Neural networks construction application to the kinematical analysis of the five-point suspension

GAMIL A. AZIM[1*], HAZEM A. ATTIA[2]

In this paper, a General Neural Networks Regression (GNNR) is presented. The position analysis of multi-link five-point suspension system is solved using GNNR. The mechanism of wheel suspension is a multi-body system (MBS) which is a system of bodies (in this case rigid links of given length) whose mutual position is bounded by geometrical constraints (joints) and by active kinematical driving constraints in the form of prescribed motion related to the degrees-of-freedom (DOF) of the given mechanism. GNNR is first trained with coordinates of the defined points and input driving variables. After training, performance is measured by having the network generate the coordinates of the defined points in terms of driving input variables. It is found that GNNR provides a simple and effective way to model the spatial kinematics of the five point's suspension and eliminate the convergence problems associated with algorithmic solution methods.

K e y w o r d s: General Neural Networks Regression, kinematical analysis

1. Introduction

Artificial Neural Networks (ANNs) are massively parallel, highly connected structures consisting of a number of simple, nonlinear processing units. Due to their massively parallel structure, they can perform computation at very high rate if implemented on a dedicated hardware. Moreover of their adaptive nature, they can learn the characteristics of input signals and adapt to changes in the data. The nonlinear nature of ANNs can help in performing function approximation and signal filtering operations which are beyond optimal linear techniques. Feed Forward Neural Networks (FFNNs) are the basic types of neural networks capable of approximating generic classes of functions, including continuous and integrable ones.

[1] Qassim University, College of Computer, P. O. Box 103, Buraydah 81999, Saudi Arabia
[2] Faculty of Engineering, El-Fayoum University, El-Fayoum, Egypt
[*] Corresponding author, e-mail address: gazim3@hotmail.com

An important class of Feed Forward Neural Networks is Multi Layers Perceptrons Neural Networks (MLPNNs). The MLPNNs have features (data learning) such as the ability to learn and generalize smaller training set requirements, fast operation, and easy implementation. MLPNN consists of various layers: an input layer, an output layer and one or more hidden layers. The three-layer network with sigmoid activation function is capable to produce a decision function with enough hidden units. The input layer contains the input nodes which interact with the outside environment. The input units are only buffer units that pass the signal without changing it. The hidden layer size is left to the appreciation of the user that will estimate the number of hidden nodes by his experience or by trial and error. This number must not be too large to avoid waste of computations and slow convergence process and not too low to make the network capable to absorb the set of training. The output layer represents the number of nodes which are equal to the number of classes, each output node representing a class [1–3].

2. Radial Basis Functions

Radial Basis Function (RBF) is a type of neural network employing a hidden layer of radial units and an output layer of linear units, and is characterized by reasonably fast training and reasonably compact networks. Radial functions are simply a class of functions in principle, they could be employed in any sort of networks (single-layer or multi-layer). However RBF networks have traditionally been associated with radial functions in a single layer, such as is shown in Fig. 1.

The kinematical analysis of a given mechanism and the determination of the displacements, velocities and accelerations of its various members is a classical engineering problem that can be analysed by either graphical or analytical methods. Graphical methods are nowadays not used in practice, due to the availability of

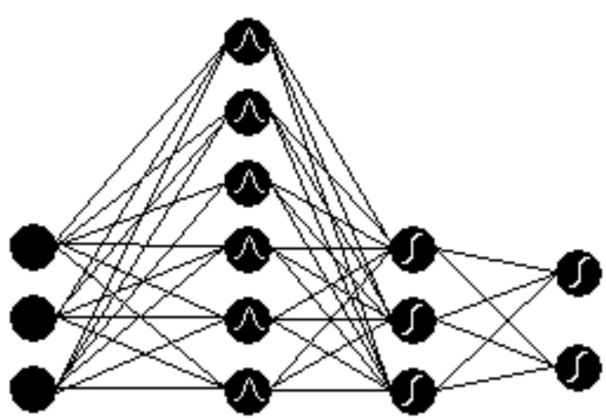


Fig. 1. RBF Neural Network.

Computer-Aided-Engineering (CAE) tools. Graphical methods are introduced only in the education from methodological point of view and they are restricted for planar systems. The analytical methods used can be classified according to the type of coordinates chosen to determine their configuration and specify their constraints. Some formulations use a large set of absolute coordinates [4, 5]. The position and orientation of the rigid links in the mechanism are described with respect to the global reference coordinate system. Other formulations use sets of relative coordinates [5, 6], where the position of each link is defined with respect to the previous link by means of relative joint coordinates that depend on the type of the joint used.

