A novel auto regressive exogenous-local model network for modeling and controlling dynamic systems

Hamdi A. Awad and Tarek A. Mahmoud
Industrial Electronic and Control Eng. Dept., Faculty of Electronic Eng., Menoufia University, Menouf, Egypt

This paper introduces a novel neural network named Auto Regressive eXogenous Local Model (ARX-LM) network. It basically adopt the philosophy of forming a process be modeled with a set of fuzzy-wavelet submodels. Each submodel comprises a Takagi-Sugeno-Kang (TSK) fuzzy model fertilized by a wavelet function. The outputs of these submodels are weighted and summed to produce the final output. The main feature of the proposed LM network is that its structure is an ARX model that inherits the simplicity of the conventional linear and non-linear control systems, the learning capabilities of neural networks, the transparency of fuzzy systems and the locality of wavelet networks. The former means that the Lyapunove's direct method can be employed to check the stability of the proposed ARX-LM network. The latter three features make the proposed network, powerful, transparent, and plastic. The plastic feature signifies that a new pattern can be assigned to uncommitted cluster if this pattern does not match previously stored fuzzy models. It is inherited from employing the fuzzy Adaptive Resonance Theory (ART) to form that local submodel. The parameters of the proposed network are adapted using the Recursive Least Square (RLS) algorithm. The soundness of the proposed ARX- LM network is tested in modeling and controlling non-linear dynamical systems. The proposed ARX-LM network is employed to develop a long range predictive control scheme for SISO and MIMO time varying medical systems.

يقدم هذا البحث شبكة عصبية جديدة تسمى ARX-LM. تتبنى هذه الشبكة بشكل اساسى فلسفة نمذجة الانظمة الديناميكية بمجموعة من النماذج الفرعيه الغيميه المخصبة بدوال المويجة. حيث يشتمل كل نموذج فرعى على نموذج غيمى (TSK) مطعم بدالة مويجه. وبوزن و تجميع خروج النماذج الفرعية يتم الحصول على الخرج النهائى للشبكة. ان السمة الاساسية لهذه الشبكة المقترحة هى ان هيكلها من النموذج ARX الذى يرث بساطة نظم التحكم الخطية و الاخطية التقليدية, المقدرة التعليمية للشبكات العصبية, شفافية النظم الغيمية و محدودية الشبكات المويجية . يعنى الاول ان طريقة Lyapunove يمكن توظيفها لفحص اتزان الشبكة المقدمة. أما الثلاث خصائص الاخيرة تجعل الشبكة المقدمة , قوية, شفافة, تمددية. حيث تعنى الاخيرة انه تتقبل الشبكة الزيادة فى هيكلها و ذلك فى حالة وجود عينة من الدخل لايمكن تسكينها فى النماذج الغيمية المخزنة مسبقا. هذه الخاصية اكتسبت من توظيف ART الغيمى والذى استخدم لتشكيل هذه النماذج الفرعية. كذلك تكييف متغيرات الشبكة المقترحة باستخدام خوارزم RLS. قوة الشبكة المقترحة تم اختبارها فالنمذجة من خلال انظمه ديناميكية لا خطية. بالاضافة الى ذلك تم بناء نظم تحكم تنبائى ليس فقط للنظم الطبية الديناميكية أحادية الدخل و الخرج و لكن ايضا للنظم الطبية الديناميكية متعددة الدخل و الخرج. ويمكن تلخيص اسهامات هذا البحث كالاتى: * بناء شبكة عصبية جديدة تسمى ARX-LM. * توظيف خوارزمات ART , RLS لبناء وتعليم الشبكة المقترحة.* اختبار مدى اتزان الشبكة المقترحة باستخدام طريقة Lyapunove.* بناء نظم تحكم تنبائى يعتمد على الشبكة المقترحة.

Key words: Wavelet neural networks, Model-based predictive control, Fuzzy ART algorithms, The RLS algorithm, Medical systems

1. Introduction

Fuzzy neural schemes mainly divided into two models, Mamadani [1] and TSK [2]. The latter replaced the linguistic expression of the consequent in the former by a set of linear equations. A self-constructing fuzzy interface network with an on-line learning ability was introduced based on the TSK model [3]. Lee and Tang proposed fuzzy neural network based on the TSK model [4]. Despite these modifications, the TSK model still lacks the plasticity feature and thus will forget previously learned information when presented with sustained sequences of new data. A set of neural networks are structured based on the concepts of the wavelet [5-8]. There are two kinds of Wavelet Neural Networks (WNNs), one with fixed dilation and translation parameters, and the other with adjustable dilation and translation parameters. Both of the two types have adjustable weights at the output layer. A set of researches have been conducted to improve the capabilities of these networks for function approximation and control purposes [9-11]. Daniel et al. developed a fuzzy wavelet neural network (FWN) inspired by the theory of multi resolution analysis and TSK fuzzy model [12]. The proposed network comprises a set of TSK fuzzy rules and WNNs. The former determines the contributions of the latter. The output of these sub-models are weighted and summed to provide the final output. Although this structure succeeded in modeling purposes compared with WNN [5] and Adaptive Network based-Fuzzy Inference Systems (ANFIS) [13], it suffers from a set of limitations. First, using a set of WNNs, increase the complexity of the network. Second, its stability has not yet been analyzed mathematically. Third, having a computation demand algorithm, limits the network application in modeling single input single output (SISO) processes. Hybrid fuzzy WNNs were proposed [14, 15] to simplify that network; however, their stability has not yet been proved mathematically.

This paper introduces an ARX local model network. It consists of a set of TSK fuzzy rules fertilized by wavelet functions. Each wavelet determines the contribution of the corresponding TSK fuzzy model. These sub-models are merged to provide the final output of the proposed network. Learning of the proposed LMN comprises two phases, structure learning and parameter learning. In the former, the fuzzy ART algorithm [16] is used to assign a new pattern to an uncommitted cluster if this pattern does not well match the generated sub-models. Using the fuzzy ART algorithm, makes the proposed network is a plastic network. The proposed network is an ARX model, which enables us to analyses the stability of the network using the Lyapunove's direct method with ease. Fertilizing the proposed network with the wavelet function makes it very useful for function approximation purposes. Finally, the RLS identification method is used to identify the parameters of the proposed network.

The motivations of this paper can be concluded as follows:

- Development a novel version of local model network named ARX-LM network.
- Employing the ART and the RLS algorithms to self-organize and learn the proposed network.
- Testing the stability of the proposed network using Lyapunove method.
- Development a model-based predictive control scheme based on the proposed network, named ARX-LM–based predictive control.

This paper can be organized as follows. Section II develops the proposed ARX local model network. It describes the network structure, stability, and its application for function approximation. Section III employs the proposed network for control purposes. Section V concludes the topics discussed in this paper.

2. The ARX- LM network

This section briefly reviews the proposed ARX-LM network. This network is applied to simulate nonlinear plants.

2.1. The structure of the ARX-LM network

Fig. 1 depicts the structure of the proposed ARX-LM network. It consists of five layers that are described as follows:

Layer -1: A node at this layer just transmits the input values to the next layer.

Layer -2: This layer consists of two groups, universes of discourse of the input fuzzy variables and their wavelets. The former cover the universes of discourse of the input variables by a set of triangular-shipped function $\mu_{A_j^i}(x_i)$.

That is:

$$\mu_{A_j^i}(x_i) = 1 - \frac{2\left|x_i - c_{ij}\right|}{\delta_{ij}}, \quad (1)$$

where A_j^i is the j^{th} fuzzy set of the i^{th} input variable x_i , and c_{ij}, and δ_{ij} are the center and width of this fuzzy set.

