\ & + e_{z3}(\bar{f}_3 + d_3 e_{z4} - k_3 e_{z3} - g_3 - d_2 e_{z2}) \\ & + e_{z4}(\bar{f}_4 + U_1 - \hat{\gamma}_4 + \gamma_4 - \dot{z}_4 - \dot{\beta}_3) \\ & + \frac{\tilde{w}^T \dot{\tilde{w}}}{b_1} + \frac{\tilde{c}^T \dot{\tilde{c}}}{b_2} + \frac{\tilde{\sigma}^T \dot{\tilde{\sigma}}}{b_3} + \frac{\tilde{r}^T \dot{\tilde{r}}}{b_4} + \frac{\tilde{E}\dot{\tilde{E}}}{b_5} \end{aligned} \tag{38}$$

Substituting control input in Eq. (35) and sum manipulations give:

$$\dot{V}(t) = -\sum_{i=1}^{4} k_i e_{zi}^2 + \sum_{i=1}^{4} e_{zi}(\bar{f}_i - g_i) + \sum_{i=1}^{4} e_{zi}(\gamma_i - \hat{\gamma}_i)$$

$$+e_{z4}(\tilde{\omega}^T \hat{L} + \tilde{c}^T A\hat{\omega} + \tilde{\sigma}^T \gamma\hat{\omega} + \tilde{r}^T D\hat{\omega} + \varepsilon) \tag{39}$$

$$+\frac{\tilde{w}^T \dot{\tilde{w}}}{b_1} + \frac{\tilde{c}^T \dot{\tilde{c}}}{b_2} + \frac{\tilde{\sigma}^T \dot{\tilde{\sigma}}}{b_3} + \frac{\tilde{r}^T \dot{\tilde{r}}}{b_4} + \frac{\tilde{E}\dot{\tilde{E}}}{b_5}$$

$$\dot{V}(t) = -\sum_{i=1}^{4} k_i e_{zi}^2 + \sum_{i=1}^{4} e_{zi}(\bar{f}_i - g_i) + \sum_{i=1}^{4} e_{zi}\varepsilon_i +$$

$$\tilde{\omega}^T (e_4 \hat{L} + \frac{\dot{\tilde{\omega}}}{b_1}) + \tilde{c}^T (e_4 A\hat{\omega} + \frac{\dot{\tilde{c}}}{b_2}) + \tag{40}$$

$$\tilde{\sigma}^T (e_4 \gamma\hat{\omega} + \frac{\dot{\tilde{\sigma}}}{b_3}) + \tilde{r}^T (e_4 D\hat{\omega} + \frac{\dot{\tilde{r}}}{b_4}) + \frac{\tilde{E}\dot{\tilde{E}}}{b_5}$$

here the tuning laws parameter can be selected as:

$$\dot{\hat{\omega}} = -\dot{\tilde{\omega}} = b_1 e_4 \hat{L} \tag{41}$$

$$\dot{\hat{c}} = -\dot{\tilde{c}} = b_2 e_4 A\hat{\omega} \tag{42}$$

$$\dot{\hat{\sigma}} = -\dot{\tilde{\sigma}} = b_3 e_4 Y\hat{\omega} \tag{43}$$

$$\dot{\hat{r}} = -\dot{\tilde{r}} = b_4 e_4 D\hat{\omega} \tag{44}$$

hence:

$$\dot{V}(t) = -\sum_{i=1}^{4} k_i e_{zi}^2 + \sum_{i=1}^{4} e_{zi}(\bar{f}_i - g_i) + \frac{\tilde{E}\dot{\tilde{E}}}{b_5} + \sum_{i=1}^{4} e_i \varepsilon_i \tag{45}$$

taking: $\varepsilon^2 = 2s_i \succ 0$

$$,\ e_i \bar{f}_i \le s_i e_i^2 |\bar{f}_i|^2 + \frac{1}{4s_i}$$

$$\dot{V} \le -\sum_{i=1}^{4} k_i e_i^2 + \sum_{i=1}^{4} \frac{1}{4s_i} + \sum_{i=1}^{4} s_i e_i^2 |\bar{fi}|^2$$

$$-\sum_{i=1}^{4} e_i g_i + \sum_{i=1}^{4} e_i \varepsilon_i \tag{46}$$

chosee: $g_i = s_i e_i |\bar{f}_i|^2$

So: $$\dot{V} \le -\sum_{i=1}^{4} k_i e_i^2 + \frac{1}{4s_i} + e_i \varepsilon \tag{47}$$

