



Journal of Applied Sciences

ISSN 1812-5654

science
alert

ANSI*net*
an open access publisher
<http://ansinet.com>

Modeling of a Five-Link Biped Robot Dynamics Using Neural Networks

N. Bigdeli, K. Afshar, B.I. Lame and A. Zohrabi
Department of Electrical Engineering,
Imam Khomeini International University, Qazvin, Iran

Abstract: In this study, a method for dynamic modeling of a five-link seven degree of freedom (DOF) biped robot has been developed. The method, which is based on neural networks, considerably reduces the complexities in solving the dynamic model equations of the biped robot. Seven neural networks have been synthesized in order for modeling of the seven DOF of the robot being the coordinates of the torso center of mass, the torso angle and the left and right knee and thigh angles. In order for generating data for neural network training, the robot dynamics while walking on a non-smooth two-dimensional surface has been considered. The input sets of the trained neural networks consist of four applied torques in addition to the last sample time value of each output. These fed-back outputs not only account for the system second order dynamics but also help on compensating the surface non-smoothness very well. Evaluation results are representative of high performance and much lower complexity of the trained networks with respect to the nonlinear second order robot dynamics model.

Key words: Biped robot, dynamic modeling, neural network, surface model, center of mass

INTRODUCTION

With the recent development of advanced humanoid robots, legged robots have become one of the main topics in advanced robotic research (Hitomi *et al.*, 2006; Mousavi *et al.*, 2008; Mousavi *et al.*, 2007). Projects addressing legged robots usually study the stability and mobility of these mechanisms in a range of environmental conditions for applications where wheeled robots are unsuitable. One of the main interests in this area is getting the robot to remain stable as it walks in a straight line, especially when the walking surface is not necessarily smooth (Lope *et al.*, 2003). There are two approaches for achieving this purpose. The first, called static balance, aims to maintain the projection of the Center of Masses (COM) of the robot inside the area inscribed by the feet that are in contact with the ground. The COM represents the unique point in an object or system which can be used to describe the system's response to external forces and torques. The second method, also referred to as dynamic balance, uses the Zero Moment Point (ZMP), which is defined as the point on the ground around which the sum of all the moments of the active forces equal zero (Lope, 2003; Vukobratovic *et al.*, 1990). If the ZMP is within the convex hull of all contact points between the feet and the ground, the biped robot is stable and will not fall over (Huang *et al.*, 2001; Furuta *et al.*, 2001). Due to their multivariable, high order, nonlinear and time-variant

dynamics modeling and control of biped robots is not a straight forward task and so is an open problem, yet (Mousavi *et al.*, 2008; Mousavi and Bagheri, 2007). While precise control strategies need to employ the robot dynamics in on line control strategies (Kong *et al.*, 2008; Zhou and Meng, 2003; Hurmuzlu *et al.*, 2004; Beigzadeh *et al.*, 2008; Hitomi *et al.*, 2006), solving highly complex and nonlinear dynamics equations of the biped robot model is not computationally-efficient (Kong *et al.*, 2008). On the other hand, solving the control problems regardless of the robot dynamics may degrade the robot control performance, especially when there exist some kinds of disturbances such as ground surface un-smoothness.

Based on the above discussions, development of an easy-to-use dynamic model which considers both the nonlinear properties of the biped robot as well as surface model seems very helpful. Therefore, in this study, dynamic modeling of a five-link robot in effect of joint torques and surface non-smoothness has been considered. Regarding the high performance of neural networks in modeling nonlinear complex systems behavior, here they have been employed as the modeling tool. For this purpose, proper model inputs has been chosen via a systematic approach and then the underlying neural networks for modeling the position of the center of mass of the torso and the robot joint angles being the torso angle and the left and right knee and thigh

angles has been trained. In order for generating data for neural network training, the robot dynamics while walking on a non-smooth two-dimensional surface has been considered. The input sets of the trained neural networks consist of four applied torques in addition to the last sample time value of each output. These fed-back outputs not only account for the system second order dynamics but also help on compensating the surface non-smoothness very well. So, the objective of the study is the development of method for dynamic modeling who is representative of high performance and much lower complexity of the trained networks with respect to the nonlinear second order robot dynamics model.

MATERIALS AND METHODS

A. Five-Link Biped Robot Dynamic Model

Biped robot dynamics: The biped robot considered in this study is a two-dimensional biped consists of five links which are connected with frictionless joints (Haavisto and Hyötyniemi, 2004). The identical legs have knee joints between the shank and thigh parts and one rigid body forms the torso. Figure 1a shows the model structure and variables used. As the system can move freely in the x-y-plane and contains five links, it has seven degrees of freedom. The corresponding seven coordinates are selected according to Fig. 1a as:

$$q = [x_0 \ y_0 \ \alpha \ \beta_L \ \beta_R \ \gamma_L \ \gamma_R]^T \quad (1)$$

The coordinates (x_0, y_0) stands for the position of the center of mass of the torso and the rest of the coordinates describe the joint angles as:

- α : Torso angle
- β_L and β_R : Left and right thigh angles
- γ_L and γ_R : Left and right knee angles

The link lengths are denoted as (l_0, l_1, l_2) and masses as (m_0, m_1, m_2) . The centers of mass of the links are located at the distances (r_0, r_1, r_2) from the corresponding joints. The model is actuated with four moments:

$$M = [M_{L1} \ M_{R1} \ M_{L2} \ M_{R2}]^T \quad (2)$$

Two of them acting between the torso and both thighs and two at the knee joints (Fig. 1b). The walking surface is modeled using external forces:

$$F = [F_{Lx} \ F_{Ry} \ F_{Lx} \ F_{Ry}]^T \quad (3)$$

that affect the both leg tips. When the leg should touch the ground, the corresponding forces are switched on to support the leg. As the leg rises, the forces are zeroed.