The latter is a wavelet function generated by dilating and translating the mother wavelet function $h(x)=(1-x^2)\exp(-\frac{x^2}{2})$. That is:

$$\Phi_j(X)=\prod_{k=1}^{n}h(Z_{jk}), \quad (2)$$

where $Z_{jk}=\frac{x_k-m_{jk}}{d_{jk}}$, n is the number of their inputs, m and d are the translation and dilation parameters respectively.

Layer -3: The firing strength can be obtained using Larsen's product [17] as follows:

$$\omega^i=\prod_{j=1}^{n}\mu_{A_j^i}(x_i), \quad (3)$$

where, its normalized value is:

$$\varpi^i=\frac{\omega^i}{\sum_{i=1}^{q}\omega^i}. \quad (4)$$

Layer -4: A node at this layer is a submodel that merges the normalized firing strength of a TSK fuzzy rule with a wavelet. That is:

$$y_i^4=\Phi_i(X)w^i. \quad (5)$$

Layer -5: Based on the approximate Center Of Area (COA) defuzzification method, the crisp output y_m can be deduced. That is:

$$y_m=\sum_{i=1}^{q}\varpi^i f_i(X)\Phi_i(X). \quad (6)$$

The network described above performs the following rule:

R^i: *IF* x_1 *is* A_1^i and … and x_n *is* A_n^i THEN
$y^i=f_i(X) * \Phi_i(X)$. (7)

where, $f_i(X)$ is a linear function of the TSK model. That is:

$$f_i(X)=w_1^i * x_1+w_2^i * x_2+. \; . \; .+w_1^i * x_n, \quad (8)$$

and, $\Phi_i(X)$ is the wavelet function defined in (2).

Reforming (8), results an ARX-LM defined below:

$$f_i(X)=w_{u1}^i * u(k-1)+...+w_{ur}^i * u(k-r)$$
$$+w_{y1}^i * y(k-1)+...+w_{ys}^i * y(k-s). \quad (9)$$

Substituting (9) in (6), results:

$$y_m(k)=b_1 * u(k-1)+...+b_r * u(k-r)$$
$$+a_1 * y(k-1)+...+a_s * y(k-s), \quad (10)$$

where,

$$b_i=\sum_{j=1}^{q}w_{ui}^j * \Phi_j(X)\varpi^j, \qquad i=1,2,\ldots,r$$

$$a_h=\sum_{j=1}^{q}w_{yh}^j * \Phi_j(X)\varpi^j, \qquad h=1,2,\ldots,s \quad (11)$$

q is the number of rules generated, and r and s are the orders of the plant input and output respectively. Eq. (10) represents the proposed ARX-LM with the input and output vectors defined below:

$\underline{X}$ = [u (k-1) u (k-2) . . . u (k-r) y (k-1) y (k-2) . . ., .y (k- s)] ^{T}

$\underline{Y}$=[y (k)] ^{T}

At this stage the ARX-LM network is developed for modeling and controlling dynamic processes. This is our first motivation.

2.2. Structure/parameter learning of the ARX-LM network

Learning of the proposed ARX-LM network consists of two phases, structure learning and parameter learning. The former structure is the corner stone to develop an optimal ARX-LM network. It should determine the optimal number of the following seeds

- Fuzzy clusters of each fuzzy variable.
- Wavelet nodes

• Fuzzy rules

The fuzzy ART algorithm [16] is employed to determine the above three seeds. Basically, the fuzzy ART was introduced to finding the parameters of an input membership function (the center "c_{ji}" and the width "δ_{ji}"). This is equivalent to forming the proper fuzzy hyper-boxes clusters in the input space, which defines the number of fuzzy rules and wavelet nodes. Forming these clusters requires the initial values of their centers and widths. Determining the parameters of a fuzzy cluster (a fuzzy set), includes the parameters of the corresponding wavelet function. That is

$$m_{ji} = \alpha_1 * c_{ji}, \qquad d_{ji} = \alpha_2 * v_{ji}, \tag{12}$$

where j =1,2, ... q (number of wavelet functions), i=1,2, . . . n (number of input variables), c_{ji}, v_{ji} are the center and the width of the j^{th} fuzzy set of the i^{th} input , and α_1, α_2 are scaling factors.

The fuzzy ART algorithm is employed in this paper to self-organize the proposed ARX-LM. It can be summarized as follows [16]:

Input vector: Each input X is n dimensional vector (x_1, ..., x_n), where each component x_i is in the interval [0, 1].

Weight vector: Each category (j) corresponds to a vector Wj = (w_{j1}, w_{j2}, ..., w_{jn}) of adaptive weights. The number of potential categories N= (j = 1, ..., N) is arbitrary. Where the weight vector is defined by $W_j = (c_{j1}, c_{j2}, . \ . \ ., c_{jn})$ Initially

$$w_{j1} = ... = w_{jn} = 1, \tag{13}$$

and each category is said to be uncommitted. Alternatively, initial weights w_{ji} may be taken greater than one. Larger weight w_{ji} bias the system against selection of uncommitted nodes, leading to deeper searches of previously coded categories.

After a category is selected for coding it becomes committed.

Parameters: Fuzzy ART dynamics are determined by a choice parameter $\eta > 0$; a learning rate parameter $\beta \in [0,1]$; and a vigilance parameter $\rho \in [0,1]$.

Category choice: For each input X and category j, the choice function T_j is defined by:

$$T_j = \frac{|X' \wedge W_j|}{\eta + |W_j|}, \tag{14}$$

where the fuzzy and operator ^ is defined by

$$(\mathbf{x} \wedge \mathbf{y})_i = \min(x_i, y_i), \tag{15}$$

and where the norm |.| is defined by

$$|X| = \sum_{i=1}^{n} x_i . \tag{16}$$

The category choice is indexed by ***J***, where

$$T_J = \max(T_j : j = 1,...N). \tag{17}$$

If more than one T_j is maximal, the category j with the smallest index is chosen. In particular, nodes become committed in order j=1, 2, 3, . . .

Resonance or rest: Resonance occurs if the match function of the chosen category meets the vigilance criterion. That is:

$$\frac{|X' \wedge W_j|}{|X'_j|} \geq \rho . \tag{18}$$

Learning then ensues, as defined in (20). Mismatch rest, on other hand, occurs if

$$\frac{|X' \wedge W_j|}{|X'_j|} < \rho . \tag{19}$$

Then the value of choice function is rest to zero for the duration of the input presentation to prevent its persistent selection during search.

