3ème conférence Internationale des énergies renouvelables CIER-2015
Proceedings of Engineering and Technology - PET
pp.168-173

Identification of two-mass system parameters using neural networks

GHOZZI Dorsaf [1],NOURI KHaled[2]
Rached Dhaouadi[3]

[#]*Laboratoire de Recherche des systèmes Avancés, Ecole Polytechnique de Tunisie*
B.P. 748 La Marsa, 2078 Tunisie
[1]ghozzidorsaf113@gmail.com
[2]khaled.nouri@gmail.com

Department of Electrical Engineering, school of engineering, American University of Sharjah
PO Box 26666, Sharjah
United Arab Emirates
[3]rdhaouadi@aus.edu

Abstract— **This paper presents the design and development of a neural network method for the identification of elastic drive systems. First, a two-mass model (2MM) system is formulated and a feedforward neural network architecture is developed to identify the inverse model of the system. The multilayer perceptron (MLP) is trained off-line by using the gradient back propagation (BP) algorithm to adjust the network weights. Next, a frequency analysis method is used for the identification of the 2MM system parameters including the resonant frequency and damping factor. Simulation results are used to verify the effectiveness of the proposed method.**

***Keywords*— Elastic tow-mass system, multilayer neural network, back propagation algorithm, neural network identification, frequency analysis.**

I. Introduction

Constrained parameters identification has been extensively studied in the past decades and has been widely applied in scientific and engineering areas, such as signal processing, robot control, image fusion, filter design, pattern recognition, regression analysis [1–4].

In practical applications, these optimization problems have a time varying characteristics, so it is essential to solve the optimal solution in real time [2]. However, most of conventional algorithms based on general-purpose digital computers may not be very efficient since the computing time required for a solution broadly relies on the dimension and structure of these optimization problems. One promising approach for handling real-time optimization is to employ artificial neural networks based on circuit implementation [5–8]. As a result of the inherent massive parallelism, the neural network approach can solve optimization problems in running time much faster than those of the most traditional optimization algorithms executed on digital computers [9].

The introduction of artificial neural networks was first proposed by Tank and Hopfield [10] in 1986. In recent years, artificial neural networks (ANN) have introduced new aspects to solve complex nonlinear uncertain problems by their strong learning ability and high ability of parallel computing. ANN received a lot of attention in process control, system identification and many other domains [6–7]. For example, Kennedy and Chua [11] presented a neural network which contains finite penalty parameters and generates approximate solution for solving nonlinear programming problems. Studies in [13-14] indicated that neural networks could be used effectively in identifying nonlinear systems. Their papers proposed static and dynamic back-propagation algorithms to optimally generate the weights of neural networks and to adjust of parameters [3].

The different sections of this paper are organized as follows: in section 2, we define the model structure of the elastic drive system. In section 3, we present the considered neural networks and back-propagation algorithm used for updating the weighs and bias parameters.

Simulation results of the identification and control system are illustrated in section 4, and a conclusion is drawn in section 5.

II. Model Structure of the Elastic Drive System

A. System overview

The system considered in our analysis is a special class of electromechanical systems with an elastic tow-mass model structure. It is composed of a motor inertia connected to a load inertia with an elastic shaft as shown in fig. 1.

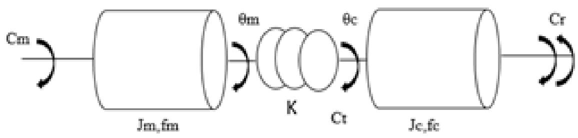

Fig. 1 Elastic two masses system [14]

Where C_m: motor torque, C_r: load torque, J_m: motor inertia, J_c: load inertia, f_m: motor damping coefficient, f_c: load damping coefficient, θ_m: motor angular position, θ_c: load angular position, C_t: shaft torque, K: stiffness shaft

ISSN: 2356-5608

B. Inverse model of 2MM system

In this section, we will present the dynamic equation by using motor speed and load speed instead of motor and load angular position. The dynamic equations are given as:

$$J_m \frac{dW_m}{dt} = C_m - C_t$$

$$J_c \frac{dW_c}{dt} = C_t \; - \; C_r \qquad (1)$$

$$C_t = K \int (w_m - w_c)\, dt + \beta (w_m - w_c)$$

To identify the block diagram of our system (fig.2), we use Laplace transform method.

