

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSI*net*

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

A Novel Network Control Algorithm for Multi-Axis Manipulator Systems

¹Kuei-Shu Hsu, ¹Jung-Hui Yu and ²Tsung-Han Li

¹Department of Applied Geoinformatics,

Chia Nan University of Pharmacy and Science, Taiwan 717, Republic of China

²Department of Mechanical Engineering, Tatung University, Taiwan 717, Republic of China

Abstract: Many researchers have studied robot system identification in recent years. Nonlinear system identification, whether, it be forward or inverse model identification, is becoming increasingly important in modern control applications. Nonlinear system identification can improve control performance significantly, especially when the system dynamic behaviors are unknown and exhibit great nonlinearity. Additionally, the network has been applied to control engineering. The concept of network simulates the concentration of a set of antibodies. The network system has the following features: self-organization, memory, recognition, adaptability and the ability of learning. Therefore, the network could be applied to nonlinear system identification and provide various feasible options for system models with robust and adaptive characteristics.

Key words: Network algorithm, identification, manipulator system

INTRODUCTION

Nonlinear system identification, whether it be forward or inverse model identification, has been suitable to establish the model of unknown systems. Therefore, nonlinear system identification is becoming increasingly important in modern control applications (Babuska *et al.*, 1998; Liu and Kadirkamanathan, 1999; Yu and Li, 2001; Venkatesh and Dahleh, 2001). Nonlinear system identification can improve control performance significantly, especially when the system behaviors are unknown, complex and exhibit great nonlinearity. In general, the system models can be classified into two groups: the physical model is using the physical property to derive the model of the system, while the input-output model is using the relationship between input and output data of the system in order to establish the model. The first method to establish the model usually depends on complex mathematical equations to derive the model of system and the model has many limits to follow which then affect the performance of the model. The second method to establish the model is using the relationship between the input and output data of the system. This is the great advantage of the system identification.

System identification is used to establish the mathematical model of unknown systems. If the input and output data of an unknown system can fulfill the model we established, then we call it an approximate model of the unknown system. Nonlinear system identification consists of model structure selection and parameter estimation, which are both very important. For multi-variable (or MIMO) systems, the nonlinear structure

could be treated as a link from acquired input and output data to a regression vector from the unknown system. Then one must establish the mathematical model of system dynamics and also use few parameters in order to estimate the behavior of the model of the nonlinear system.

For nonlinear system identification, the NARX model has been implemented in several types of configurations such as genetic programming (Enab, 1995; Petridis *et al.*, 1998; Hwang *et al.*, 2000), neural network (Li *et al.*, 2000; Akhmetov *et al.*, 2001), fuzzy modeling (Sousa and Setnes, 2000; Wang *et al.*, 2000) and polynomial representation (Billings *et al.*, 1989). On the other hand, biological information processing systems, such as the network system, have been increasingly applied to engineering fields. Recent studies on the network system have clarified through simulation the interactions between various components of the network system, such as the stimulation and suppression chains between various cells or the overall behaviors of the system based on an immunology point of view and rapidly responding to the presence of foreign material while quickly stabilizing the network system. In short, the network system has the following features: self-organization, memory, recognition, adaptation and the ability to learn. Therefore, network systems could be applied to nonlinear system identification and provide various feasible ideas for system models with robust and adaptive characteristics. The purpose of this study is to use the network identified nonlinear system's NARX model. A two-link rigid manipulator simulation is used so as to evaluate the proposed methodology.

In recent years, many studies of nonlinear system model establishment have been proposed and have become an essential branch of control theory. Common model building methods are using physical properties, neural networks, or the relationship between the input and output data of the system in order to establish the model of system (in addition to other methodologies). However, all the model types have their advantages and disadvantages for deriving a system model with robust and adaptive characteristics. The network system has the following features: self-organization, memory, recognition, adaptability and ability to learn. Therefore, the network system could be applied to nonlinear system identification and provide various feasible ideas for system models with robust and adaptive characteristics.

SYSTEM IDENTIFICATION

In recent years, there have been many studies proposing the methods for establishing the model of an unknown system. They can be classified into two groups: those using physical properties to derive the model of the system and others using the relationship between the input and output data of the system in order to establish the model.

In general, the first method to establish the model usually depends on complex mathematical equations to derive the model of system, so the model always has some limits to follow which then affect the performance of the model. Due to its derivation from physical properties, the model does not include the dynamics of the system to a large degree. Regarding this kind of method for establishing the model, one must consider another method for building system models.

Conversely, the second method for establishing the model is using the relationship between the input and output data of the system. Due to the practical use of the input and output data of the system, the dynamic behavior of the system is evident to a greater degree than with the first method. Furthermore, for different systems this kind of the method can still establish the model of unknown systems without undergoing any changes. However, the first kind of method operates according to the physical properties of the unknown system so as to derive the model again. To compare these two methods, one can surmise that the second method is more generally applicable to a variety of systems than the first method.

NARX model: The system identification uses the relationship between the actual input and output data of the unknown system in order to establish the behavior of the model. In this study, we used the NARX (The

Nonlinear Auto Regressive Model with eXogenous Inputs) model as our main object. Based on this model, we will establish the model of the unknown system and introduce its theory as follows.