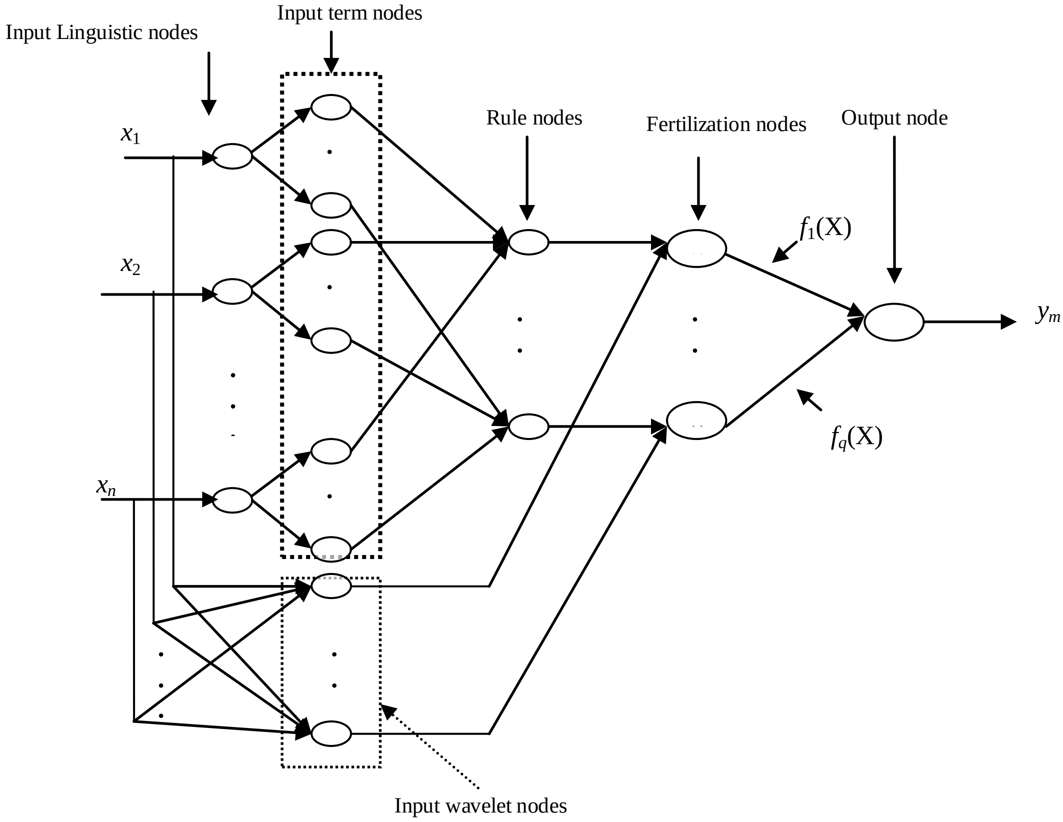

Fig. 1. The structure of the proposed ARX-LM network.

A new index ***J*** is chosen, by (17). The search process continues until the chosen category ***J*** satisfies (18)

Learning: The weight vector W_J is update according to the equation

$$W_J^{new} = X' \wedge W_J^{old} + (1-\beta)W_J^{old}. \quad (20)$$

Fast learning corresponds to setting $\beta = 1$.

Fast –Commit slow-record option: For efficient coding of noisy input sets, it is useful to set $\beta = 1$ when J is an uncommitted code, and then to take $\beta < 1$after the category is committed. Then $W_J^{new} = X$ the first time category J becomes active.

Input normalization option: A category proliferation problem can occur in some analog ART systems when a large number of inputs erode the norm of weight vectors. Proliferation of categories is avoided in fuzzy ART if inputs are normalized; that is, for some $\gamma > 0$,

$$|X| \equiv \gamma, \quad (21)$$

for all inputs X. Normalization can be achieved by preprocessing each incoming vector ***x***, for example setting

$$X = \frac{\mathbf{x}}{|\mathbf{x}|}. \quad (22)$$

An alternative normalization rule, called complement coding, achieves normalization while preserving amplitude information. Complement coding represents both on-response and off-response to ***x***. to define this operation in its simplest form, let ***x*** itself represent the on-response. The complement of ***x***, denoted by $\boldsymbol{x}^c$, represents the off-response, where

$$x_i^c = 1 - x_i. \quad (23)$$

The complement coded input X to the recognition system is the 2n-dimensional vector

$$X = (\mathbf{x}, \mathbf{x}^c) \equiv (x_1, . \ . \ ., x_n, x_1^c, . \ . \ ., x_n^c) . \quad (24)$$

Note that

$$|X| = |(\mathbf{x}, \mathbf{x}^c)| = \sum_{i=1}^{n} x_i + (n - \sum_{i=1}^{n} x_i^c) = n , \quad (25)$$

so inputs preprocessed into complement coding form are automatically normalized. Where complement coding is used, the initial condition (13) is replaced by

$$w_{j1} = . \ . \ . = w_{j2n} = 1 . \quad (26)$$

Basically, training the proposed ARX-LM network is a nonlinear optimization problem. Research has been conducted in nonlinear optimization methods; however, the best known method is the RLS method. This method can be summarized as follows. Suppose the weight and the input/output vectors are:

$$\theta_j = [w_1^j \quad w_2^j . \ . \ . \quad w_n^j]^T ,$$

$$v_j = \Phi_j \varpi^j [u(k). \ . \ . u(k-r) \quad y(k-1). \ . \ . \quad y(k-s)]^T ,$$

and

$$\beta_j = \Phi_j \varpi^j v_j , \quad (27)$$

where ϖ^j is the normalized value of the firing strength defined in (4), Φ_j is the wavelet function defined in (2), and j =1,2, . . . , q
Reforming (26), results:

$$\Theta = [\theta_1, \theta_2, . \ . \ ., \theta_q]$$

$$\mathrm{B} = [\beta_1, \beta_2, . \ . \ ., \beta_q]$$

That leads to:

$$y_m(k) = B * \Theta . \quad (28)$$

The linear parameters Θ are recursively estimated as defined below:

$$\Theta(k) = \Theta(k-1) + P(k-1)B(k)e(k)$$

$$P(k) = P(k-1) - \frac{P(k-1)B(k)B^T(k)P(k-1)}{1 + B^T(k)P(k-1)B(k)} , \quad (29)$$

where $e(k)$ is the error between the desired and the actual outputs respectively. That is:

$$e(k) = (Y_d(k) - Y(k)) . \quad (30)$$

Employing the ART and the RLS algorithms to develop the proposed ARX-LM network, results a simple self-organizing network for modeling and control dynamic systems. This is our second motivation.

2.3. Stability analysis of the ARX-LM network

Forming the proposed local model network in the ARX model, investigates its stability using Lyapunove's method with ease. This is our third motivation in this paper. The Lyapunove's direct method is employed to investigate the stability of a fuzzy model mathematically [18-20]. In a similar manner, the stability of the proposed network is analyzed as follows:
Definition (1): A matrix Q is positive definite matrix if all successive principle minors are positive
Definition (2): A matrix Q is positive semi definite matrix if it is singular and all successive principle minors are nonnegative.
Definition (3): A matrix Q is negative definite matrix if $-Q$ is positive definite
Theorem (1): The stability of a local model network is globally asymptotically stable if there exists a common positive definite matrix $\boldsymbol{Q}$ for all the subsystems such that $A_i^T Q A_i - Q$ is negative definite for $i \in \{1,2,..., q\}$
where $\boldsymbol{A}_i$ is the control canonical form of an individual local model and is defined as follows:

$$A_i = \begin{bmatrix} w_{r+1}^i & w_{r+2}^i & . & . & . & w_s^i \\ 1 & . & . & . & . & . \\ . & 1 & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & 1 & . \end{bmatrix}$$

2.4. Modeling simulations

In this section, two examples of nonlinear dynamical systems are tested to demonstrate the soundness of the proposed ARX-LM network in the system modeling. It investigates also the stability of the proposed network for the two systems.