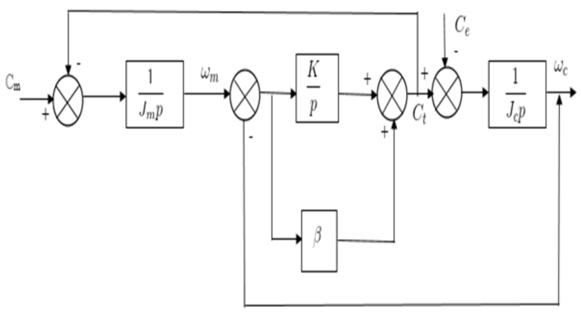

Fig. 2 2MM system block diagram

The load speed is usually not available for measurement. The transfer function between the motor torque and the load speed in the absence of load torque is given as:

$$H(p) = \frac{1}{J_m(p)} \left[\frac{p^2 + 2\zeta_z w_z p + w_z^2}{p^2 + 2\zeta_n np + w_n^2} \right] \qquad (2)$$

Where:

ωn:natural resonance angular frequency of the system

ωz:anti-resonance pulsation system

$\xi n, \xi z$: Damping coefficient

To keep constant the DC signal at the input of the system, between two instants of conversion, it is necessary to determine its discrete time model, it can be written as [3, 5]:

$$y(k+1) = a_0 y(k) + a_1\, y(k-1) + a_2\, y(k-2) + b_0\; u(k) \qquad (3)$$

where k is the discrete time index, *u(k)* and *y(k)* denotes the system input (motor torque) and output (motor speed) respectively. The parameters a and b are based on the mechanical system parameters and the sampling period [10, 11].

Assuming that the system represented by (eq.4) is invertible. i.e. there exist a function *g(.)* such that

$$u(k) = g(x_c) \qquad (4)$$

Where

$$x_c(k) = [y(k+1), y(k), y(k-1), y(k-2), u(k-2), u(k-1)] \quad (5)$$

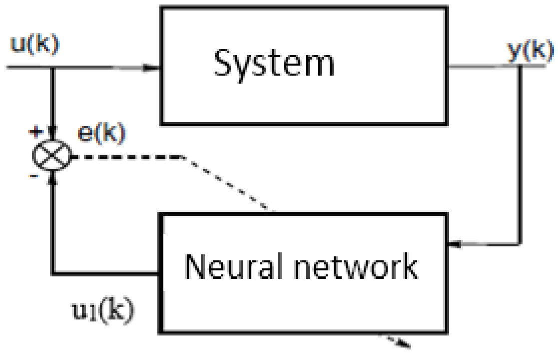

Fig. 3 Training configuration of MLP for identification 2MM inverse model [15]

Our objective, therefore, is to design a neural network that performs this mapping and identified the direct model of 2MM, for this, we consider a neural network consisting of an input layer with n_i neurons, a hidden layer with n_h neurons and the output layer with one neuron (n_o=1).

III. NEURAL NETWORK IDENTIFICATION OF 2MM SYSTEM

A. Feed forward multilayer neural network

In the last years, various neural network models have been developed for different applications including signal processing, pattern recognition and system modeling (Cho et al., 2008). The multi-layer perceptron with back-propagation learning is the most popular and commonly used neural network structure due to its simplicity, effectiveness and excellent performance in many applications that require to learn complex patterns (Kuo et al., 2014; Ruan and Tan, 2009). Multi-Layer perceptron is a feed-forward neural network with one or more hidden layers between input and output layer. Feed-forward means that data flows are in the forward direction, from input to output layer. MLP can solve problems, which are not linearly separable (Kirubavathi Venkatesh and Anitha Nadarajan, 2012).

Let's consider a multilayer feedforward network with three layers: an input layer with 6 neurons, one hidden layer with sigmoid neurons and an output layer with one linear neuron as shown in fig.4.