The NARX model proposed by Leontaritis and Billings (1985) functions to identify the model of the system. In using this model, one can predict the behavior of an unknown system, assessing the input and output data of the unknown system to establish the dynamics system of the mathematical model. It can be used to present a nonlinear system with a small number of parameters. However, the main objective for the NARX model is to determine the structures an of unknown system.

Consider a general NARX model taking the following form:

For r is number of input and m is number of output, the equation expands as seen in Eq. 1:

$$y(t) = f[y(t-1), \dots, y(t-n_y), \dots, u(t-d), \dots, u(t-n_u)] + e(t) \quad (1)$$

Where:

$$y(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_m(t) \end{bmatrix}, u(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_r(t) \end{bmatrix}, e(t) = \begin{bmatrix} e_1(t) \\ e_2(t) \\ \vdots \\ e_r(t) \end{bmatrix} \quad (2)$$

where, $y(t)$, $u(t)$ and $e(t)$ are the vectors of system output, input and prediction error respectively, d is the time delay and n_u , n_y are the maximum lags in the inputs, outputs, $f(\cdot)$ is some vector-valued nonlinear function. Equation 1 can be presented as a polynomial expansion shown as Eq. 3, θ_k is an unknown parameter and $p_k(t)$ is the monomial of degree 0 to 1 and consists of delayed outputs and inputs:

$$y(t) = \sum_{k=1}^m \theta_k p_k(t) + e(t) \quad (3)$$

$$m = \sum_{i=1}^l n_i + 1; n_i = (n_{i-1}(n_y + n_u + i - 1)) / i, n_0 = 1 \quad (4)$$

For example, if the maximum time lag of y and u is 10 (i.e., $n_u = n_y = 10$) and the polynomial degree l is 2, then the number of candidate (possible) terms, m is 231. Although any system can be estimated on Eq. 3, this might result in an excessively complex model and ill conditions. Therefore, the determination of which terms to be included in the model from the large number of candidate terms is essential. In many real systems, it has been shown that models with 10 terms are usually sufficient to capture the dynamics of highly nonlinear processes. After

building the structures of the NARX model, it is necessary to find the coefficient for each term. With different terms, there are different coefficients to arrange in pairs. Next, we will introduce the way of linear regression in order to find the coefficient for each term in the NARX model. One must use linear regression to find the coefficients of each term of the NARX model. Here, are the methods of linear regression:

$$Y = BX \tag{5}$$

$$B = [b_1 \ b_2 \ \dots \ b_k], \ b_i \text{ is the coefficient of each term, } i = 1, 2, \dots, k \tag{6}$$

$$X = [X_1 \ X_2 \ \dots \ X_k]^T, \ X_i \text{ is the vector of estimators, } i = 1, 2, \dots, k \tag{7}$$

Where:

$$X_1 = \begin{bmatrix} x_{11} \\ x_{12} \\ \vdots \\ x_{1n} \end{bmatrix}, X_2 = \begin{bmatrix} x_{21} \\ x_{22} \\ \vdots \\ x_{2n} \end{bmatrix}, \dots, X_k = \begin{bmatrix} x_{k1} \\ x_{k2} \\ \vdots \\ x_{kn} \end{bmatrix}, \ n = 1, 2, \dots, N \tag{8}$$

By some simple mathematical operations one can get:

$$X^T X \hat{B} = X^T Y \tag{9}$$

if $(X^T X)^{-1}$ exists, then:

$$\hat{B} = (X^T X)^{-1} X^T Y \tag{10}$$

where, \hat{B} is the coefficient vector and it is found by using linear regression. This vector is the unknown parameter θ_k as mentioned before. Subsequently, bring θ_k to Eq. 3 and then we get the scheme of the NARX model. It is also called the forward NARX model.

However, there are some details which should be considered, especially the stability of the NARX model. In general, it is assumed that a system could be modeled as the following nonlinear discrete time difference equation:

$$\hat{y}(t) = f(y(t-1), \dots, y(t-n_y), u_1(t-d_1), \dots, u_1(t-n_{u1}), u_2(t-d_2), \dots, u_2(t-n_{u2}), \dots, u_r(t-d_r), \dots, u_r(t-n_{ur})) \tag{11}$$

In Eq. 11, the model output is a function of the past values of the input and output of the system. In contrast with the model described above, the output of the identification model (meaning, rather than the plant) could be fed back into the identification model. Thus, we could change Eq. 11-12 and the model for this has the form:

$$\hat{y}(t) = f(\hat{y}(t-1), \dots, \hat{y}(t-n_y), u_1(t-d_1), \dots, u_1(t-n_{u1}), u_2(t-d_2), \dots, u_2(t-n_{u2}), \dots, u_r(t-d_r), \dots, u_r(t-n_{ur})) \tag{12}$$

Therefore, the identified model was required to check its stability before it could be utilized for control purposes. Instead of minimizing the least-squares error between the outputs of the plant and the series-parallel model, the parallel model is used to calculate the error. Furthermore, the coefficients of the NARX model are determined using the series-parallel model. Subsequently, the residual between the plant and model output is calculated for the case of the inverse model. Both residuals' estimated input follow the parallel-type NARX model. The researchers have observed very good performance in simulation and experiment. This study has also used the same scheme of the parallel-type NARX model for using the network to identify the system model.