Example one: Although the proposed ARX local network is aimed for modeling complex processes, it was first tested using a simple nonlinear system. The discrete time difference equation of the system is [8] is:

$$y(k) = \frac{24 + y(k-1)}{30} y(k-1) - 0.8 \frac{u(k-1)^2}{1 + u(k-1)^2} y(k-1) + 0.5u(k-1) . \quad (31)$$

The input and the output vectors are

$$X = [y(k-1) \;\; y(k-2) \;\; u(k)]^T .$$
$$Y_d = [y(k)]^T$$

The input training sequence consists of square pulses with random amplitude in the range [-5, 5] and with random frequencies. For comparison reasons, the following performance index (J_1) is computed for the proposed ARX local model network and the WNN published in [8].

$$J_1 = \frac{1}{N_t} \sum_{k=1}^{N_t} (Y_d(k) - Y_m(k))^2 . \quad (32)$$

The output of the network and the process is depicted in fig. 2. The performance index defined in (32) is computed for the two trained networks using the same test data. Table 1 depicts the error values obtained and the number of wavelets used. Using the parameters obtained from the proposed network results two local models and have two canonical control forms. Those are:

$$A_1 = \begin{bmatrix} 0.0516 & -0.102 \\ 1 & 0 \end{bmatrix}, \text{ and}$$
$$A_2 = \begin{bmatrix} 0.808 & -0.5618 \\ 1 & 0 \end{bmatrix}$$

suppose a positive definite matrix, $\boldsymbol{Q}$ defined below:

$$Q = \begin{bmatrix} 1 & -0.82 \\ -0.82 & 0.73 \end{bmatrix}.$$

To check the stability of the proposed network, the following two matrices are obtained from the proposed ARX-LM. Those are:

$$A_1^T Q A_1 - Q = \begin{bmatrix} -0.1827 & 0.9089 \\ 0.9089 & -0.7196 \end{bmatrix},$$
$$| A_1^T Q A_1 - Q | = -0.69$$
$$A_2^T Q A_2 - Q = \begin{bmatrix} -0.9420 & 0.8312 \\ 0.8312 & -0.4144 \end{bmatrix},$$
$$| A_2^T Q A_2 - Q | = -0.30 .$$

It is clear that both the two matrices are negative definite. According to the Lyapunove's method described above, the network is stable.

Example two: A more complex system is employed to assure the capabilities of the proposed network in the dynamic systems modeling. The discrete time difference equation of the system is [12]:

$$y(k+1) = \frac{\{y(k)y(k-1)y(k-2)u(k-1) * [y(k-2)-1] + u(k)\}}{1 + y^2(k-1) + y^2(k-2)} . \quad (33)$$

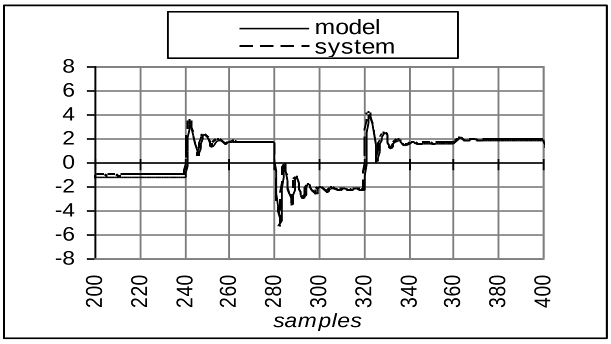

Fig. 2. The output of the system and the proposed ARX-LM network (example one).

Table 1
Comparison between the proposed ARX-LM network with the WNN published in [8] (Example one)

The two network	The J_1 error values	Number of wavelets
WNN	0.0302	5
ARX-LMN	0.0200	2

The training signal u(k) is defined below:

$$u(k) = 0.8\sin(2\Pi k/50)\,. \tag{34}$$

The input and the output vectors are:

$$X = [y(k)\;\; y(k-1)\;\; u(k)]^T \,.$$
$$Y_d = [y(k+1)]^T$$

Two different test signals u_1(k) and u_2(k), are used to assure the modeling capabilities of the proposed network. Those are [12]:

$$u_1(k) = \begin{cases} \sin(2\Pi k/250) & k \le 500 \\ 0.8\sin(2\Pi k/250) + 0.2\sin(2\Pi k/25) & k > 500 \end{cases}, \tag{35}$$

$$u_2(k) = 0.8\sin(2\Pi k/200) + 0.2\sin(2\Pi k/25)\,. \tag{36}$$

For the comparison reason, the following performance index is computed for the proposed ARX-LM network and the FWN published in [12]:

$$J_1 = \sqrt{\frac{\sum_{k=1}^{N_t}(Y_m(k) - Y_d(k))^2}{\sum_{k=1}^{N_t}(Y_d(k) - \hat{Y}_d)^2}}\,.$$

With

$$\hat{Y}_d = \frac{1}{N_t}\sum_{k=1}^{N_t} Y_d(k)\,. \tag{37}$$

where N_t is the number of training data, and $Y_m(k)$ is the model output at the k^{th} time step. The output of the network and the process using the two test signals defined in (35 and 36) is depicted in fig. 3 and 4 respectively. The performance index defined in (37) is computed for the two trained network using the test signals defined in (35 and 36). Table 2 depicts the error values obtained. Using the parameters obtained from the proposed network, results two local models that have two canonical control forms. Those are:

$$A_1 = \begin{bmatrix} 0.128 & -0.123 \\ 1 & 0 \end{bmatrix}.$$
$$A_2 = \begin{bmatrix} 0.229 & -0.0141 \\ 1 & 0 \end{bmatrix}.$$

To check the stability of the proposed network, the following two matrices are obtained from the proposed ARX-LM network by using the positive definite matrix Q used in the example one. Those are:

$$A_1^T Q A_1 - Q = \begin{bmatrix} -0.4635 & 0.9051 \\ 0.9051 & -0.7149 \end{bmatrix}.$$
$$|\, A_1^T Q A_1 - Q \,| = -0.44$$
$$A_2^T Q A_2 - Q = \begin{bmatrix} -0.5931 & 0.8443 \\ 0.8443 & -0.7283 \end{bmatrix}.$$
$$|\, A_2^T Q A_2 - Q \,| = -0.28$$

The two matrices are negative definite that satisfies the Lyapunove's condition and confirm the stability of the network.

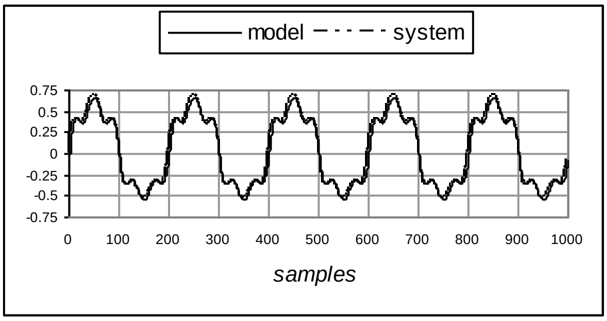

Fig. 3. The output of the system and the ARX -LM network using the test signal u_1.

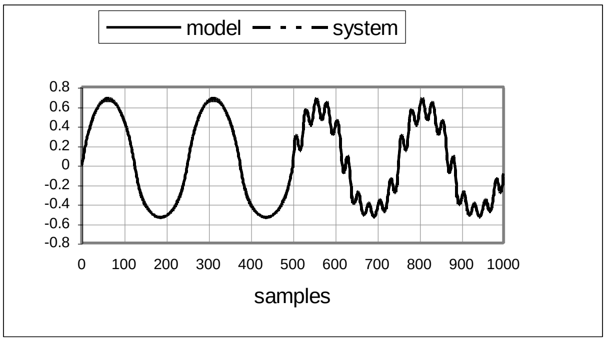

Fig. 4. The output of the system and the ARX-LM network using the test signal u_2.

Table 2
Comparison between the proposed ARX-LM network and the FWN published in [12] (Example two)

The two networks	The J_2 error values using u_1	The J_2 error values using u_2
The FWN	0.0406	0.0449
The proposed ARX-LM network	0.03102	0.03757

Investigation shows that best simulation results are obtained using the proposed ARX – LM network compared with the WNN published in [8] and the FWN published in [12], in the sense of the performance indexes defined in (32 and 37). Controlling dynamic systems using the proposed ARX-LM network is discussed in section III.