In this paper, a GNNR is presented and applied to solve the position analysis problem of a multi-link five-point suspension system. The position of different links in the suspensions is carried out in terms of the rectangular Cartesian coordinates of some defined points in the links and at the joints. Geometrical constraints are introduced to fix the relative position between the points belonging to the same rigid link. Solution of the geometrical constraints gives the coordinates of the defined points for known driving variables. The GNNR is first trained with the coordinates of the defined points and the input driving variables. After training, performance is measured by using the network to generate the coordinates of the defined points in terms of driving input variables.

3.1 Modelling of the multi-link five-point suspension system

The multi-link five-point suspension system is usually used for rear driven axles of current productions of Mercedes-Benz cars, Mazda 929, some BMW and Toyota Supra cars [7]. The mechanical system consists of the main chassis, a multi--link five-point suspension system, and the wheel as shown in Fig. 2. The system has three degrees of freedom (DOF). Since our goal is the study of the motion of the suspension system only, the chassis, it is constrained to move vertically upward or downward, only one DOF out of its six DOF remains. The wheel has one DOF corresponding to the rolling motion. Since the suspension mechanism connects the driven wheel to the chassis (specifically to the axle carrier) by rubber mountings, it can be simulated as five binary links connecting the chassis and the wheel knuckle through spherical joints at both ends of each link. Thus, the suspension system consists of five links and ten spherical joints, and has only one DOF (see Fig. 3).

3.2 Position analysis

The configuration of the mechanism can be specified by defining a set of points on the links and at the joints. Figure 3 presents the mechanism with the assigned points. Each binary link is replaced by two points located at the centre of the spherical joints at both ends, while the adjacent links are sharing common points.

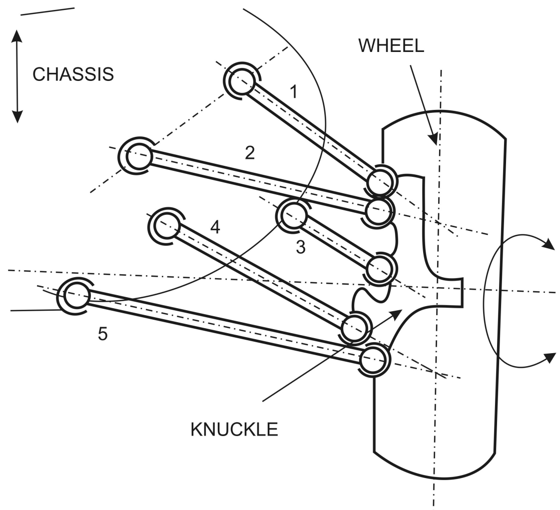


Fig. 2. The multi-link five point suspension system.

The whole mechanism is then replaced by ten points. Points $1, \ldots, 5$, that are located at the chassis, are known points. The Cartesian coordinates of the unknown points $6, \ldots, 10$, located on the knuckle, define the motion variables. Therefore, 15 constraint equations are needed to be solved for the 15 unknown Cartesian coordinates. The initial positions of points $1, \ldots, 5$ are known from the driver data.

The constraints are either geometrical or kinematical ones. Geometrical constraints are distance constraints that fix the relative positions of the points on a rigid link of the mechanism [8]. The geometrical constraint equations are expressed

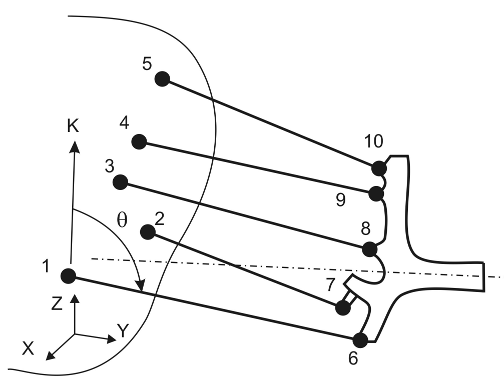


Fig. 3. The multi-link five point suspension system with the assigned points.

in the Cartesian coordinates of the points as follows:

$$(x_6 - x_1)^2 + (y_6 - y_1)^2 + (z_6 - z_1)^2 - d_{6,1}^2 = 0, \tag{1.1}$$

$$(x_7 - x_2)^2 + (y_7 - y_2)^2 + (z_7 - z_2)^2 - d_{7,2}^2 = 0, \tag{1.2}$$

$$(x_8 - x_3)^2 + (y_8 - y_3)^2 + (z_8 - z_3)^2 - d_{8,3}^2 = 0, \tag{1.3}$$

$$(x_9 - x_4)^2 + (y_9 - y_4)^2 + (z_9 - z_4)^2 - d_{9,4}^2 = 0, \tag{1.4}$$

$$(x_{10} - x_5)^2 + (y_{10} - y_5)^2 + (z_{10} - z_5)^2 - d_{10,5}^2 = 0, \tag{1.5}$$