Using: $$-k_i e_i^2 + |e_i||\varepsilon_i| \le \frac{-1}{2}(k_i e_i^2 - \frac{\varepsilon_i^2}{c_{e_i}}) \tag{48}$$

$$\dot{V} \le -\sum_{i=1}^{4} K_i e_i^2 + \frac{1}{4s_i} + e_i \varepsilon \tag{49}$$

$$\dot{V} \le \frac{-1}{2} k_i e_i^2 + \frac{\varepsilon_i^2}{2p_i} + \frac{1}{4s_i} \tag{50}$$

Hence: $$\dot{V} \le -c_e V + \lambda \tag{51}$$

$$c_e = \min\{c_{e1}, \ldots\ldots\ldots, c_{en}\} \tag{52}$$

IV. SIMULATION RESULTS

To validate the effectiveness of the presented control algorithm, the BEASNN control methodology is tested for the trajectory tracing control of two-link robotic system shown in Fig. 2. The dynamical equations of the system (Eq. (1)) are given by

$$B(q) = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \tag{53}$$

$$C(q,\dot{q}) = \begin{bmatrix} c_{11}(q,\dot{q}) & c_{12}(q,\dot{q}) \\ c_{21}(q,\dot{q}) & c_{22}(q,\dot{q}) \end{bmatrix} \tag{54}$$

$$G(q) = \begin{bmatrix} g_1(q) \\ g_2(q) \end{bmatrix} \tag{55}$$

where

$$B_{11} = m_1 d_{l1}^2 + m_2 l_1^2 + I_1 + m_2 dl_2^2 + I_2 + 2m_2 l_1 d_{l2}(\cos(q_2)) \tag{56}$$

$$B_{12} = m_2 dl_2^2 + I_2 + m_2 l_1 d_{l2}(\cos(q_2)) \tag{57}$$

$$B_{21} = m_2 dl_2^2 + I_2 + m_2 l_1 d_{l2}(\cos(q_2)) \tag{58}$$

$$B_{22} = m_2 dl_2^2 + I_2 \tag{59}$$

$$c_{11} = -q_2 \times m_2 d_{l2} \sin(q_2) \tag{60}$$

$$c_{12} = -(\dot{q}_1 + \dot{q}_2) \times m_2 d_{l2} \sin(q_2) \tag{61}$$

$$c_{21} = \dot{q}_1 \times m_2 d_{l2} \sin(q_2) \tag{62}$$

$$c_{22} = 0 \tag{63}$$

$$g_1 = m_1 d_{l1} + m_2 l1 \cos(q_1) + m_2 d_{l2} \cos(q_1 + q_2) \tag{64}$$

$$g_2 = m_2 d_{l2} \cos(q_1 + q_2) \tag{65}$$

Where i=1,2,mi denotes the mass, li denotes the length, Ii denotes the mass moment of inertia about the center of mass, and d_{li} denotes the distance to the center of mass [1].

The BSAENN is conducted to illustrate the effectiveness of the proposed control algorithm. The simulation was performed in case of desired angular position for q_1 *and* q_2 where:

$$q_{1d} = (\pi/6) \times (1 - \cos(1.5 \times \pi \times \mathrm{t}) + \sin(\pi \times \mathrm{t})) \tag{66}$$

$$q_{2d} = (\pi/6) \times (1 - \cos(2 \times \pi \times \mathrm{t}) + \sin(1.5 \times \pi \times \mathrm{t})) \tag{67}$$