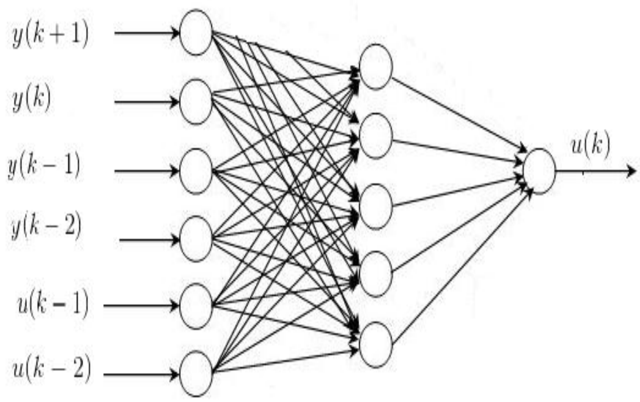

Fig.3 Multilayer perceptron architecture [16]

Let y be the input to the neural network, and *u, S* the output from output layer and the hidden layer respectively. The weights and biases between the output and the hidden layer are represented by w^h and θ^h, while between the hidden and output layer are w^o and θ^o with a desired output *d*, the overall sum-squared error of the network can be written as:

$$E_p = \frac{1}{2}\sum_k (d_{pk} - y_{pk})^2 \qquad (6)$$

and *u*, *S*, *y* are related by

$$y_{pk} = \mathrm{f}(e^o_{pk})$$
$$S^h_{pj} = f(e^h_{pj})$$

Where

$$e^o_{pk} = \sum_{j=1}^{n_h} W^o_{kj} S^h_{pj} + \theta^o_k$$
$$e^h_{pj} = \sum_{i=1}^{4} W^h_{ji} y(k+2-i) + \sum_{i=5}^{6} W^h_{ji}\ u(k+4-i) + \theta^h_j$$

and *f(.)* is the sigmoid activation function.

B. *Neural network training algorithm:Backpropagation*

In the training phase of the MLP, the training set is presented at the input layer and the parameters of the network (weights and biases) are dynamically adjusted using gradient-descent based delta-learning rule (back-propagation learning) to achieve the desired output (Gomathy and Lakshmipathi, 2011; Barabas et al.,2011). The training process of MLP neural network is defined as follows:

Step 1 – network initialization: The connection weights and bias of the network are initialized randomly, setting up the network learning rate *η*, the error threshold ε, and the maximum iterations T.

Step 2 – data preprocessing: Data samples are usually partitioned into three sets: training, validation and test. The training sets are used for training (to adjust the weights and biases) the network; the validation sets are the part that assesses or validates the predictive ability of the model during the training to minimize over fitting; the test sets are used for independent assessment of the model's predictive ability (generalization performance) after training.

Step 3 – training network: Input the training sets into MLP, compute network predicted output values, and calculate the error E between output and the target value.

Step 4 – updating the weights and biases: Update network weights and biases according to the prediction error E, making the predictive value of the network as close to actual values through a Back-propagation algorithm.

Step 5 – judgment of whether the end condition is met: If E≤ε, network training is stopped and go to step 7.

Step 6 – judgment of whether an over fitting has occurred: If accuracy of the validation error has not been satisfied network training is stopped and go to step 7; otherwise, return to step 3 to continue training.

Step 7 – judgment of generalization performance: Run test data set by trained network for generalization performance measurement.

IV. SIMULATION

In this section, we conduct several experiments to demonstrate the effectiveness of the proposed algorithm. The training data-pattern used in our study is generated off-line from an experimental setting which represents the 2MM system.

To display the results clearly, simulation is divided into two parts. One part presents the learning results of the neural network in Matlab. Another part is achieved as a numerical result which includes identification of parameters of 2MM system using frequency analysis.

A. *Training results*

The MLP consists of an input layer with 6 neurons, one hidden layer with 6 sigmoidal neurons and an output layer with one neuron. The number of neurons in the hidden layer is determined empirically. The optimal number is obtained by simulation.

In this phase, using back propagation algorithm, the weights and biases are adapted by the gradient descent methodology and the derivatives of an error function. If the output pattern is different from the target output, an error will be obtained and then propagated backward through the network from the output layer to the input layer. The weights will be modified as the error is propagated, and the modified network will output the pattern that is closer to the desired output: the error between output pattern and target pattern is to be reduced as illustrated in fig 6.