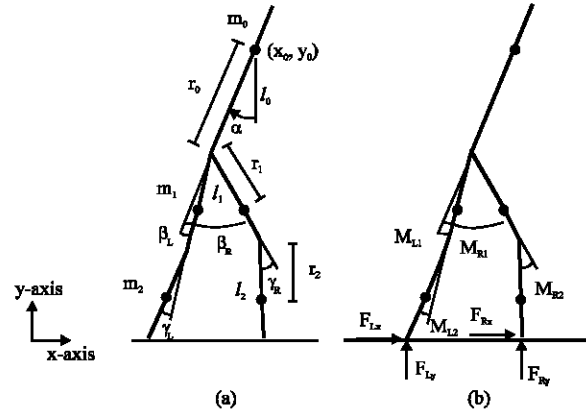


Fig. 1: (a) Biped model coordinates and constants, (x_0, y_0) : the position of the center of mass of the torso, α : torso angle, β_L and β_R : left and right thigh angles, γ_L and γ_R : left and right knee angles, (l_0, l_1, l_2) : the link lengths, (m_0, m_1, m_2) : the link masses and (r_0, r_1, r_2) : the distances of centers of mass of the links from the corresponding joints and (b) external forces, $M = [M_{L1} \ M_{R1} \ M_{L2} \ M_{R2}]^T$: actuating moments and $F = [F_{Lx} \ F_{Ry} \ F_{Lx} \ F_{Ry}]^T$: walking surface external forces (Haavisto and Hyötyniemi, 2004)

Using Lagrangian mechanics, the dynamic equations for the biped system can be derived as:

$$A(q)\ddot{q} = b(q, \dot{q}, M, F) \quad (4)$$

Here, $A(q) \in \mathbb{R}^{7 \times 7}$ is the inertia matrix and $b(q, \dot{q}, M, F) \in \mathbb{R}^{7 \times 1}$ is a vector containing the right hand sides of the seven partial differential equations. The closed form formulas for both A and b are listed in Appendix A (Haavisto and Hyötyniemi, 2004).

Ground contact forces model: The ground surface is modeled as a set of points (x, y) that are connected with straight lines. Figure 2 shows a sketch of one leg in touch with the ground. A new coordinate system (x', y') is now defined so that its origin is the next ground point to the negative x direction of the leg tip. The axis x' is aligned tangential to the ground surface and y' equals the surface normal direction as shown in Fig. 2. When the leg tip touches the ground at the point $(x'_0, 0)$, normal and tangential forces are applied to it. The normal force dynamic is calculated as (Haavisto and Hyötyniemi, 2004):

$$F_n = -k_y y'_0 - b_y \dot{y}'_0 \quad (5)$$

where, y'_0 is the current (negative) leg tip in y' coordinate, k_y the ground normal elastic constant and b_y the normal damping ratio. Additionally the normal force is limited to positive values to prevent the leg sticking to the ground.

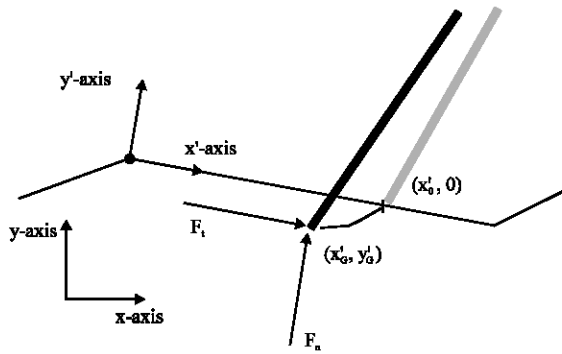


Fig. 2: The leg tip touches the ground in point $(x'_0, 0)$ (grey) and penetrates it. The current position of the leg tip is (x'_0, y'_0) (black). F_t and F_n are the tangential and normal touching forces, respectively. Note that the penetration is exaggerated for clarity (Haavisto and Hyötyniemi, 2004)

In the tangential direction the force F_t acting is caused by friction. The static friction force is determined similar to the normal force calculation (Haavisto and Hyötyniemi, 2004). Now the nominal value, however, is the initial touching point x'_0 :

$$F_t = -k_x(x'_0 - x'_0) - b_x \dot{x}'_0 \quad (6)$$

where, k_x and b_x are the ground tangential properties. If the required force exceeds the maximum static friction force

$$F_{t,max} = \mu_s F_n \quad (7)$$

μ_s being the static friction coefficient, the leg starts to slide. In that case the tangential force is:

$$F_t = \mu_k F_n \quad (8)$$

where, μ_k is the kinetic friction coefficient. The stored value of x'_0 is continuously set to the corresponding leg tip x'_0 position during the sliding. After the normal and tangential forces are computed, they need to be projected to the original coordinate system (x, y) to attain the dynamic model input forces of Eq. 3.

B. Modeling with Artificial Neural Networks (ANN's):

Regarding the high performance of neural networks in modeling nonlinear complex systems behavior, here we have employed them as our modeling tool. Here, we shortly review the basics of neural networks and then switch to the developed models and their performance.

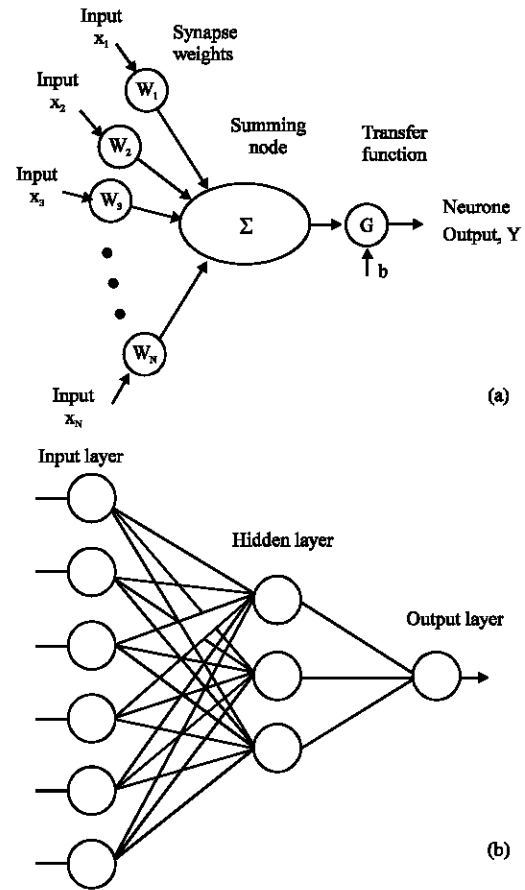


Fig. 3: (a) Internal structure of a neuron and (b) Example of a three-layered feed forward neural network

The fundamentals of ANNs: Neural networks are highly interconnected simple processing units designed in a way to model how the human brain performs particular task (Catalão *et al.*, 2007; Haykin, 1999). An ANN is composed of a number of interconnected neurons which are arranged in a few layers, called input, hidden and output layers. The output of each node is a weighted sum of its inputs added to a constant term called bias (Fig. 3).