NETWORK SYSTEM

The network control system protects living bodies from the invasion of foreign substances such as viruses, bacteria and other parasites (called antigens) which enter the bloodstream. The main task of the system is to detect the antigens using lymphocyte cells and to produce a network response to eliminate antigens, such as has enabled the human species to survive. The basic components of the biological network system are macrophages and lymphocytes. The lymphocytes play an important role in all network responses. The main cells of the network response are introduced as follow:

Phagocytes: Phagocytes are an important type of macrophage. They have surface receptors to detect and destroy invaders (antigens, viruses and bacteria) to the human body. The phagocytes use primitive non-specific recognition systems to bound and destroy invaders. This is called 'non-specific network response' as the first line of defense.

Network responses: The network system possesses two types of network response: innate (or non-adaptive) responses and adaptive network responses. When an infectious foreign pathogen attacks the human body, the innate network system is activated as the first line of defense. Innate immunity is not directed in any way towards specific invaders, but rather against any pathogens that enter the body. It is called non-adaptive network response. The adaptive network response occurs when, the network system encounters the antigen for the

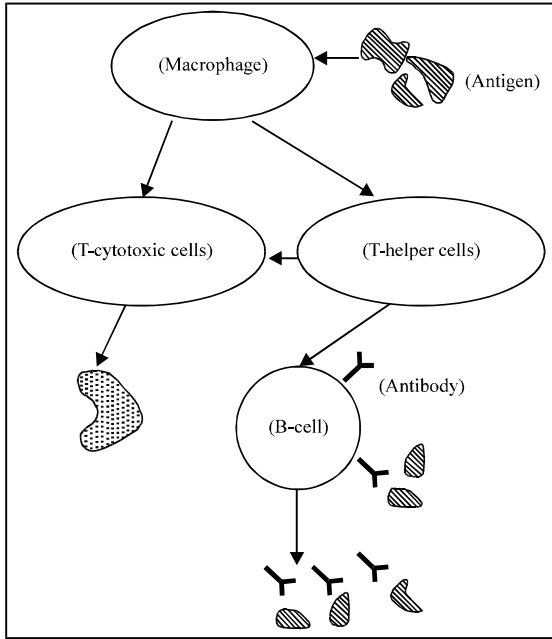


Fig. 1: Illustration of the biological network system

first time and reacts against it. For example, it may produce antibodies by producing lymphocytes which combine with the antigen to cause its elimination. The adaptive network response learns about the specificity of antigens, thus preparing the body for any further invasion from that same antigen. This learning mechanism creates the network system memory. In addition, the adaptive network responses can be elicited from an antigen which is similar, although not identical, to the original one which established the memory. Hence, the network system possesses a content-addressable memory.

Figure 1 depicts the model describing the relationship between components in the network system. The macrophage has surface receptors to detect and destroy invaders to the human body. Subsequently, the macrophage becomes an Antigen Presenting Cell (APC) and releases interleukines, a kind of molecule called Major Histocompatibility Complex (MHC) to distinguish a self from other non-self. The APC transfer information about the antigen to T-cells, activating them. Then T-cells stimulate B-cells and antibodies are produced by B-cells to neutralize antigens. In addition, diversity in the network system is maintained because the least stimulated B-cells die daily and are replaced by an equal number of completely new B-cells generated by the bone marrow. These new B-cells are only added to the network if they possess an affinity to the cells already within, otherwise they die.

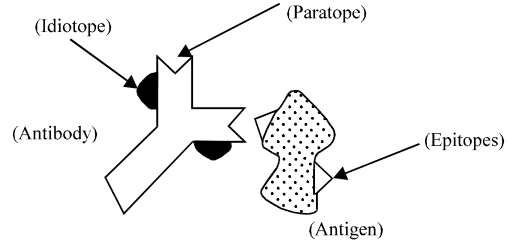


Fig. 2: Structure of an antigen and antibody

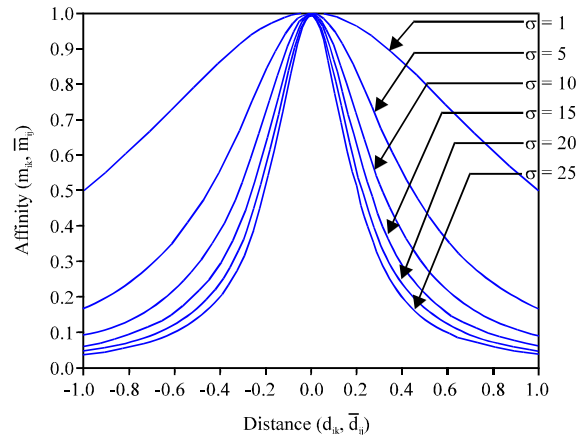


Fig. 3: The relation between affinity m_{ik} (\bar{m}_{ij}) and distance d_{ik} (\bar{d}_{ij})

Network: For the sake of convenience of the following explanation, we show the structure of an antigen and antibody in Fig. 2. In Fig. 2, the relationship between an antibody and its corresponding antigen is described. The portion of the antigen recognized by the antibody is called epitope (antigen determinant) and the one on the antibody that recognizes the corresponding antigen determinant is called paratope. Additionally, each type of antibody has also its specific antigen determinant called idiotope.