3. ARX-LM -based predictive control

Developing the proposed model in the form of ARX model helps the engineers to construct their model-based predictive control schemes with ease. This is our fourth motivation. The methodology of Model Predictive Control (MPC) can generally

Be summarized as follows [21, 22]. First, based on the model employed the future outputs for a specified prediction horizon are predicted at time k. Second, using an optimization scheme to minimize a cost function, the linear programming (LP) and nonlinear programming (NLP) problems can be solved and hence a set of optimal control actions can be obtained. Third, from these possible actions, the control system at time k is selected and sent to the actual process and the rest of the actions are rejected based on the receding horizon strategy. The MPC comprises on-step ahead or multi-step ahead strategies. The former suffers from some limitations particularly in controlling non-minimum phase systems. To overcome these shortcomings, the idea of using long-range predictive control has been developed. Since the early 1980s, a long-range predictive control has gained its popularity in control field [21, 22]. A number of algorithms of predictive control based on fuzzy models have been proposed [23-25]. Also, a number of researchers have been conducted using neural networks or fuzzy neural network to develop a model-predictive control scheme [26- 29]. A nonlinear predictive control based on wavelet neural networks and fuzzy wavelet neural networks has been presented in [30, 14]. Linkens and Kandiah introduced a long range predictive control scheme based on TSK fuzzy model that has been applied for controlling a nonlinear SISO system [24]. This paper extends this scheme for controlling not only SISO systems but also for controlling MIMO systems using the proposed ARX-LM network.

3.1. *ARX-LM- based long range predictive control*

First, consider a SISO discrete time system described by the proposed ARX local model network. The model prediction over the costing horizon n_2 is given by:

$y_p(t+1) = a_1 y(t) + \ldots + a_s\, y(t+1-s) + b_1\, u(t) + \ldots + b_r\, u(t-r+1) + err(t)$

.
.
.

$y_p(t+i) = a_1 y(t+i-1) + \ldots + a_s\, y(t+i-s) + b_1\, u(t+i) + \ldots + b_r\, u(t+i-r) + err(t),$ (38)

.
.
.

$y_p(t+n_2) = a_1 y(t+n_2-1) + \ldots + a_s\, y(t+n_2-s) + b_1\, u(t+n_2-1) + \ldots + b_r\, u(t-r+n_2) + err(t)$

where err(t) represents the modeling error, and $y_p(t+i)$ is the i^{th} predicted output. It has been assumed that the modeling error is constant over the entire prediction horizon and the values of
$u(t+m-1)$ is equal zero over the control horizon..

Accordingly, the above equations can be reformed as follows:

$$Y(t) = P\,X(t) + Q\,U(t) + R\,err(t), \tag{39}$$

where, $Y(t)=[y_p(t+1)\ \ldots\ y_p(t+n_2)]^T$, denotes a vector of the model predicted outputs over the prediction horizon, $X(t)=[y(t)\ \ y(t-1)\ \ldots\ y(t+1-s)\ \ u(t-1)\ \ldots\ u(t+1-r)]^T$ is a vector of the past plant and controller outputs, and $U(t)=[u(t)\ldots u(t+m-1)]^T$, is a vector of the future outputs of the controller. The matrices P, Q, and R are given below

$$P = \begin{bmatrix} p_{11} & \cdots & p_{1(s+r-1)} \\ \vdots & & \vdots \\ p_{n_2 1} & \cdots & p_{n_2(s+r-1)} \end{bmatrix},$$

$$Q = \begin{bmatrix} q_{11} & 0 & \cdots & 0 \\ q_{21} & q_{22} & \cdots & 0 \\ & \vdots & & \\ q_{m1} & q_{m2} & \cdots & q_{mm} \\ q_{n_2 1} & q_{n_2 2} & \cdots & q_{n_2 m} \end{bmatrix},$$

$$R = [r_1 \cdots r_{n_2}]^T.$$

The general aim of the MPC scheme is that the future outputs on the considered horizon should follow a pre-determined reference trajectory and, at the same time, the necessary control effort should be minimized. A typical cost function includes increments of the control signal, the control signal itself or neither of them. Accordingly, it can be defined as follows [24]:

$$J_{pr} = \sum_{i=1}^{n_2} [y_p(t+i) - y_d(t+i)]^2, \tag{40}$$

where $y_d(t)$ is the reference trajectory used over the prediction horizon. The optimal controller output is found by minimizing the above cost function such that.

$$\frac{\partial J_{pr}}{\partial u} = 0. \tag{41}$$

Differentiating the cost function defined in (40), leads to the following optimal solution:

$$U(t) = [Q^T Q]^{-1} Q^T [W(t) - PX(t) - \mathrm{R}err(t)]. \tag{42}$$

The above scheme can be extended to MIMO systems. Consider a MIMO system with two inputs and two outputs is described by the folowing nonlinear discrete time differnce equation.

$$\vec{y}(t+1) = F[\vec{y}(t), \vec{y}(t-1), \ldots, \vec{y}(t-s), u_1(t), u_1(t-1), \ldots, u_1(t-m1), u_2(t), u_2(t-1), \ldots, u_2(t-m2]^T, \tag{43}$$

Where $\vec{y}(t) = [y_1(t)\quad y_2(t)]^T$, the index s indicates the previus values of $\vec{y}$, and $m1$, and $m2$ are the number of the previous values of u_1's and u_2's respectively. This system can be modeled by the proposed ARX-LM network as follows:

$$\bar{y}_p(t+1) = A\bar{y}(t) + B\bar{u}(t), \tag{44}$$

where

$$\vec{y}_p(t+1) = \begin{bmatrix} y_{p1}(t+1) \\ y_{p2}(t+1) \end{bmatrix}$$

$$A = \begin{bmatrix} a_{11} & a_{12}\ldots a_{1s} & a_{1(s+1)}\ldots a_{1(2s)} \\ a_{21} & a_{22}\ldots a_{2s} & a_{2(s+1)}\ldots a_{2(2s)} \end{bmatrix},$$

$$\vec{y}(t) = \begin{bmatrix} y_1(t) & y_1(t-1) & \ldots & y_1(t-s) & y_2(t) & \ldots & y_2(t-s) \end{bmatrix}^T$$

$$B = \begin{bmatrix} b_{11} & b_{12} & \ldots & b_{1m1} & b_{1(m1+1)} & \cdots & b_{1(m1+m2)} \\ b_{21} & b_{22} & \ldots & b_{2m1} & b_{2(m1+1)} & \cdots & b_{2(m1+m2)} \end{bmatrix},$$

$$\vec{u}(t) = \begin{bmatrix} u_1(t) & u_1(t-1) & \ldots & u_1(t-m1) & u_2(t) & \ldots & u_2(t-m2) \end{bmatrix}^T$$

The model prediction over a costing horizon of n_2 time steps is given by:

$$\vec{y}_p(t+1) = A\vec{y}\ (t) + B\vec{u}(t) + \vec{err}(t)$$

$$\vdots$$

$$\vec{y}_p(t+i) = A\vec{y}\ (t+i-1) + B\vec{u}(t+i) + \vec{err}(t)$$

$$\vdots$$

$$\vec{y}_p(t+n_2) = A\vec{y}\ (t+n_2-1) + B\vec{u}(t+n_2-1) + \vec{err}(t), \quad (45)$$

where $err(t)$ is the vector of the estimation of the modeling errors. As in the case of SISO system, the above equations can be transformed into the following forms using the back-substitutions:

$$Y_1(t) = P_1\ X(t) + Q_1\ U(t) + R_1\ err_1(t)$$
$$Y_2(t) = P_2\ X(t) + Q_2\ U(t) + R_2\ err_2(t), \quad (46)$$

where $Y_1(t) = [y_{p1}(t+1)\ .\ .\ .\ y_{p1}(t+n_2)]^T$, and $Y_2(t) = [y_{p2}(t+1)\ .\ .\ .\ y_{p2}(t+n_2)]^T$ denote the predicted values of both the first and the second output of the model over the prediction horizon, is a vector of the past plant and controller outputs,

$$X(t) = \begin{bmatrix} y_1(t) & y_1(t-1) & \ldots & y_1(t-s) & y_2(t) & \ldots & y_2(t-s). & . & . \\ u_1(t) & u_1(t-1), & \ldots & u_1(t-m1) & u_2(t) & \ldots & u_2(t-m2) \end{bmatrix}^T$$