Modeling with neural networks involves two steps: These two steps are Training and testing. In training, a proper ANN is constructed via some different stages. At first, proper inputs should be selected. This stage is the most important stage which is usually done through input-output linear/nonlinear correlation analyses and probably the experiment of the ANN developer about the underlying system. The selected inputs and the output of the network are then normalized to feed them to the network. The next stage is to choose the numbers of layers and nodes of each layer as well as the transfer

functions of the each neuron. This stage is normally a trial and error stages which is repeated until the best performance of network is achieved. Final stage in training includes in learning of the chosen network. A learning process in the neural network constructs an input-output mapping, adjusting the weights and biases at each iteration based on the minimization of some error measure between the produced and the desired output. Thus, learning entails an optimization process. There are different learning algorithms in the literature (Almeida, 1997). One of most efficient learning algorithms is the well-known Levenberg-Marquardt algorithm. This algorithm is actually a modified Gauss-Newton method that converges 10-100 times faster than the well-known back propagation algorithm. Newton's update for minimizing a function $V(\bar{x})$ with respect to the vector \bar{x} is given by:

$$\Delta \bar{x} = -[\nabla^2 V(\bar{x})]^{-1} \nabla V(\bar{x}) \quad (9)$$

where, $\nabla^2 V(\bar{x})$ is the Hessian matrix and $\nabla V(\bar{x})$ is the gradient vector. Assuming that $V(\bar{x})$ is the sum of square errors, given by:

$$V(\bar{x}) = \sum_{h=1}^N e_h^2(\bar{x}) \quad (10)$$

Then:

$$\begin{aligned} \nabla V(\bar{x}) &= 2J^T(\bar{x})e(\bar{x}) \\ \nabla^2 V(\bar{x}) &= 2J^T(\bar{x})J(\bar{x}) + 2S(\bar{x}) \end{aligned} \quad (11)$$

where, $e(\bar{x})$ is the error vector and $J(\bar{x})$ is the Jacobian matrix given by:

$$J_{i,j}(\bar{x}) = \frac{\partial e_i(\bar{x})}{\partial x_j}, \quad i = 1, \dots, N \text{ and } j = 1, \dots, n$$

where, N is the number of data in training set and n is the dimension of \bar{x} . $S(\bar{x})$ is:

$$S(\bar{x}) = \sum_{h=1}^N e_h(\bar{x}) \nabla^2 e_h(\bar{x})$$

Neglecting the second-order derivatives of the error vector (i.e., assuming that $S(\bar{x})=0$) and substituting the resulting Hessian matrix into Eq. 11 and 9 we obtain the Gauss-Newton update, as:

$$\Delta \bar{x} = -[J^T(\bar{x})J(\bar{x})]^{-1} J^T(\bar{x})e(\bar{x}) \quad (12)$$

The advantage of Gauss Newton over the standard Newton's method is that it does not require calculation of

second-order derivatives. Nevertheless, the matrix $J^T(\bar{x})J(\bar{x})$ may not be invertible. This is overcome with the Levenberg-Marquardt algorithm, which consists in finding the update given by:

$$\Delta \bar{x} = -[J^T(\bar{x})J(\bar{x}) + \mu I]^{-1} J^T(\bar{x})e(\bar{x}) \quad (13)$$

where, parameter μ is conveniently modified during the algorithm iterations. When μ is very small or null the Levenberg-Marquardt algorithm becomes Gauss-Newton, which should provide faster convergence, while for higher μ values, when the first term within square brackets of Eq. (13) is negligible with respect to the second term within square brackets, the algorithm becomes steepest descent (Catalão *et al.*, 2007; Saini and Soni, 2002).

Evaluation indices: As stated earlier, for the evaluation of the ANN's performance, a testing set containing new input data that it has never seen before is applied to the trained network. The performance of the trained network is then evaluated by comparison of the network output with its actual value. There are some statistical evaluation indices which are commonly used to judge about an ANN's performance. Let A_i and P_i be the actual and network output, respectively, related to i th input vector, where N is the number of points in the testing set. Then the evaluation indices are defined as (Pao, 2007):

- Root Mean Square Error (RMSE):

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (P_i - A_i)^2}{N}} \quad (14)$$

- Mean Absolute Error (MAE):

$$MAE = \frac{\sum_{i=1}^N |P_i - A_i|}{N} \quad (15)$$

- Mean Absolute Percentage Error (MAPE):

$$MAPE = \frac{1}{N} \sum_{i=1}^N \frac{|P_i - A_i|}{A_i} \times 100\% \quad (16)$$

- Modified mean absolute percentage error (Modified_MAPE): In the relationship in Eq. 16 if the actual value is large and its prediction becomes small, the computed relative error will become near 100%. On the other hand if the actual value is small, the relative error may become very large even though the difference is small. In this case, the relationship in Eq. 16 is modified in this manner. At first, the average of actual output values is computed as:

$$A_v = \frac{1}{N} \sum_{i=1}^N A_i$$

and then, the Modified MAPE will be computed as (Shahidehpour *et al.*, 2002):

$$\text{Modified_MAPE} = \frac{1}{N} \sum_{i=1}^N \frac{|P_i - A_i|}{A_v} \times 100\% \quad (17)$$

RESULTS AND DISCUSSION

Supported by the above mentioned dynamics model of the biped robot modeling with artificial neural networks methodology, in this section, the dynamics of the robot center of mass and the robot joint angles will be modeled by neural networks. The dynamic equation of robot is considered as the system model which will be used for biped robot simulation and data generation purposes. This model has been implemented in Bipedsim for MATLAB (2008) as well. The developed models are seven different neural networks which have been synthesized and evaluated via data sets of length 12000 generated by (Bipedsim). These data sets simulate walking of the underlying biped robot while surface smoothness varies as time goes on for 60 sec and sampled every 5 min. Figure 4 and 5 show the applied torques and variations of each parameter for the above mentioned period. In Fig. 4, the applied moments have been shown. Figure 5, show the resulting joint variables, i.e. y_0 , α , β_L and γ_L . x_0 , β_R and γ_R has not been shown here, because x_0 exhibits as a straight line with constant slope (i.e., it changes with a constant speed) and β_R and γ_R are closely similar to β_L and γ_L , respectively.