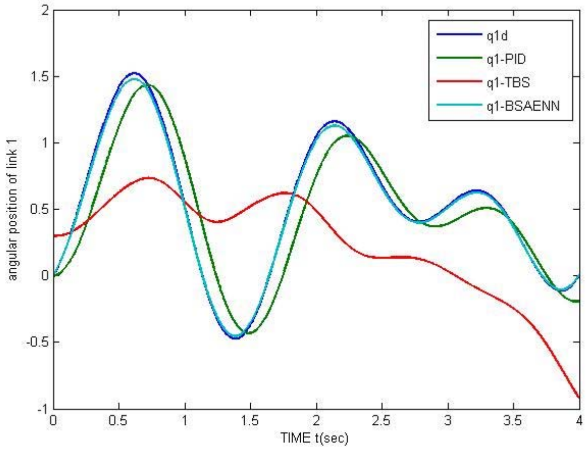

Fig. 4 Angular position response for q_{1d}

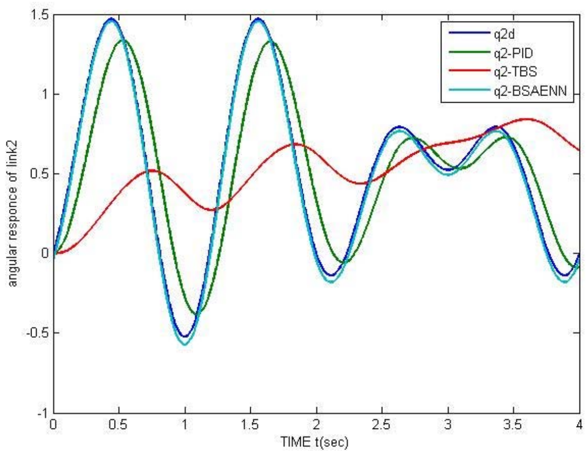

Fig. 5 Angular position response for q_{2d}

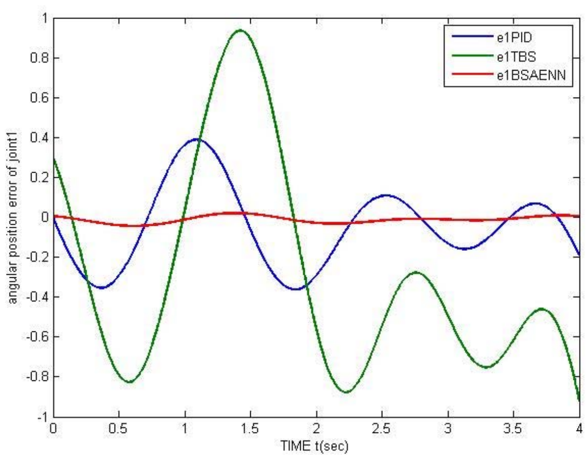

Fig. 6 Angular position error for joint1

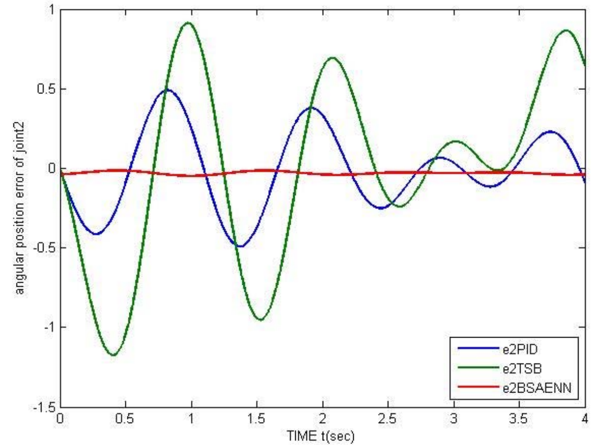

Fig. 7 Angular position error for joint2

From the simulation results (as shown in Fig. (4), Fig. (5), Fig. (6), and Fig (7)), it can be noted that a best performance can be achieved. It can be seen that the proposed algorithm (BAENN) gives a better trajectory tracking behavior. A comparison of tracking response characteristics upon the different control techniques with the performance indices calculated for the trajectory tracking control, for both joints, is summarized in Table I. The table shows that the presented (BSAENN) control system provides a superior control performance with the smallest error.

The performance indices are integral squared error (ISE) and integral time multiplied absolute error (ITAE) given by:

$$ISE = \int e^2(t)dt$$
$$ITAE = \int t\left|e(t)\right|dt \quad (68)$$

	Performance indices			
	ISE		***ITAE***	
	Joint	**Joint2**	**Joint1**	**Joint2**
PID	167.9576	251.9973	1.0234e+03	1.2884e+03
TBS	1.426e+03	1.318e+03	4.5778e+03	3.0853e+03
BSAENN	1.7067	4.1918	113.7693	256.6165

Table.1 Performance indices for ISE and ITAE

V. CONCLUSION

In this paper, a proposed backstepping controller based on an adaptive Elman neural network (BSAENN) system that adopted using the combination of the Elman neural network (ENN), which is a recurrent multilayered neural network, and the traditional backstepping control (TBS) approach has been used to control the nonlinear underactuated robot, the proposed Elman neural network (ENN)–based approximator with the advantage of faster convergence rate can handle dynamic response faster. The backstepping control is a recursive control method which has the ability to prevent useful nonlinearities from being canceled .To overcomes the drawback that the system has an input and two subsystems, a transformation from underactuated form to a classic cascade form has been made. The proposed controller has applied to a nonlinear underactuated robotic system. The stability of the closed-loop system are proved using Lyapunov stability. Numerical simulations with dynamical model of the two link robot, compared to conventional controllers (PID and TBS), show that the proposed controller provides robustness for trajectory tracking performance under the occurrence of uncertainties.