$U(t) = [\ u_1(\ t\ ), . . ., u_1(\ t + m1 - 1),$

$u_2\ (t), . . ., u_2\ (t + m2 - 1)]^T$, is a vector of future outputs of the controller. The matrices P^i, Q^i, and R^i are defined below.

$$P^i = \begin{bmatrix} p^i{}_{11} & \cdots & p^i{}_{1(2s+m1+m2-1)} \\ \vdots & & \vdots \\ p^i{}_{n_2 1} & \cdots & p^i{}_{n_2(2s+m1+m2-1)} \end{bmatrix}, \ i=1,2$$

$$Q^i = \begin{bmatrix} q^i{}_{11} & 0 & \cdots & 0 \\ q^i{}_{21} & q^i{}_{22} & \cdots & 0 \\ \vdots & & & \\ q^i{}_{m1} & q^i{}_{m2} & \cdots & q^i{}_{mm} \\ q^i{}_{n1} & q^i{}_{n2} & \cdots & q^i{}_{n_2(2m)} \end{bmatrix},$$

$$R^i = [r^i{}_1 \cdots r^i{}_{n_2}]^T.$$

In order to show how the process outputs tracks the desired responses, the cost function defined in (40) is modified as follows:

$$J_m = \sum_{j=1}^{2}\sum_{i=1}^{n_2}(y_{pj}(t+i) - y_{d\,j}(t+i))^2$$
$$= (P_1X(t) + Q_1U(t) + R_1 err1(t) - D_1(t))^T$$
$$(P_1X(t) + Q_1U(t) + R_1 err1(t) - D_1(t)) +$$
$$(P_2X(t) + Q_2U(t) + R_2 err2(t) - D_2(t))^T$$
$$(P_2X(t) + Q_2U(t) + R_2 err2(t) - D_2(t)), \quad (47)$$

where $y_{dj}(t)$ is a vector of the desired responses of the j^{th} output over the prediction horizon. Differentiating the cost function defined in (47) leads to the following optimal solution:

$$U(t) = [Q1^T Q_1 + Q2^T Q_2]^{-1} (Q1^T [D_1(t) - P_1 X(t) - R_1 err1(t)] + Q2^T [D_2(t) - P_2 X(t) - R_2 err2(t)]) . \quad (48)$$

3.2. Simulation results

The long range predictive control scheme based on the ARX-LM network is evaluated using two cases study, one is SISO and the other is MIMO. These cases have time varying parameters, dead time, and non-minimum phase behavior. The first case is the mean arterial blood pressure control system using a typical vasoactive drug. The second is the regulation of the mean arterial blood pressure and the cardiac output using two typical drugs, one is a vasoactive drug and the second is an inotropic drug.

3.2.1. SISO Case

The model of the Mean Arterial Blood Pressure (MABP) of a patient under the influence of sodium nitroprusside (SNP) described in Slate and Sheppard [31] is:

$$P(t) = P_0 + \Delta P(t) + V(t) + P_c(t) , \quad (49)$$

where $P(t)$ is the mean arterial blood pressure (mmHg), P_o is the initial blood pressure, $\Delta P(t)$ is the change in pressure due to infusion of SNP, $P_c(t)$ is the change of pressure due to renin reflex action which is the body's reaction to the use of a vasodilator drug, $V(t)$ is a stochastic noise which associated with disease, treatment drugs or liquids administered, pain , and recovery from anesthesia. The transfer function that describes the relation between the change in the blood pressure and drug infusion rate U, is:

$$\Delta P(s) = -\frac{K\, e^{-T_i S} (1 + \alpha\, e^{-T_C S})}{(1 + T_1 s)} U(s), \quad (50)$$

where U is the drug infusion rate (ml/h), K is the drug sensitivity, α is the recirculation constant, T_i is the initial transport delay time (second), T_c is the recirculation delay time (second), T_1 is the response time constant (second). Disregarding the effect of renin reflex, the corresponding ARMA model of MABP under the influence of SNP is:

$$P(t) = P_o - \frac{z^{-d}\, (b_o + b_m\, z^{-m})}{1 - a_1\, z^{-1}} U(t) + P_n ;$$
$$P_n = \frac{C(z^{-1})}{1 - a_1 z^{-1}} \zeta(t) \;, \quad (51)$$

where $\xi(t)$ is a broadband random sequence, and $C(z^{-1})$ is a stable polynomial. The nominal values of the parameters in model (51) are (a_1=0.741, b_o=0.187, b_m =0.075, d =3, m=3) at sampling period T= 15 sec [32]. The parameter values of different patients including the system time delay are drastically changed. Moreover a patient's characteristics usually vary during the course of operation. The time constant and gains of the system change in an exponential manner. The time constant and gains of the system change in an exponential manner. The variation of parameters is modeled as follows [33]:

$$\text{par}(k)=\text{par}(0)+(\text{par}(\infty)-\text{par}(0))(1-e^{-k/\tau}) . \quad (52)$$

The control objective was to decrease the blood pressure from an initial large value (e.g., P_o=150 mmHg) to a desired level (e.g., P_d =100 mmHg) within 5 ~ 20 min, and to maintain this level within ± 15 mmHg.

The performance of the controller scheme was tested, using four cases summarized as follows: The first case: The plant parameters are set to the nominal values (a_1=0.741, b_o=0.187, b_m=0.075, d=3, and m=3). The second case: The patient has variable gain "sensitivity" such that the value of b_o is set to 0.187 and then varies exponentially with time as defined in (52) and the other parameters are kept constants at the nominal values (b_0=0.187, b_m=0.075, d=3, and m=3). The third case: The patient has variable time constant such that the value of (a_1) is initially set to 0.741 and then varies exponentially with time as defined in (52), and the other parameters are kept constants at the nominal values. The fourth case: The patient with variable delay time such that the delay time is abruptly increased from d=3, m=3 to d=5, m=5 at 60 minute mark, and the other parameters are set to nominal values.

Figs. 5 and 6 show the response of the blood pressure system to the control signal of the proposed scheme when the second and the third case are presented. The steady state error, overshoots, and undershoots of the blood pressure shown in the figures are acceptable compared with the results shown in [32, 33]. Compared the ARX-LM network-based long range predictive control with the hybrid wavelet fuzzy neural network (HWFNN)- based predictive control scheme [14], in the sense of the Root Mean Squares (RMS) error defined in (53) , the former is superior to the latter as depicted in table 3.

$$RMS = \sqrt{\frac{1}{N} \sum_{t=1}^{N} (y_d(t) - y(t))^2} , \quad (53)$$

where N is the number of samples, y_d is the desired output of the system and y is its actual output.