In order for synthesizing the neural networks for modeling of each robot output variable, different input sets as well as different number of layers with different

number of neurons has been considered. Due to complex nature of the robot dynamics, selection of proper inputs is a major task. In choosing input sets, one should note that the robot dynamic is a second-order one (Eq. 4). Therefore, in addition of the instantaneous values of applied input torques, some previous values of robot inputs and/or the underlying outputs should be considered as the neural networks' inputs. Considering complexity of the dynamic equations (Appendix A), up to 2 sample time previous values of each input torques and 2 sample-time previous values of the desired outputs as well as the combination of them have been proved, in this research. In this context, from different input sets the best performance of the developed networks have been achieved when the input sets consist of the applied four torques in addition to the last sample time value of the output. This fed-back output not only considers the system dynamic but also compensate for the surface non smoothness very well.

The trained networks for different variables are 5-input (4 applied moments and one fed-back output) neural networks with one hidden layer of 7 neurons and the output layer of one neuron. The transfer function of hidden layer neurons is chosen as hyperbolic tangent sigmoid transfer function, defined by:

$$G(x) = \frac{2}{1 + e^{-2x}} - 1 \quad (18)$$

The transfer function used for the output layer is a pure linear transfer function. For training the network, the neural network toolbox of MATLAB (2008) was selected due to its flexibility and simplicity (Catalão *et al.*, 2007). The network was trained by the Levenberg-Marquardt

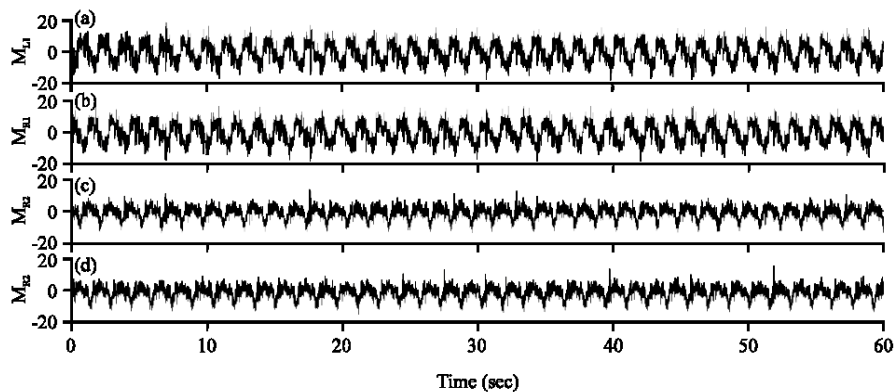


Fig. 4: Applied input torques to the biped robot, (a) M_{L1} (b) M_{R1} , (c), M_{L2} and (d) M_{R2}

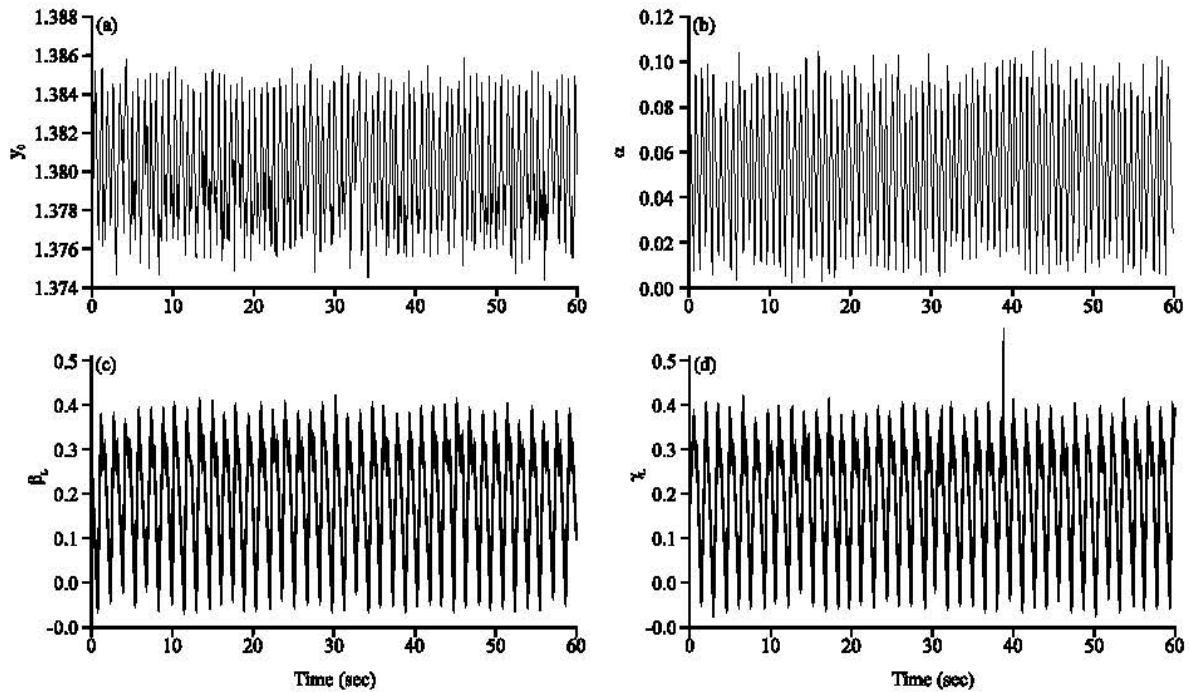


Fig. 5: The robot center of mass and the robot joint angles variations in response to applied torques in Fig. 4, (a) y_0 , (b) α , (c) β_L and (d) γ_L