References

1. Albahkali, T., Mukherjee, R., Das, T., “Swing-up control of the pendubot an impulse–momentum approach”, IEEE Trans, Robot, vol.25, no. (4), pp. 975–982, 2009.
2. Chun-Fei Hsu, “Adaptive backstepping Elman-based neural control for unknown nonlinear system”, Neurocomputing , vol. 136, pp. 170-179, 2014.
3. Fantoni, I., Lozano, R., Spong, M.W. , ”Energy based control of the Pendubo”, IEEE Trans. Autom. Control, vol.45, no. (4), pp. 725-729, 2000.
4. H.J. Shieh, K.K. Shyu,, ” Nonlinear sliding-mode torque control with adaptive backstepping approach for induction motor drive”, IEEE Trans. Ind. Electron, vol. 46, no. (2), pp. 380–389, 1999.
5. Ichida, K., Izumi, K., Watanabe, K., Uchida, N, “Control of three-link underactuated manipulators using a switching method of fuzzy energy regions”, Artif. Life Robot, vol. 12, no. (1–2), pp.258–263, 2008.
6. J. Wu, W. Chen, D. Zhao, J. Li, “Globally stable direct adaptive backstepping NN control for uncertain nonlinear strict-feedback systems”, Neurocomputing, vol. 122, no.(25), pp.134–147, 2013
7. K.H. Cheng, “Self-structuring fuzzy-neural backstepping control with a Bspline-based compensator”, Neurocomputing, vol. 117, no.(6), pp.138–149, 2013.
8. Kulkarni, A., Kumar, A. , ”Adaptive backstepping control for uncertain underactuated systems with input constraints”, Proced. Eng., vol. 38, pp. 1001–1010, 2012.
9. L. Cheng, Z.G. Hou, M. Tan, “Adaptive neural network tracking control for manipulators with uncertain kinematics, dynamics and actuator model”, Automatica, vol. 4, no. (10), pp. 2312–2318, 2009.
10. M. M. Azimi·H. R. Koofigar..”Adaptive fuzzy backstepping controller design for uncertain underactuated robotic systems”, Nonlinear Dyn, vol. 79, pp.1457–1468, 2015 .
11. Olfati-Saber, R. “Normal forms for underactuated mechanical systems with symmetry”, IEEE Tran. Autom. Control, vol. 47, no. (2), pp. 305–308, 2002.
12. Q. Zhu, T. Zhang, S. Fei, K. , Zhang, T. Li, ” Adaptive neural control for a class of output feedback time delay nonlinear systems”, Neurocomputing , vol..72 , no. (7), pp.1985–1992, 2009.
13. Ravichandran, M.T., Mahindrakar, A.D.,”Robust stabilization of a class of underactuated mechanical systems using time scaling and Lyapunov redesign” , IEEE Trans.Ind.Electron, vol. 58, no.(9), pp.4299–4313 , 2011.
14. S. Tong, C. Liu, Y. Li, “Adaptive fuzzy backstepping output feedback control for strict feedback nonlinear systems with unknown sign of high-frequency gain”, Neurocomputing , vol. 77, no . (1), pp. 58–70, 2012.
15. Sankaranarayanan,V., Mahindrakar, A.D, ”Controlofaclass of underactuated mechanical systems using sliding modes”, IEEE Trans. Robot, vol. 25, no. (2), pp. 459–467, 2009.
16. Spong, M.,"Underactuated mechanical systems”, Control Probl. Robot. Autom, vol. 230, pp. 135–150, 1998.
17. Spong, M.W., Block, D.J.,”The pendubot a mechatronic system for control research and education.”, 34th IEEE Conference, Decision and Control, pp. 555–556, 1995.
18. Tsai, C.C., Huang, H.C., Lin, S.C” Adaptive neural networkcontrolofaself-balancingtwo-wheeled scooter”, IEEE Trans.Ind. Electron, vol. 57, no. (4), pp. 1420–1428, 2010.
19. Y. Gao, S. Tong, Y. Li, “Adaptive fuzzy backstepping output feedback control for a class of uncertain stochastic nonlinear system in pure-feedback form”, Neurocomputing, vol.122, no. (25), pp. 126–133, 2013.
20. Zhang, M., Tarn, T.-J, “Hybrid control of the Pendubot”, IEEE/ASME Trans. Mechatron, vol. 7, no.(1), pp. 79–86, 2002.