3.2.2. MIMO case

Clinically, it is required to regulate simultaneously the cardiac output (CO) and the mean arterial pressure (MAP) of a patient in hospital intensive care using various drugs. Two typical drugs used are dopamine (DOP), which is an inotropic drug, and sodium nitroprusside (SNP), which is a vasoactive drug. For the purpose of the simulation study Linkens and Nie adopt the same model used in [34, 35] which is given by [36]:

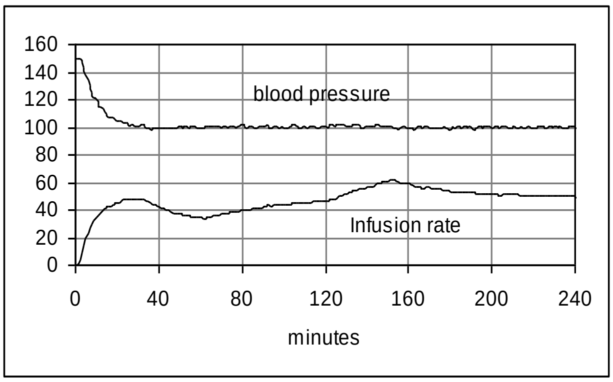

Fig. 5. The mean arterial blood pressure response of the ARX-LM- based long range predictive control (case two).

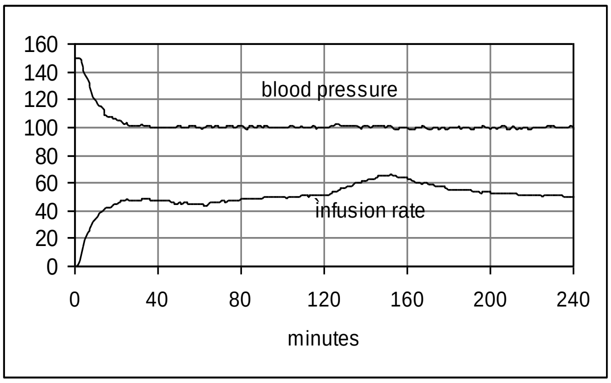

Fig. 6. The mean arterial blood pressure response of the ARX-LM-based long rang predictive control (case three).

Table 3
Comparison between the proposed ARX-LM- based and the HWFNN-based [14] long range predictive control schemes

System cases	The RMS error values	
	The proposed ARX-LM-based long range predictive control	The HWFNN-based long range predictive control
Case one	1.411	1.935
Case two	1.496	2.350
Case three	1.460	2.302
Case four	1.653	1.935

$$\begin{bmatrix} \Delta CO \\ \Delta MAP \end{bmatrix} = \begin{bmatrix} 1.0 & -24.76 \\ 0.6636 & 76.38 \end{bmatrix} \begin{bmatrix} \frac{K_{11}{}^{-\tau_1 s}}{sT_1 + 1} & \frac{K_{12}{}^{-\tau_2 s}}{sT_1 + 1} \\ \frac{K_{21}{}^{-\tau_2 s}}{sT_2 + 1} & \frac{K_{22}{}^{-\tau_2 s}}{sT_2 + 1} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} , \quad (54)$$

where ΔCO (ml/s) is the change in cardiac output to U_1 and U_2 ; ΔMAP (mmHg) is the change in mean arterial pressure due to $U1$ and $U2$; U_1 ($\mu g / Kg / \min$) is the infusion rate of dopamine; U_2 (ml/h) is the infusion rate of sodium nitroprusside ; K_{11}, K_{12}, K_{21}, and K_{22} are steady state-gains with nominal values of 8.44, 5.275, -0.09 and -0.15 respectively ; τ_1 and τ_2 represent two time delays with nominal values of τ_1= 60 s and τ_2= 30 s; and T_1 and T_2 are time constants with nominal values of 84.1 s and 58.75 s respectively. It is evident that the model is characterized by strong interaction between the variables and large time delays in control. Through this work, the following values are used. The sampling time is 30 s. Set- points for *CO* and *MAP* was set to be 20 ml/s and -10 mmHg changing from nominal values of 100ml/s and 120 mmHg respectively [37]. This paper employs this MIMO system to test the soundness of the proposed ARX-LM-based long range predictive control scheme. To investigate the effectiveness of the proposed scheme, four cases of the system parameters are performed; these cases are listed as follows [37]: The first case: The plant parameters (K_{11}, K_{12}, K_{21}, K_{22} ,τ_1, τ_2, T_1 and T_2) are set to the nominal values. The second case: The plant parameters (K_{21}, K_{22}, τ_1, τ_2, T_1 and T_2) are set to the nominal values and the two parameters K_{11}, K_{12} were abrupt changed by 10% from their nominal values. The third case: The plant parameters (K_{11}, K_{12}, τ_1, τ_2, T_1 and T_2) are set to the nominal values and the two parameters K_{21}, K_{22} were abrupt changed by 10% from their nominal values.

The fourth case: The plant parameters (K_{11}, K_{12}, K_{21}, K_{22}, τ_1 and τ_2) are set to the nominal values and the two parameters T_1 and T_2 were abrupt changed by 10% from their nominal values. Fig. 7 shows the response of the process using the proposed predictive scheme when the first case of the system parameters is presented. These dynamics are acceptable according to the results published in [37]. The RMS errors values defined in (55) of the proposed ARX-LM –based predictive control scheme are computed and shown in table 4.

$$RMS1 = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(\Delta CO_d - \Delta CO))^2}$$

$$RMS2 = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(\Delta MAP_d - \Delta MAP))^2} \ . \qquad (55)$$

Fig. 7. The changes for both cardiac output and blood pressure using ARX-LM- based long range predictive control (case one).

Table 4
The RMS errors obtained using the proposed ARX-LM-based predictive control scheme

System case	The RMS error-1	The RMS error-2
Case one	0.34827	0.23922
Case two	0.57250	0.60371
Case three	0.43903	0.89736
Case four	0.34825	0.24143

where N is the number of samples, ΔCO_d, ΔMAP_d are the desired changes of ΔCO and ΔMAP respectively. From the time responses and the performance indexes, the developed scheme has the ability to regulate the multi-variable with ease.

4. Conclusions

This paper introduces a new local model network named ARX-LM network. This network mergs the locality feature of the TSK fuzzy model and the wavelets that are very useful for function appoximation.

It basically adopts the philosophy of forming a process be modeled with a set of fuzzy-wavelet submodels. Each sub-model comprises a TSK fuzzy model fertilized by a wavelet function. The output of these sub-models are weighted and summed to produce the final output. Developing of the proposed network, requires two phases structure and recall. The former comprises structure learning using the fuzzy ART algorithm and parametrs learning using the RLS algorithm. The latter used unseened data to test the identifiability, generaly, and plastisty featuery of the network. The main two notable points of the proposed network are that its structure is an ARX

form and its computational demand is relatively small. The former means that the conventional linear and nonlinear control systems e.g. laypunove's method can be applied with ease. The latter is enherits from using few TSK and fuzzy rules and wavelets that makes the proposed ARX-LM network is promising in modeling and controlling real time systems. Modeling simulationd results show that best results have been acheived using the proposed ARX-LM network compered with the WNN and the FWN. The proposed ARX-LM network was employed to develop a long range predictive control scheme for SISO and MIMO systems. The proposed ARX-LM-based predictive control scheme was applied to control the MABP with ease. Results show that better results have been achived using the proposed ARX-LM-based long range predictive control scheme compared with HWENN-based long range predictive control.

References

[1] C.T. Lin. and C.S.G. Lee., "Neural-Networks-Based Fuzzy Logic Control and Decision System," IEEE Trans. on Computers, Vol. 40, pp. 1320-1336 (1991).

[2] T. Takagi and M. Sugeno, "Fuzzy Identification of Systems and Its Applications to Modeling and Control," IEEE Trans. Syst. Man. Cybern., Vol. SMC-15, pp. 116-132 (1985).