Based on the fact that antibodies are not just isolated, namely they are communicating to each other among different kinds of antibodies (Jerne, 1973) proposed a hypopaper: idiotypic network hypopaper is the concept that the network system is constructed as a large-scale closed system of lymphocytes paying close attention to mutual interaction between lymphocytes. This idea of Jerne's is schematically shown in Fig. 3. The idiotope Id1 of antibody 1 stimulates the B-lymphocyte 2, which attaches the antibody 2 to its surface, through the paratope P2. Viewed from the standpoint of the antibody 2, the idiotope Id1 of antibody 1 works simultaneously as an antigen. As a result, the B-lymphocyte 1 with the

antibodies 1 are suppressed by the antibody 2. On the other hand, the antibody 3 stimulates the antibody 1 since the idiotope Id3 of the antibody 3 works as an antigen for the antibody 1. These stimulation and suppression chains between antibodies form the large closed-chain loop which works as a self and non-self recognizer.

The theory of identifying system models using network:

The biological network system has the following features: self-organization, memory, recognition, adaptation and ability to learn. The main task of the network system is to detect the antigens by lymphocyte cells and produce the network response to eliminate antigens. In this study, a novel theory of the nonlinear system identification scheme based on the NARX network is proposed. This can be further explained as follows:

Consider that a MIMO system can be decomposed as a set of coupled MISO models for system identification. For a r-input-one-output NARX model the equation expands as Eq. 13:

$$y(t) = f(y(t-1), \dots, y(t-n_y), u_1(t-d_1), \dots, u_1(t-n_{u1}), u_2(t-d_2), \dots, u_2(t-n_{u2}), \dots, u_r(t-d_r), \dots, u_r(t-n_{ur})) + e(t) \quad (13)$$

Without loss of generality, it can be simplified as Eq. 14 assuming zero input time delay and identical maximum time lag n_{lag} :

$$y(t) = f(F) + e(t) \quad (14)$$

With:

$$F = \begin{bmatrix} y(t-1), \dots, y(t-n_{lag}), u_1(t), \dots, u_1(t-n_{lag}), \\ u_2(t), \dots, u_2(t-n_{lag}), u_r(t), \dots, u_r(t-n_{lag}) \end{bmatrix}$$

The regression vector Φ is considered as an epitope of the antigen (antigen determinant) and its behavior is the antigen response $y(t)$. Therefore, the total number of antigens will be $N-n_{lag}$ for the whole network, where, N is the number of data sets acquired. In addition, the antibodies in the proposed network are defined as below:

$$Ab_i = \{(\hat{y}(\bar{m}_{ij}), x_i)\} \quad i = 1, \dots, N_{Ab} \quad (15)$$

where, vector x_i is the receptor vector of the i th antibody similar to Φ , includes both paratope and idiotope and $\hat{y}(\bar{m}_{ij})$ is the corresponding network response of the i th antibody. N_{Ab} is the number of antibodies used in the network.

In the whole network, the stimulation and suppression chains between antibodies and the matching ratio between the antigen epitope and the antibody receptor are called affinity. In addition, if the paratope and epitope shapes are not quite complementary, then the two molecules may still bind, but with lower affinity. To calculate the affinity m_{ik} between i th antibody and k th antibody, the expression can be given as below:

$$m_{ik} = \frac{1}{1 + \sigma * d_{ik}^2} \quad (16)$$

where, $d_{ik} = \|x_i - x_k\|$ represents the distance between the receptor vectors x_i of i th antibody and x_k of k th antibody.

On the other hand, the following expression is applied to calculate the affinity \bar{m}_{ij} between j th antigen and i th antibody:

$$\bar{m}_{ij} = \frac{1}{1 + \sigma * \bar{d}_{ij}^2} \quad (17)$$

where, $\bar{d}_{ij} = \|x_i - \Phi_k\|$ represents the distance between the receptor vectors x_i of i th antibody and Φ_k of j th antigen.

Figure 3 shows the relation between affinity m_{ik} (\bar{m}_{ij}) and distance d_{ik} (\bar{d}_{ij}) associated with different values of σ . Clearly, the larger the value of σ , the wider the area covered by the curve.

For the whole network, the network response \hat{y}_i of the i th antibody to the j th antigen is defined as the following equation:

$$\hat{y}_i = \frac{1}{1 + \sigma_n * (\bar{m}_{ij} - \bar{m})^2} \quad (18)$$

With:

$$\sigma_n = \frac{1 - y_j}{y_j * (1 - \bar{m})^2} \quad (19)$$

where, \bar{m} is a constant and equal to -1.5 in this study. Under such a situation, the network response of the i th antibody to the j th antigen (i.e., \hat{y}_i) will be of equal value to real output y_j if their affinity \bar{m}_{ij} is one (which means the receptor of the antibody and epitope of the antigen are totally matched).

The dynamic equations of the idiotypic network proposed by Jerne (1973) is applied in this study to calculate the concentration of the i th antibody in the network below:

$$\frac{dA_i(t+1)}{dt} = \left(\sum_{k=1}^{N_{Ab}} m_{ik} a_k(t) - \sum_{k=1}^{N_{Ab}} m_{ki} a_i(t) + \bar{m}_{ij} - k_i \right) a_i(t) \quad (20)$$

$$a_i(t+1) = \frac{1}{1 + \exp(0.5 - A_i(t+1))} \quad (21)$$

where, $i, k = 0, 1, \dots, N_{Ab}$ is number of the antibody, A_i is stimulus of the antibody i , a_i is the concentration of the antibody i , m_{ik} is the affinity of the antibody i and antibody k , \bar{m}_{ij} is the affinity of antigen j and antibody i and k_i is the natural death coefficient.