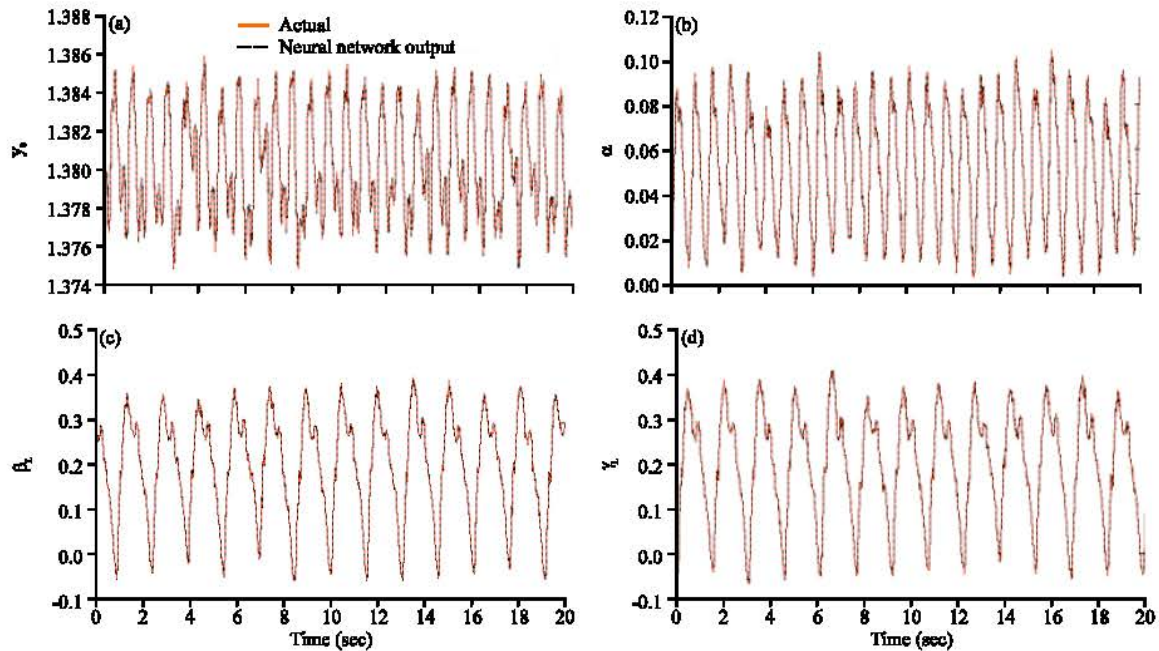


Fig. 6: Neural networks' evaluation results for (a) y_0 , (b) α , (c) β_L and (d) γ_L

algorithm (Saini and Soni, 2002). However, some other training methods were tested which amongst, the convergence rate and the prediction error for the selected method was the best. From the available data the last 40-sec data i.e. sets of 8000 data has been used for training

and the remaining data have been used for the network performance evaluation.

The evaluation results of synthesized neural networks have been shown in Fig. 6-8 and the evaluation indices have been brought in Table 1. Figure 6 shows the

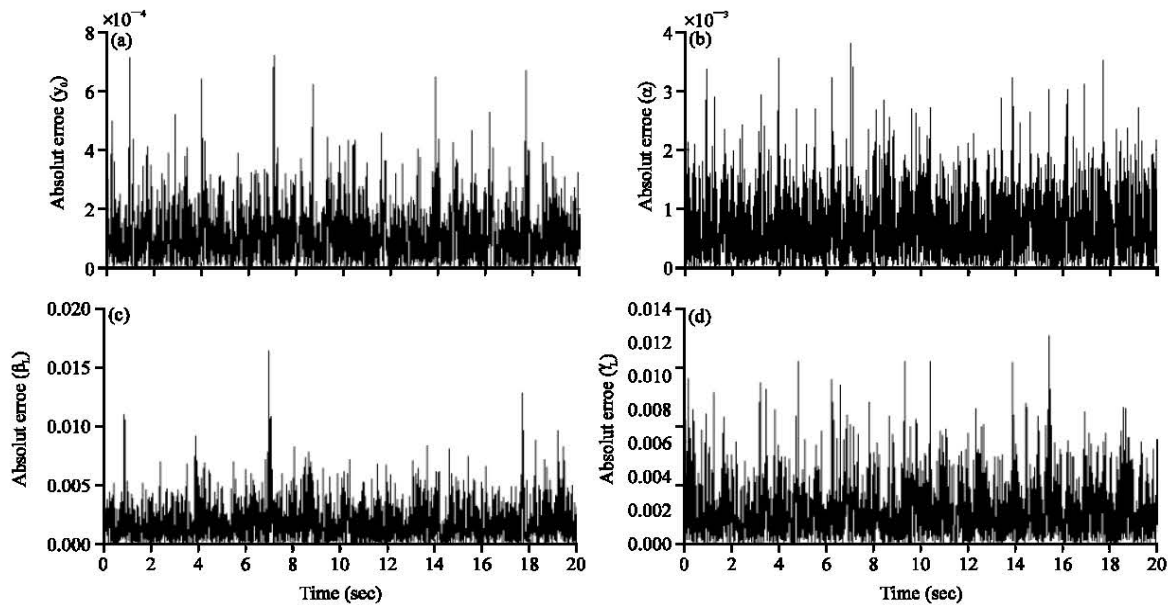


Fig. 7: Absolute modeling error of the Neural networks for (a) y_0 , (b) α , (c) β_L and (d) γ_L

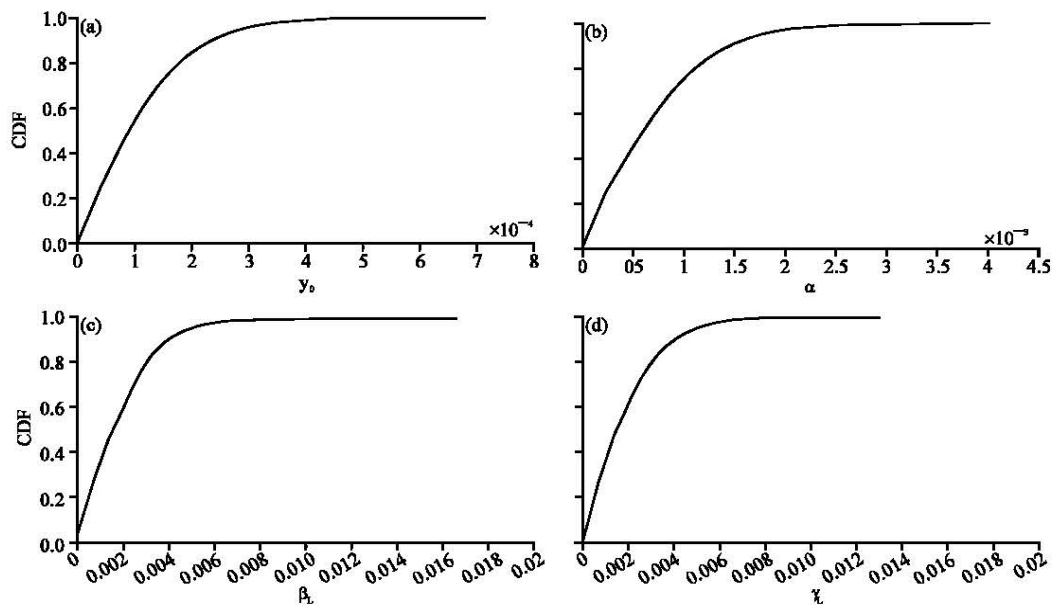


Fig. 8: Cumulative distribution function of the absolute modeling errors shown in Fig. 5, for (a) y_0 , (b) α , (c) β_L and (d) γ_L

Table 1: Evaluation indices for different ANN models

Index	Data set			
	y_0	α	β_L	γ_L
MAE	1.1168e-4	6.9002e-4	0.0020	0.0019
RMSE	1.4316e-4	8.8097e-4	0.0025	0.0025
Modified MAPE, (%)	0.0081	1.3053	0.9453	0.9433

networks' outputs in comparison with the actual data, while Fig. 7 and 8 show the absolute errors and their cumulative distribution functions, respectively.