[3] C.F. Juang and C.T. Lin, "An Online Self-Constructing Neural Fuzzy Inference Network and Its Applications," IEEE. Trans. Fuzzy. Sys., Vol. 6, pp. 12-32 (1998).

[4] C.H. Lee. and C.C. Teng., "Identification and Control of Dynamic Systems Using Recurrent Fuzzy Neural Networks," IEEE. Trans. Fuzzy. Sys., Vol. 8 (4) (2000).

[5] Q. Zhang and A. Benveniste, "Wavelet Networks," IEEE Trans. Neural Network, Vol. 3, pp. 889-898 (1992).

[6] K. Kunikazu and T. Toyoshi, "A Wavelet Neural Network for Function Approximation and Network Optimization," Proceeding of ANNIE' 94, Vol. 4 (1994).

[7] Q. Zhang, "Using Wavelet Networks in Nonparametric Estimation," IEEE Trans. Neural Network, Vol. 8, pp. 227-236 (1997).

[8] P. Seda, and B. Yasar, "Nonlinear System Modeling by Using Wavelet Networks," JICI Proceeding of International Conference on Signal Processing, ISSN, Vol. 1 (2) (2003).

[9] G. Lekutai, Adaptive Self-Tuning Neuro Wavelet Network. Ph.D. Thesis, Blackshurg, Virginia (1997).

[10] Y. Min, B.S. Chen, and F.Y. Shiau, "Adaptive Wavelet Network Control Design for Nonlinear Systems," Proceeding of the 35Th Conference on Decision and Control, Kobe, Japan (1996).
[11] C.K. Lin, "Adaptive Tracking Controller Design for Robotic Systems Using Gaussian Wavelet Networks," IEE Proceeding of Control Theory Application, Vol. 149 (4) (2002).
[12] W. Daniel, C. Ho., Ping-An Zhang, and Xu. Jinhua, "Fuzzy Wavelet Networks for Function Learning," IEEE. Trans. Fuzzy Syst., Vol. 9 (1) (2001).
[13] J.S.R. Jang, "ANFIS: Adaptive-Network-Based Fuzzy Inference System," IEEE Trans. Sys. Man. Cybern., Vol. 23 (3), pp. 665-684 (1993).
[14] H.A. Awad, N.M. El-Rabaie and T.A. Mahmoud, "Wavelet Fuzzy Neural Network -Based Predictive Control System," IEEE Melecon 2004, May 12-15, Dubrovnik, Croatia (2004).
[15] N.M. El-Rabaie, H.A. Awad, and T.A. Mahmoud, "Wavelet Fuzzy Neural Network -Based Internal Model Control System," The 5th IEE International Conference on Computational Aspects and Their Applications in Electrical Eng., CATAEE'04, Philadelphia Unv., Amman, Jordon (2004).
[16] G.A. Carpenter, S. Grossberg, and D.B. Rosen, "Fuzzy ART: Fast Stable Learning and Categorization of Analog Patterns by an Adaptive Resonance Theory," Neural Networks, Vol. 4, pp. 759-771 (1991).
[17] P.M. Larsen, "Industrial Applications of Fuzzy Logic Control," Int. J. Man Mach. Studies, Vol. 22, pp. 3-10 (1980).
[18] K. Tanaka, "Stability and Stabilization of Fuzzy-Neural-Linear Control Systems," IEEE. Trans. Fuzzy Syst, Vol. 3 (4) (1995).
[19] B. Kosko, "Global Stability of Generalized Adaptive Fuzzy Systems," IEEE Trans. Sys. Man. Cybern. , Part C, Vol. 28 (3) (1998).
[20] Y.K. Daio and M. Passino, "Adaptive Neural / Fuzzy Control for Interpolated Nonlinear Systems" IEEE. Trans. Fuzzy Syst., Vol. 10 (5) (2002).
[21] D.W. Clarke, C. Mohtadi and P.S. Tuffs, "Generalized Predictive Control- Part I: The Basic Algorithm," Automatica, Vol. 23, pp. 137-148 (1987).
[22] H.A. Awad Fuzzy Neural Networks For Modeling and Controlling Dynamic Systems. Ph.D. Thesis, Intelligent systems Int. School of Engineering, Cardiff University, Cardiff, U.K. (2001).
[23] J. Valent de Olivveira, and J.M. Lemos, "Long-Range Predictive Adaptive Fuzzy Relational Control," Fuzzy Sets and Systems, Vol. 70, pp 337-357 (1995).
[24] D.A. Linkens, and S. Kandiah, "Long-Range Predictive Control Using Fuzzy Process Models," Trans. Icheme, Vol. 74 (1996).
[25] M. Mahfouf, M.F. Abbod and J.D. Linkens, "Online Elicitation of Mamdani-Type Fuzzy Rules Via Tsk-Based Generalized Predictive Control," IEEE Trans. Syst., Man. Cybern., Vol. 33 (3) (2003).
[26] M. Lee and S. Park, "A New Scheme Combining Neural Feedforward Control with Model-Predictive Control," AIChE Journal, Vol. 38 (2), (1992).
[27] Y.S. Choi, A.D. Whittaker, and D.C. Bullock, "Predictive Neuro-Fuzzy Controller for Multivariable Process Control," American Society of Agriculture Eng., Vol. 39 (1996).
[28] A.G. Parlo, S. Parthasarathy and A.F. Atiya, "Neuro-Predictive Process Control Using on-Line Controller Adaptation," IEEE. Trans. on Cont. Syst. Tech. Vol. 9 (5) (2001).
[29] N.M. El-Rabaie, "Improvements in the Mean Arterial Blood Pressure Using Fuzzy-Neural Model-Based Predictive Control," IEEE 7th International Conference on Intelligent Engineering Systems, Assuit, Luxor, Egypt (2003).
[30] X. Xia, D. Huang and Y. Jin, "Nonlinear Adaptive Predictive Control Based on Orthogonal Wavelet Networks," Proceeding of the 4th World Congress in Intelligent Control and Automation, Shanghai, Chine, pp. 10-14 (2002).
[31] J.B. Slate, and L.C. Sheppard, "Automatic Control of Blood Pressure by

Using Drug Infusion," IEE. Proceeding, Vol. 129, Part a (9) (1982).

[32] C.T. Chen, W.L. Lin, T.S. Kuo and C.Y. Wang, "Adaptive Control of Arterial Blood Pressure with Learning Controller Based on Multilayer Neural Networks," IEEE Trans. on Biomed. Eng., Vol. 44 (7) (1997).

[33] G.A. Pajunen, M. Steinmetz and R. Shankar, "Model Reference Adaptive Control with Constraints for Postoperative Blood Pressure Management," IEEE Trans. Biomed. Eng. Vol. 37, pp. 679-686 (1990).

[34] D.A. Linkens, and J. Nie, "A Unified Real Time Approximate Reasoning Approach for Use in Intelligent Control Application to Multivariable Blood Pressure Control," Int. J. Control, Vol. 56, pp. 334-363 (1992a).

[35] D.A. Linkens and J. Nie, "A Unified Real Time Approximate Reasoning Approach for Use in Intelligent Control Application to Multivariable Blood Pressure Control," Int. J. Control, Vol. 56, pp. 365-398 (1992b).

[36] D.A. Linkens and J. Nie, "FCMAC: A Fuzzified Cerebellar Model Articulation Controller with Self-Organizing Capacity," Automatica Vol. 30 (4), pp. 655-664 (1994).

[37] D.A. Linkens and J. Nie, Fuzzy-Neural Control: Principles, Algorithms, and Applications. Prentice Hall International (UK) limited (1995).

Received September 15, 2005
Accepted December 31, 2005