Equation 20 is composed of four terms. The first term is the stimulation between i th antibody and k th antibody. The second term is suppressive interaction between antibodies. The third term is the stimulus from antigen. The final term is a natural extinction term which represents the dissipation tendency in absence of any interaction.

The stimulated and suppressive relation between antibodies can be integrated as the following affinity matrix M_{ik} :

$$M_{ik} = \begin{bmatrix} 1 & m_{12} & \dots & \dots & m_{1k} \\ m_{21} & 1 & \dots & \dots & m_{2k} \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ m_{k1} & m_{k2} & \dots & \dots & 1 \end{bmatrix} \quad (22)$$

where, row vectors represent the antibody i to stimulate the other antibodies and column vectors represent the other antibodies to suppress the antibody i .

Let, the error between the real output value y_j and network response \hat{y}_i exist as follows:

$$e_i = \hat{y}_i - y_j \quad (23)$$

If absolute error $|e_i|$ is smaller than a threshold value e_{th} , the receptor vector x_i of the antibody i and the controlling variable σ_n of network response of the antibody i will be modified according to the following equation:

$$(x_i)_{new} = (x_i)_{old} + \alpha * e_i * (\Phi_j - x_i) \quad (24)$$

$$(\sigma_n)_{new} = (\sigma_n)_{old} + \beta * e_i \quad (25)$$

where, α and β are the modification coefficients for x_i and σ_n , respectively. The effect of tuning receptor vector x_i is to decrease the distance \bar{d}_{ij} between the i th antibody and j th antigen and thus increase their affinity value \bar{m}_{ij} as shown in Fig. 4. The adjusting of the variable σ_n results in changing the network response \hat{y}_i and consequently decreases its error e_i as shown in Fig. 5.

If error $|e_i|$ is larger than the threshold value e_{th} , the components of the affinity matrix M_{ik} will be altered

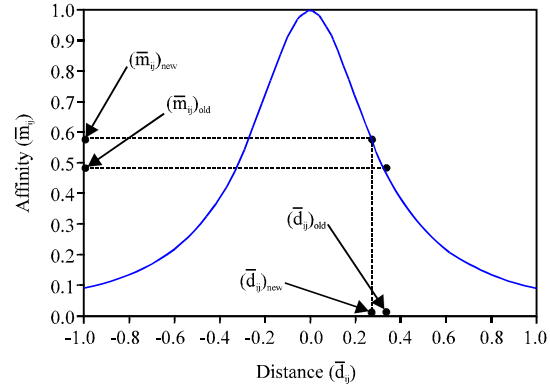


Fig. 4: The effect of tuning receptor vector x_i ($e_i > 0$)

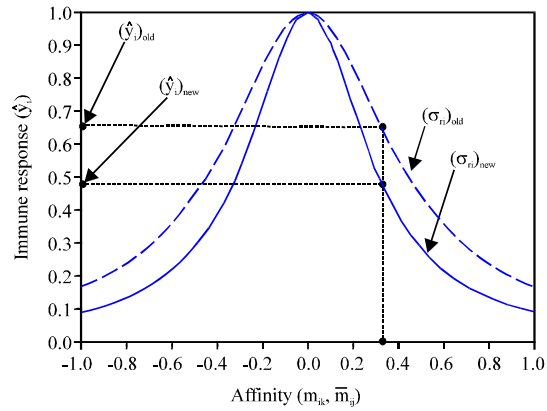


Fig. 5: The effect of adjusting of the variable σ_n ($e_i > 0$)

according to the following equations. Decreasing the affinity m_{ik} will cause the reduction of the concentration value of the i th antibody:

$$(m_{ik})_{new} = (m_{ik})_{old} - T(|e_i|) \quad (26)$$

and

$$T(|e_i|) = \eta \left[\frac{\exp(e_{th} - |e_i|) - \exp(|e_i| - e_{th})}{\exp(e_{th} - |e_i|) + \exp(|e_i| - e_{th})} \right] \quad (27)$$

where, η is an adjusting variable as shown in Fig. 6.

After the tuning procedure, new antibodies should replace the poorer antibodies to improve the overall performance of network. The i th antibody will be exterminated if its survival probability ρ_i is smaller than a tolerance value ρ_{tol} .

$$\rho_i = \frac{num_i}{N} \quad (28)$$

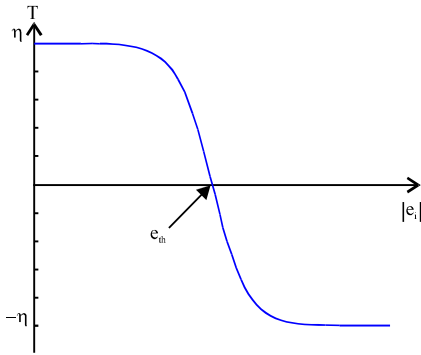


Fig. 6: The adjusting function $T(|e_i|)$

where, N is the total number of antigens utilized in network and num_i denotes the accumulated count when the i th antibody has the largest concentration value for the j th antigen. Replace these eliminated antibodies with the same number randomly selected from the antigens satisfying largest absolute error (i.e., $|e_i| > e_{tol}$).