As it is obvious from these figures, the biped robot dynamic has been very well modeled by the synthesized neural networks. Considering the simple structure of the developed networks in comparison with the robot dynamic equations, a major reduction in complexity are concluded. The evaluation results in Table 1 also emphasized the accuracy and high performance of the developed neural networks.

CONCLUSION

Employing the highly complex and nonlinear dynamics of the biped robots for their control/trajectory planning is a cumbersome task both in researches and practical implementations. On the other hand, try to solve the problems regardless of the robot dynamics may degrade the robot control performance especially where there exists disturbances such non-smooth ground surface or in case of robot dynamic mismatches. Based on these discussions, development of an easy-to-use dynamic model which considers both the nonlinear properties of the biped robot as well as the disturbance model seems very helpful. Therefore, in this study, dynamic modeling of a five-link biped robot in effect of joint torques and surface non-smoothness has been considered. Simplicity and high precision are the characteristics of the developed model, which is based on neural networks. In order for generating data for neural network training, the robot dynamics while walking on a non-smooth two-dimensional surface has been considered. Seven neural networks have been synthesized in order for modeling of the seven DOF of the robot. The input sets of the trained neural networks consist of four applied torques in addition to the last sample time value of each output. These fed-back outputs not only account for the system second order dynamics but also help on compensating the surface non-smoothness very well. Evaluation results are representative of high performance and much lower complexity of the trained networks with respect to the nonlinear second order robot dynamic model.

APPENDIX A

The dynamic model of the biped has the form:

$$A(q)\ddot{q} = b(q, \dot{q}, M, F) \quad (A1)$$

The structure of the biped and the parameters used are shown in the Fig. 1. The vector q contains the generalized coordinates, F the ground support forces and M the joint moments. As the system has seven degrees of freedom, there exists also seven partial differential equations. In the following, the exact formulas of the inertia matrix $A(q)$ and the right hand side vector $b(q, \dot{q}, M, F)$ are presented (13). $A(q)$ is a 7×7 symmetric matrix with its upper triangular elements as:

$$A_{11} = m_0 + 2m_1 + 2m_2$$

$$A_{12} = 0$$

$$A_{13} = -2(m_1 + m_2)r_0 \cos(\alpha) - (l_1 m_2 + m_1 r_1) \cos(\alpha - \beta_L) - (l_1 m_2 + m_1 r_1) \cos(\alpha - \beta_R) - m_2 r_2 [\cos(\alpha - \beta_L + \gamma_L) + \cos(\alpha - \beta_R + \gamma_R)]$$

$$A_{14} = (l_1 m_2 + m_1 r_1) \cos(\alpha - \beta_L) + m_2 r_2 \cos(\alpha - \beta_L + \gamma_L)$$

$$A_{15} = (l_1 m_2 + m_1 r_1) \cos(\alpha - \beta_R) + m_2 r_2 \cos(\alpha - \beta_R + \gamma_R)$$

$$A_{16} = -m_2 r_2 \cos(\alpha - \beta_L + \gamma_L)$$

$$A_{17} = -m_2 r_2 \cos(\alpha - \beta_R + \gamma_R)$$

$$A_{21} = 0$$

$$A_{22} = m_0 + 2m_1 + 2m_2$$

$$A_{23} = 2(m_1 + m_2)r_0 \sin(\alpha) + (l_1 m_2 + m_1 r_1) \sin(\alpha - \beta_L) + (l_1 m_2 + m_1 r_1) \sin(\alpha - \beta_R) + m_2 r_2 [\sin(\alpha - \beta_L + \gamma_L) + \sin(\alpha - \beta_R + \gamma_R)]$$

$$A_{24} = -(l_1 m_2 + m_1 r_1) \sin(\alpha - \beta_L) - m_2 r_2 \sin(\alpha - \beta_L + \gamma_L)$$

$$A_{25} = -(l_1 m_2 + m_1 r_1) \sin(\alpha - \beta_R) - m_2 r_2 \sin(\alpha - \beta_R + \gamma_R)$$

$$A_{26} = m_2 r_2 \sin(\alpha - \beta_L + \gamma_L)$$

$$A_{27} = m_2 r_2 \sin(\alpha - \beta_R + \gamma_R)$$

$$A_{33} = 2l_1^2 m_2 + 2(m_1 + m_2)r_0^2 + 2(m_1 r_1^2 + m_2 r_2^2) + 2(l_1 m_2 + m_1 r_1)r_0 [\cos(\beta_L) + \cos(\beta_R)] + 2m_2 r_2 [\cos(\beta_L - \gamma_L) + \cos(\beta_R - \gamma_R)] + 2l_1 m_2 r_2 [\cos(\gamma_L) + \cos(\gamma_R)]$$

$$A_{34} = -l_1^2 m_2 - (m_1 r_1^2 + m_2 r_2^2) - (l_1 m_2 + m_1 r_1)r_0 \cos(\beta_L) - m_2 r_2 \cos(\beta_L - \gamma_L) - l_1 m_2 r_2 \cos(\gamma_L)$$

$$A_{35} = -l_1^2 m_2 - (m_1 r_1^2 + m_2 r_2^2) - (l_1 m_2 + m_1 r_1)r_0 \cos(\beta_R) - m_2 r_2 \cos(\beta_R - \gamma_R) - l_1 m_2 r_2 \cos(\gamma_R)$$