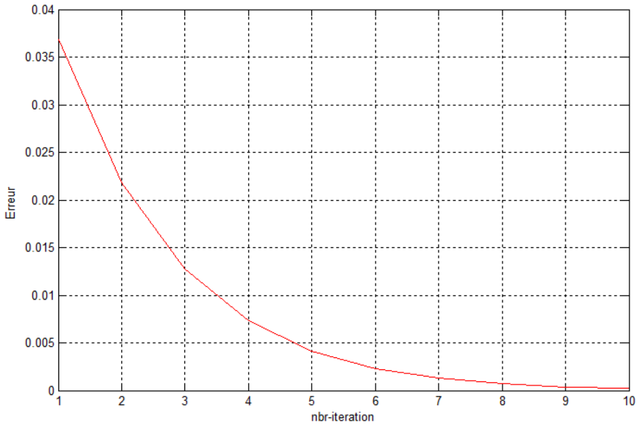

Fig.4 Total squared error during training.

The error approaches zero for a small number of iterations (10 iterations), this demonstrates the right choice of neural network architecture. Also, the speed and accuracy of neural networks are provided.

Update network weights and biases according to the off-line training

Under the same conditions, the BP algorithm is tested using any signal. Fig 5 shows that although the reference signal used for learning, the system response is very close to the desired response.

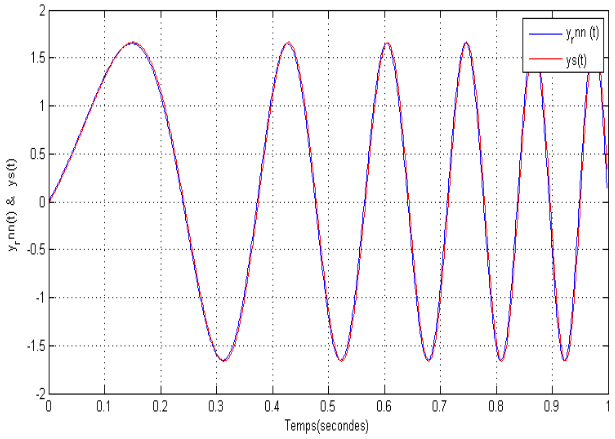

Fig.5 NN output after training ($y_{r,nn}$:neural model,y_s: test model)

These results confirm the generalization performance of neural networks.

B. *Identification results*

The frequency response function (FRF) is, in general, a complex valued function or waveform defined over a frequency range. Therefore, the process of identifying parameters from this type of measurement is commonly called curve fitting, or parameter estimation. Authors were looking for a better method for doing curve fitting in a mini-computer based modal analysis system. This type of system is used to make a series of FRF measurements on a structure, and then perform curve fitting on these measurements to identify the dynamic properties of structures and systems.

This paper presents the amplitude response of the system by the application of different sinusoidal signals. This procedure is applied with wide frequency range *[0.1 Hz; 400 Hz]*. Fig shows the frequency response of the system.

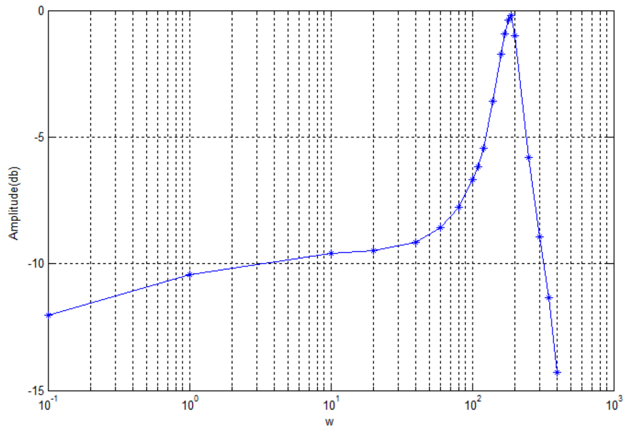

Fig.6 Frequency response of the system using NN

Using the experimental results, Eq7 and Eq8, we can identify the 2MM system parameters.

$$Q = \frac{1}{2\xi} \qquad (9)$$

$$w_r = w_n\sqrt{1-\xi^2} \qquad (10)$$

Where

w_r :resonance pulsation

w_n :natural pulsation

Q :quality factor

ξ :damping coefficient

The frequency response gives the following values:

$W_p = 191.6002\ rad.s^{-1}$
$\xi_p = 0.0912$

To compare our results, we should calculate theoretical parameters using the 2MM standardized parameters (Table.1).