SIMULATION AND TRAJECTORY CONTROL OF THE TWO-LINK ROBOT MANIPULATOR

Here, the presented model identification of the immune network will be used in the simulation system of the two-link robot manipulator. The simulation result will be discussed and the inverse model which the system has identified will be applied in the control of path tracking as well. The path tracking control of the robot manipulator will take a two-link robot manipulator's inverse model, identified by the immune network, as a mainframe and be complemented with a simple PID controller in order to conduct the entire tracking control. The entire control infrastructure of the inverse model is as shown in Fig. 7. In this framework, the inverse model identified by the immune network acts as a feedforward controller and a PID controller is used as dynamic compensation for the inverse model's output so that the path tracking of the robot manipulator can be controlled precisely.

Concerning the PID controller, the main purpose of it is to be the close-loop system control for the entire framework, for which the path error exported by the inverse model is compensated. In this article, considering a general type of PID controller, its transfer function is expressed as following:

$$G_c(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \quad (29)$$

In Eq. 29, K_p represents proportional gain constant, T_i integration time constant and T_d differentiating time constant. Then, change Eq. 29-30 as stated:

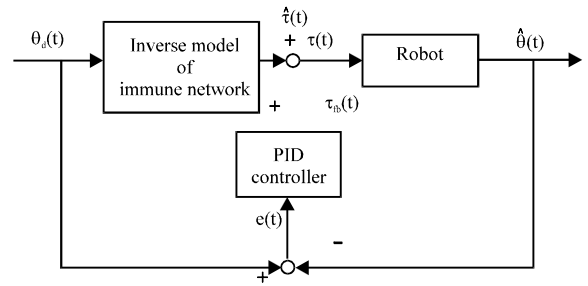


Fig. 7: Control infrastructure of inverse model using immune network

$$G_c(s) = K_p + \frac{K_i}{s} + K_d s \quad (30)$$

where, K_p is proportional gain constant, $K_i = (K_p/T_i)$ integration gain constant and $K_d = K_p T_d$ differentiating gain constant. In this structure, proportional control action is a gain-adjustable component. By changing the gain constant, system relative stability and its steady-state error can be increased. The integration control action is not only able to eliminate the steady-state error, but also to inhibit noise. However, it might also make the system unstable. Also, the differential control action is used to improve the system's damping characteristic and its transient response. This action relatively increases the system's stability. However, it also leads to the disadvantage of interference of high-frequency noise. In this research, we use traditional trial and error methods to adjust the three gain constants: K_p , K_i and K_d . In addition, the assistant inverse model produces reasonably good outcomes for the control of tracking different paths. The simulation result of this path-tracking control is illustrated as follows:

Initially, the planning for path-tracking uses the transforming formula below which is described by X-Y coordinate and the angle of robot manipulator:

$$\theta_2 = \cos^{-1} \left(\frac{X^2 + Y^2 - l_1^2 - l_2^2}{2l_1 l_2} \right) \quad (31)$$

$$\theta_1 = \tan^{-1} \left(\frac{Yl_1 + Yl_2 \cos \theta_2 - Xl_2 \sin \theta_2}{Xl_1 + Xl_2 \cos \theta_2 + Yl_2 \sin \theta_2} \right) \quad (32)$$

In replacing variables in Eq. 31 and 32 with X and Y coordinates of the planned path, the corresponding angles θ_1 and θ_2 from each axis of the robot manipulator are acquired. In this research, there are three kinds of path patterns for tracking control: circle, square and pentagram. When verifying the application of the inverse model on path-tracking control in different paths or in different positions by using the three shapes and testing each of

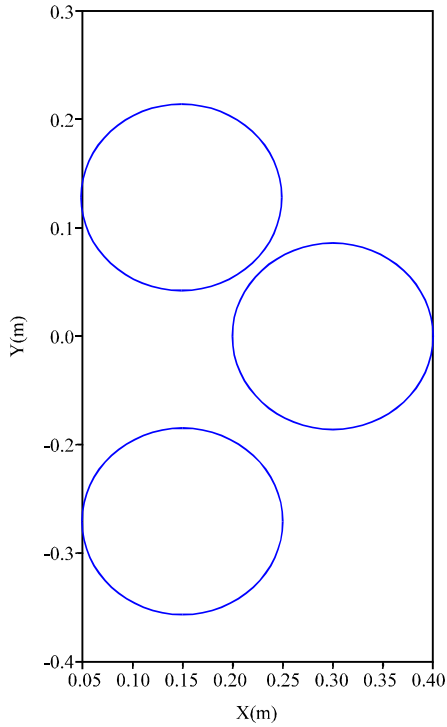


Fig. 8: Circle paths in different positions

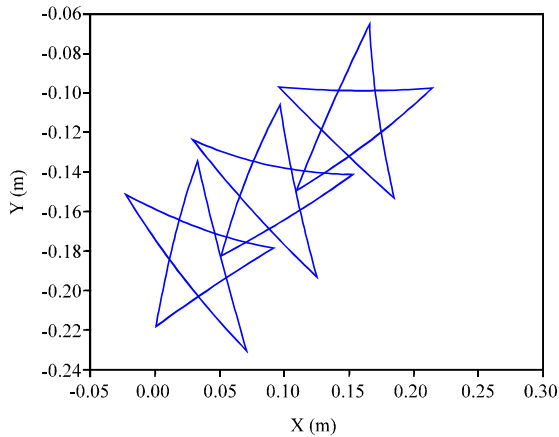


Fig. 9: Pentagram paths in different positions

them in three different positions, we always obtain a satisfying outcome. This size of each shape can be calculated by the following information: the circle with a 20-centimeter diameter, the square with a 10 cm side and the pentagram with a total of 16 cm length of sides. Each shape in each different position is schemed as 1001 dots as shown in Fig. 8 and 9. Corresponding to the inverse model of the immune network, this means proper immune responses are produced through antibodies which are