$$A_{36} = m_2 r_2 [r_2 + r_0 \cos(\beta_L - \gamma_L) + l_1 \cos(\gamma_L)]$$

$$A_{37} = m_2 r_2 [r_2 + r_0 \cos(\beta_R - \gamma_R) + l_1 \cos(\gamma_R)]$$

$$A_{44} = l_1^2 m_2 + (m_1 r_1^2 + m_2 r_2^2) + 2l_1 m_2 r_2 \cos(\gamma_L)$$

$$A_{45} = 0$$

$$A_{46} = m_2 r_2 [-r_2 - l_1 \cos(\gamma_L)]$$

$$A_{47} = 0$$

$$A_{55} = l_1^2 m_2 + (m_1 r_1^2 + m_2 r_2^2) + 2l_1 m_2 r_2 \cos(\gamma_R)$$

$$A_{56} = 0$$

$$A_{57} = -m_2 r_2 [r_2 + l_1 \cos(\gamma_R)]$$

$$A_{66} = m_2 r_2^2$$

$$A_{67} = 0$$

$$A_{77} = m_2 r_2^2$$

and the elements of vector $b(q, \dot{q}, M, F)$ are:

$$b_1 = -2\dot{\alpha}^2 (m_1 + m_2)r_0 \sin(\alpha) + F_{Rx} + F_{Lx} - m_2 r_2 [\dot{\gamma}_L^2 + \dot{\beta}_L^2 + \dot{\alpha}^2 - 2(\dot{\gamma}_L \dot{\beta}_L - \dot{\gamma}_L \dot{\alpha} + \dot{\beta}_L \dot{\alpha})] \sin(\alpha - \beta_L + \gamma_L) + (m_1 r_1 + l_1 m_2) [(2\dot{\beta}_R \dot{\alpha} - \dot{\beta}_R^2 - \dot{\alpha}^2) \sin(\alpha - \beta_R) + (2\dot{\beta}_L \dot{\alpha} - \dot{\beta}_L^2 - \dot{\alpha}^2) \sin(\alpha - \beta_L)] - m_2 r_2 [\dot{\gamma}_R^2 + \dot{\beta}_R^2 + \dot{\alpha}^2 + 2(\dot{\gamma}_R \dot{\beta}_R - \dot{\gamma}_R \dot{\alpha} + \dot{\beta}_R \dot{\alpha})] \sin(\alpha - \beta_R + \gamma_R)$$

$$b_2 = -2\dot{\alpha}^2 (m_1 + m_2)r_0 \cos(\alpha) - m_2 r_2 [\dot{\gamma}_L^2 + \dot{\beta}_L^2 + \dot{\alpha}^2 - 2(\dot{\gamma}_L \dot{\beta}_L - \dot{\gamma}_L \dot{\alpha} + \dot{\beta}_L \dot{\alpha})] \cos(\alpha - \beta_L + \gamma_L) + F_{Ly} + F_{Ly} - m_2 r_2 [\dot{\gamma}_R^2 + \dot{\beta}_R^2 + \dot{\alpha}^2 + 2(\dot{\gamma}_R \dot{\beta}_R - \dot{\gamma}_R \dot{\alpha} + \dot{\beta}_R \dot{\alpha})] \cos(\alpha - \beta_R + \gamma_R) - g(m_0 + 2m_2) + (m_1 r_1 + l_1 m_2) [(2\dot{\beta}_R \dot{\alpha} - \dot{\beta}_R^2 - \dot{\alpha}^2) \cos(\alpha - \beta_R) + (2\dot{\beta}_L \dot{\alpha} - \dot{\beta}_L^2 - \dot{\alpha}^2) \cos(\alpha - \beta_L)]$$

$$b_3 = m_2 r_2 l_1 [(\dot{\gamma}_R^2 - 2\dot{\gamma}_R \dot{\beta}_R) \sin(\gamma_R) + (\dot{\gamma}_L^2 - 2\dot{\gamma}_L \dot{\beta}_L) \sin(\gamma_L)] + r_0 \sin(\alpha) [F_{Ly} + F_{Ly} - 2g(m_1 + m_2)] - l_1 [F_{Rx} \cos(\alpha - \beta_R) + F_{Ry} \sin(\alpha - \beta_R) + F_{Lx} \cos(\alpha - \beta_L) + F_{Ly} \sin(\alpha - \beta_L)] - l_1 [F_{Rx} \cos(\alpha - \beta_R + \gamma_R) - F_{Ry} \sin(\alpha - \beta_R + \gamma_R) + F_{Lx} \cos(\alpha - \beta_L + \gamma_L) - F_{Ly} \sin(\alpha - \beta_L + \gamma_L)] - r_0 \cos(\alpha) (F_{Ly} + F_{Ry}) - g m_2 r_2 [\sin(\alpha - \beta_L + \gamma_L) + \sin(\alpha - \beta_R + \gamma_R)] - m_2 r_2 l_1 [(\dot{\gamma}_L^2 + \dot{\beta}_L^2 - 2(\dot{\gamma}_L \dot{\beta}_L - \dot{\gamma}_L \dot{\alpha} + \dot{\beta}_L \dot{\alpha})) \sin(\beta_L - \gamma_L) + (\dot{\gamma}_R^2 + \dot{\beta}_R^2 - 2(\dot{\gamma}_R \dot{\beta}_R - \dot{\gamma}_R \dot{\alpha} + \dot{\beta}_R \dot{\alpha})) \sin(\beta_R - \gamma_R)] - (m_1 r_1 + m_2 r_2) l_1 [(\dot{\beta}_R^2 - 2\dot{\beta}_R \dot{\alpha}) \sin(\beta_R) + (\dot{\beta}_L^2 - 2\dot{\beta}_L \dot{\alpha}) \sin(\beta_L)] - g(l_1 m_2 + m_1 r_1) [\sin(\alpha - \beta_R) + \sin(\alpha - \beta_L)]$$

$$b_4 = M_{x1} + F_{Lx} l_1 \cos(\alpha - \beta_L) + F_{Lx} l_2 \cos(\alpha - \beta_L + \gamma_L) - (F_{Ly} l_1 - g m_2 l_1 - g m_2 r_1) \sin(\alpha - \beta_L) - \dot{\alpha}^2 (m_1 r_1 + l_1 m_2) r_0 \sin(\beta_L) - \dot{\alpha}^2 m_2 r_2 \sin(\beta_L - \gamma_L) - l_1 m_2 r_2 [\dot{\gamma}_L^2 + 2(\dot{\alpha} - \dot{\beta}_L) \dot{\gamma}_L] \sin(\gamma_L) - (F_{Ly} l_2 - g m_2 r_2) \sin(\alpha - \beta_L + \gamma_L)$$