TABLE I
STANDARDIZED PARAMETERS OF 2MM SYSTEM

Jm	0.122 [s]
Jc	0.108 [s]
K	1507 [s^{-1}]
β	1.504

We use the following formulas to calculate the theoretical parameters:

$$w_{th} = \sqrt{k(\frac{1}{J_m} + \frac{1}{J_c})}$$

$$\xi_{th} = \frac{\beta}{2}\sqrt{\frac{1}{K}\left(\frac{1}{J_m} + \frac{1}{J_c}\right)} \qquad (11)$$

$$\omega_{th} = 162.1952\, rad.s^{-1}$$

$$\xi_{th} = 0.0809$$

The results show that the results obtained from the ANN model are very close to the theoretical parameters. This confirms the ability of the considered ANN training off-line by the BP algorithm to identify the 2MM system parameters.

V. CONCLUSIONS

In this paper, a feed-forward neural network is designed to simulate a 2MM system. The used algorithm, based on the backpropagation method, is formulated and implemented to optimize the learning process of the network. The frequency analysis was used to identify the 2MM system parameters. The simulation results obtained show the effectiveness of the neural network model.

REFERENCES

[1] Y. Yang, J. Cao, D. Zhu, A neural network methodology of quadratic optimization with quadratic equality constraints, in: Advances in Neural Networks– ISNN 2005, Springer, Chongqing, 2005, pp. 706–711.

[2] Chunlin Sha, Hongyong Zhao n , Fengli Ren C. Sha et al, A new delayed projection neural network for solving quadratic programming problems with equality and inequality constraints, Neurocomputing 168, 2015, pp. 1164–1172

[3] Nguyen Ngoc Son, Ho Pham Huy Anh ,Adaptive Displacement Online Control of Shape Memory Alloys Actuator based on Neural Networks and Hybrid Differential Evolution Algorithm ,23 March 2015.

[4] Y. Xia, H. Leung, E. Bossé, Neural data fusion algorithms based on a linearly constrained least square method, IEEE Trans. Neural Netw. 13 (2) (2002) 320–329

[5] Y. Xia, J. Wang, A recurrent neural network for solving linear projection equations, Neural Netw. 13 (3) (2000) 337–350

[6] K.J. Hunt, B. Sbarbaro, R. Zbikowski, P.J. Gawthrop, Neural networks for control system: a survey, Automatica 28 (6) (1992) 1083–1112.

[7] A.M. Schaefer, U. Steffen, G.Z. Hans, A recurrent control neural network for data efficient reinforcement learning, in: IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning, 2007, pp. 151–157

[8] Y. Yang, J. Cao, A delayed neural network method for solving convex optimization problems, Int. J. Neural Syst. 16 (4) (2006) 295–303

[9] S. Effati, A. Nazemi, Neural network models and its application for solving linear and quadratic programming problems, Appl. Math. Comput. 172 (1) (2006) 305–331.

[10] D. Tank, J.J. Hopfield, Simple'neural'optimization networks: an a/d converter, signal decision circuit, and a linear programming circuit, IEEE Trans. Circuits Syst. 33 (5) (1986) 533–541

[11] M.P. Kennedy, L.O. Chua, Neural networks for nonlinear programming, IEEE Trans. Circuits Syst. 35 (5) (1988) 554–562

[12] Kuschewski, John G. Hui S. and Zak S.H. “Application of feedforward neural networks to dynamical system identification and control”, IEEE Transactions on Control Systems Technology, 1993, vol. 1, pp 37-49.

[13] Narendra, K.S., and Parthaasarathy, K., "Identification and Control of Dynamical Systems Using Neural Networks," IEEE Trans Neural Networks, 1990, vol.1, pp 4-27.

[14] K.Nouri et R.DHaouadi, "Structure de Commande Neuronale d'un système à deux masses ", JTEA'98, tome I, pp.296-302.

[15] H. Loussifi, K. Nouri, N. Benhadj Braiek.” A New Efficient Hybrid Intelligent Method for Nonlinear Dynamical Systems Identification: The Wavelet Kernel Fuzzy Neural Network” Communications in Nonlinear Science and Numerical Simulation, Volume 32, March 2016, Pages 10–30 IF:

[16] K. Nouri, H. Loussifi, N. Benhadj Braiek. " A Comparative Study on the Identification of a Dynamical Model of Multi-mass Electrical Drives using Wavelet Transform". International Journal of Systems Science, IJSS(2014), Vol.45. Pages 1- 19