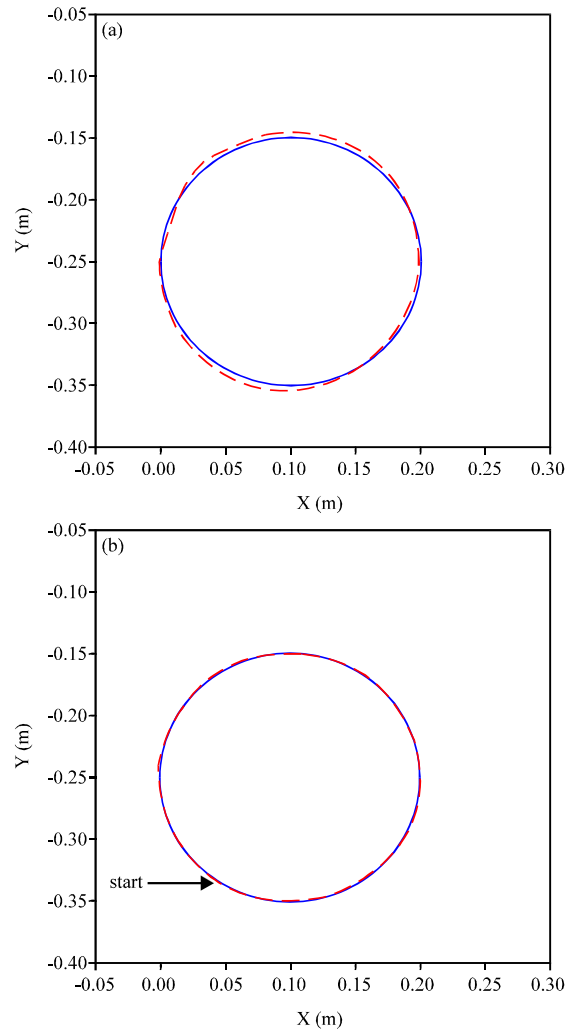


Fig. 10: Path comparison between the outcome path obtained from (a) inverse model and (b) with PID feedback controller

generated by the inverse model identified from the 1001 antigen input.

For the path-tracking control of different shapes in different positions, the simulation outcomes are discussed respectively as follows and the initial position of the robot manipulator is always set to be $(x, y) = (0.423, 0)$.

Circle path 1: Figure 10a compares the planned circle locus with the fifth loop path obtained from the inverse model identified by the immune network and not adding any PID feedback controller. The outcome shows the average error at each point is 1.2922 cm. However, Fig. 10b shows the average error changes to 0.0739 cm when the inverse model added a PID feedback controller. In this case, the PID gain value of each axis is as follows:

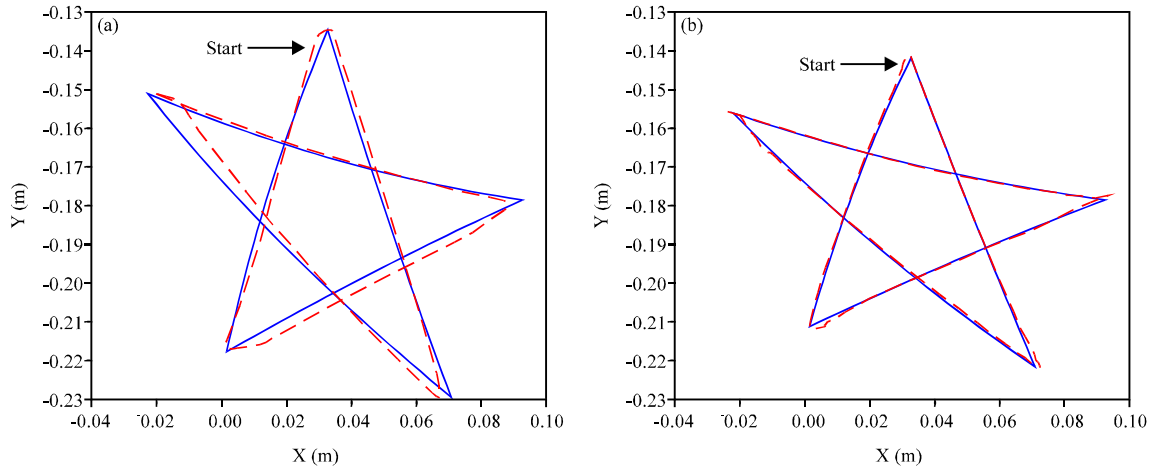


Fig. 11: Path comparison between the outcome path obtained from inverse model (a) with PID feedback controller (b)

- 1st axis $K_{p1} = 580, K_{i1} = 850, K_{d1} = 0.6$
- 2nd axis $K_{p1} = 250, K_{i1} = 720, K_{d1} = 0.5$

As a whole, when applying the inverse model to the tracking of a circle locus, the average error is between 0.06~0.075 cm. However, if using the Simple Genetic Algorithm (SGA) in searching for the best gain value of the PID controller, it is believed that the average error of each dot on the path will be reduced after dynamic compensation. Generally, it is more difficult for a robot manipulator to track a path of a straight line than that of a circular arc. The reason for this is that it is hard for a robot manipulator to be in control when on the turning point of a straight-line path. In this research, the inverse model is applied to path-tracking control of squares and pentagrams as well, so it can be verified that the tracking control also applies to a straight-line path.