$$\begin{aligned} b_5 &= M_{R1} + F_{R2}l_1 \cos(\alpha - \beta_R) + F_{R2}l_2 \cos(\alpha - \beta_R + \gamma_R) - (F_{R2}l_1 - g m_2 l_1 - g m_2 r_1) \sin(\alpha - \beta_R) \\ &\quad - \dot{\alpha}^2 (m_1 r_1 + l_1 m_2) r_1 \sin(\beta_R) - \dot{\alpha}^2 m_2 r_1 r_2 \sin(\beta_R - \gamma_R) - l_1 m_2 r_2 [\dot{\gamma}_R^2 + 2(\dot{\alpha} - \dot{\beta}_R) \dot{\gamma}_R] \sin(\gamma_R) \\ &\quad - (F_{R2}l_2 - g m_2 r_2) \sin(\alpha - \beta_R + \gamma_R) \\ b_6 &= M_{L2} - F_{L2}l_2 \cos(\alpha - \beta_L + \gamma_L) + \dot{\alpha}^2 m_2 r_1 r_2 \sin(\beta_L - \gamma_L) - l_1 m_2 r_2 [\dot{\alpha}^2 + 2(\dot{\alpha} - \dot{\beta}_L) \dot{\beta}_L] \\ &\quad \sin(\gamma_L) - (F_{L2}l_2 - g m_2 r_2) \sin(\alpha - \beta_L + \gamma_L) \\ b_7 &= M_{R2} - F_{R2}l_2 \cos(\alpha - \beta_R + \gamma_R) + \dot{\alpha}^2 m_2 r_1 r_2 \sin(\beta_R - \gamma_R) - l_1 m_2 r_2 [\dot{\alpha}^2 + 2(\dot{\alpha} - \dot{\beta}_R) \dot{\beta}_R] \\ &\quad \sin(\gamma_R) - (F_{R2}l_2 - g m_2 r_2) \sin(\alpha - \beta_R + \gamma_R) \end{aligned}$$

REFERENCES

- Almeida, L.B., 1997. Multilayer Perceptrons. In: Handbook of Neural Computation, Fiesler, E. and R. Beale, (Eds.). Institute of Physics, Oxford University Press, Oxford, ISBN-10: 075030524X.
- Beigzadeh, B., M.N. Ahmadabadi, A. Meghdari and A. Akbarimajd, 2008. A dynamic object manipulation approach to dynamic biped locomotion. *Robotics Autonomous Syst.*, 56: 570-582.
- Catalão, J.P.S., S.J.P.S. Marian, V.M.F. Mendesb and L.A.F.M. Ferreira, 2007. Short-term electricity prices forecasting in a competitive market: A neural network approach. *Electrical Power Syst. Res.*, 77: 1297-1304.
- Furuta, T., T. Tawara, Y. Okumura, M. Shimizu and K. Tomiyama, 2001. Design and construction of a series of compact humanoid robots and development of biped walk control strategies. *Robotics Autonomous Syst.*, 37: 81-100.
- Haavisto, A.O. and B.H. Hyötyniemi, 2004. Simulation tool of a biped walking robot model. Technical Report, Helsinki University Technology Control Engineering Laboratory, March 2004. www.control.hut.fi/Publications/Haavisto-2004/Simulator/bipedsim.pdf.
- Haykin, S., 1999. *Neural Networks: A Comprehensive Foundation*. 1st Edn., Prentice-Hall International Inc., New Jersey, ISBN: 0-13-908385-5.
- Hitomi, K., T. Shibata, Y. Nakamura and S. Ishii, 2006. Reinforcement learning for quasi-passive dynamic walking of an unstable biped robot. *Robotics Autonomous Syst.*, 54: 982-988.
- Huang, Q., K. Yokoi, S. Kajita, K. Kaneko, H. Arai, N. Koyachi and K. Tanie, 2001. Planning walking patterns for a biped robot. *IEEE Trans. Robotics Automation*, 17: 280-289.
- Hurmuzlu, Y., F. Génot and B. Brogliato, 2004. Modeling, stability and control of biped robots-a general framework. *Automatica*, 40: 1647-1664.
- Kong, J.S., E.H. Lee, B.H. Lee and J.G. Kim, 2008. Study on the real-time walking control of a humanoid robot using fuzzy algorithm. *Int. J. Control Automation Syst.*, 6: 551-558.
- Lope, J.d., T. Zarraonandia, R.G. Careaga and D. Maravall, 2003. Solving the inverse kinematics in humanoid robots: A neural approach. 7th International Work-Conference on Artificial Neural Networks, IWANN 2003, pp: 177-184. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.13.8318>.
- MATLAB, 2008. The MathWorks. <http://www.mathworks.com>.
- Mousavi, P.N. and A. Bagheri, 2007. Mathematical simulation of a seven link biped robot on various surfaces and ZMP considerations. *Applied Math. Model.*, 31: 18-37.
- Mousavi P.N., C. Nataraj, A. Bagheri and M.A. Entezari, 2008. Mathematical simulation of combined trajectory paths of a seven link biped robot. *Applied Math. Model.*, 32: 1445-1462.
- Pao, H.T., 2007. Forecasting electricity market pricing using artificial neural networks. *Energy Conversion Manage.*, 48: 907-912.
- Saini, L.M. and M.K. Soni, 2002. Artificial neural network based peak load forecasting using Levenberg-Marquardt and quasi-Newton methods. *IEEE Proc. Generation Transmission Distribut.*, 149: 578-584.
- Shahidehpour, M., H. Yamin and Z. Li, 2002. *Market Operations in Electric Power Systems: Forecasting, Scheduling and Risk Management*. 1st Edn., John Wiley and Sons Inc., New York, ISBN-10: 0471443379.
- Vukobratovic M., B. Borovac, D. Surla and D. Stokic, 1990. *Biped Locomotion: Dynamics, Stability, Control and Application*. 1st Edn., Springer-Verlag, Berlin-Heidelberg, Germany, ISBN-10: 0387174567.
- Zhou, C. and Q. Meng, 2003. Dynamic balance of a biped robot using fuzzy reinforcement learning agents. *Fuzzy Sets Syst.*, 134: 169-187.