Pentagram path 1: Figure 11 compares a and b the fifth loop path, generated from the inverse model which is identified by the immune network and not adding a PID feedback controller, to the planned locus of a pentagram. This comparison leads to an average error of 0.9759(cm) for dots in the figure. However, the average error in Fig. 11b reduced to 0.0955(cm) when adding a PID feedback control to the inverse model. In this case, the PID gain value for each axis is stated as follows:

- 1st axis $K_{p1} = 1050, K_{i1} = 2400, K_{d1} = 1.69$
- 2nd axis $K_{p1} = 300, K_{i1} = 400, K_{d1} = 0.54$

As far as the simulation result is concerned, it always performs well when applying the inverse model to path-tracking, no matter whether the path is in the form of

a straight line, square or pentagram. In terms of square path, the control result for the open-loop inverse model, after a dynamic compensation of a PID feedback controller, shows the average error for each dot on every different path is between 0.04 and 0.055 cm. As for the path of the pentagram under the same conditions, it shows the average error is around 0.096 cm. After observing all simulation results, it suggests that path of pentagram is the hardest shape for tracking control, due to too many turning points on the locus. It is believed that the average error on each path dot will be reduced more after dynamic compensation if applying Simple Genetic Algorithm (SGA) to the search for the best gain value of a PID controller.

CONCLUSION

The conclusion is to be aimed at the interaction of simulating each component in the biological network system; for example, in the different types of cells, to be stimulating and suppressing for each other. Additionally, the focus is on the basis of the behavior response for the viewpoint of immunology in the whole system; for example, as regards in the exterior substance invasion, the network system could be destroy them with prompt reaction. In this study, a novel nonlinear system NARX model identification scheme based on a network is proposed. Also, a Borland C++ Builder program language to complete the computer simulation software which has a user interface is utilized. A two-link rigid manipulator simulation is used to evaluate the proposed methodology. Finally, we have applied results in trajectory tracking control of a robot manipulator with potentially positive results.

REFERENCES

- Akhmetov, D.F., Y. Dote and S.J. Ovaska, 2001. Fuzzy neural network with general parameter adaptation for modeling of nonlinear time-series. *IEEE Trans. Neural Networks*, 12: 148-152.
- Babuska, R., J.A. Roubos and H.B. Verbruggen, 1998. Identification of MIMO systems by input-output TS fuzzy models. *Proceedings of the 1998 IEEE International Conference on Fuzzy Systems Proceedings, IEEE World Congress on Computational Intelligence May 4-9, IEEE Xplore, London*, pp: 657-662.
- Billings, S.A., S. Chen and M.J. Korenberg, 1989. Identification of MIMO non-linear system using a forward-regression orthogonal estimator. *Int. J. Control*, 49: 2157-2189.
- Enab, Y.M., 1995. Genetic algorithm for identifying self-generating radial basis neural networks. *Proceedings of the Fourth International Conference on Artificial Neural Networks, Jun. 26-28, IEEE Xplore, London*, pp: 65-70.
- Hwang, M.W., M.H. Kim and J.Y. Choi, 2000. Second-order multilayer perceptrons and its optimization with genetic algorithms. *Proc. 2000 Congress Evol. Comput.*, 1: 652-658.
- Jerne, N.K., 1973. The immune system. *Scientific Am.*, 229: 52-60.
- Leontaritis, I.J. and S.A. Billings, 1985. Input-output parametric models for non-linear systems. Part I: Deterministic non-linear systems. *Int. J. Control*, 41: 303-328.
- Li, Y., N. Sundarajan and P. Saratchandran, 2000. Dynamically structured radial basis function neural networks for robust aircraft flight control. *Proc. 2000 Am. Control Conf.*, 5: 3501-3505.
- Liu, G.P. and V. Kadiramanathan, 1999. Multiobjective criteria for neural network structure selection and identification of nonlinear systems using genetic algorithms. *IEEE Proc. Control Theor. Appl.*, 146: 373-382.
- Petridis, V., E. Paterakis and A.A Kehagias, 1998. Hybrid neural-genetic multimodel parameter estimation algorithm. *IEEE Trans. Neural Networks*, 9: 862-876.
- Sousa, J.M. and M. Setnes, 2000. Model predictive control: A data-driven approach using simple fuzzy tools. *Proceedings of the 9th IEEE International Conference on Fuzzy Systems, May 7-10, San Antonio, TX, USA.*, pp: 1017-1020.
- Venkatesh, S.R. and M.A. Dahleh, 2001. On system identification of complex systems from finite data. *IEEE Trans. Automatic Control*, 46: 235-257.
- Wang, H.W., H. He and K.D. Huang, 2000. A method of fuzzy modeling for non-linear systems. *Proceedings of the 3rd World Congress on Intelligent Control and Automation, 2000, Hefei, China*, pp: 2163-2166.
- Yu, W. and X.O. Li, 2001. Some new results on system identification with dynamic neural networks. *IEEE Trans. Neural Networks*, 12: 412-417.