$$(x_7 - x_6)^2 + (y_7 - y_6)^2 + (z_7 - z_6)^2 - d_{7,6}^2 = 0, \tag{1.6}$$

$$(x_8 - x_6)^2 + (y_8 - y_6)^2 + (z_8 - z_6)^2 - d_{8,6}^2 = 0, \tag{1.7}$$

$$(x_9 - x_6)^2 + (y_9 - y_6)^2 + (z_9 - z_6)^2 - d_{9,6}^2 = 0, \tag{1.8}$$

$$(x_{10} - x_6)^2 + (y_{10} - y_6)^2 + (z_{10} - z_6)^2 - d_{10,6}^2 = 0, \tag{1.9}$$

$$(x_8 - x_7)^2 + (y_8 - y_7)^2 + (z_8 - z_7)^2 - d_{8,7}^2 = 0, \tag{1.10}$$

$$(x_9 - x_7)^2 + (y_9 - y_7)^2 + (z_9 - z_7)^2 - d_{9,7}^2 = 0, \tag{1.11}$$

$$(x_{10} - x_7)^2 + (y_{10} - y_7)^2 + (z_{10} - z_7)^2 - d_{10,7}^2 = 0, \tag{1.12}$$

$$(x_9 - x_8)^2 + (y_9 - y_8)^2 + (z_9 - z_8)^2 - d_{9,8}^2 = 0, \tag{1.13}$$

$$(x_{10} - x_8)^2 + (y_{10} - y_8)^2 + (z_{10} - z_8)^2 - d_{10,8}^2 = 0, \tag{1.14}$$

where $d_{i,j}$ is the distance between points i and j belonging to the same rigid link, and x_i, y_i and z_i are the Cartesian coordinates of point i. Kinematical constraints result from the conditions imposed by the kinematical joints on the relative motion between the bodies they comprise. Points located at the centre of a spherical joint or at the axis of a revolute joint automatically eliminate all the kinematical constraints due to these joints. Moreover, driving constraints are added to the above constraints as functions of the input driving angular position (see Fig. 3) in the form [11]

$$(z_6 - z_1) - d_{6,1} \cos\theta = 0. \tag{1.15}$$

Equation (1) expresses the required 15 independent constraint equations in terms of the Cartesian coordinates of the assigned points. Given the set of the known coordinates of points $1, \ldots, 5$ and the driving variable θ at each instant, the non-linear Eq. (1) can be solved by any iterative numerical method [9] to determine the 15 unknown Cartesian coordinates of points $6, \ldots, 10$.

4. Neural implementation and parameters

GRNN is a kind of radial basis network that is often used for function approximation. We use Matlab NEWGRNN function to design a generalized regression neural network, which consists of two-layer network. The first layer has radial basis transfer function (see Fig. 4), calculates weighted inputs with Euclidean distance weight function and net input with product net input function. The second layer has hard limit transfer function neurons. We implemented GRNN as follows:

$$\text{net} = \text{newgrnn}(\mathbf{P}, \mathbf{T}, S).$$

$\mathbf{P}$ is $R \times Q$ matrix of Q input vectors and $\mathbf{T}$ is $S \times Q$ matrix of Q target class vectors. S is spread of radial basis functions.

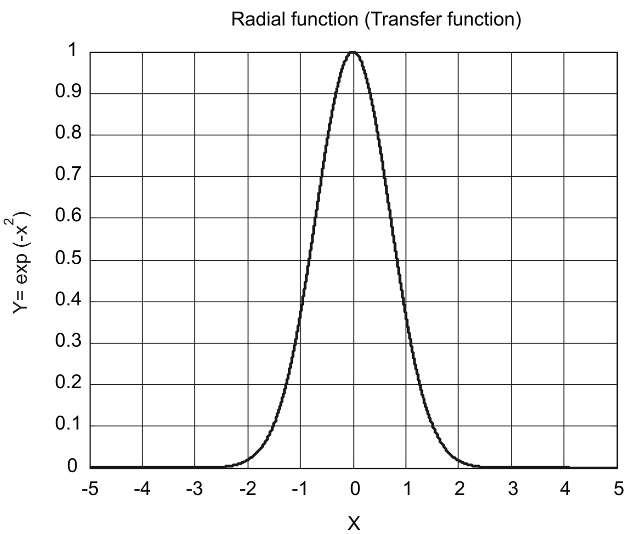


Fig. 4. Radial function is a transfer function, $y = \exp(-x^2)$.

4.1 Training and testing sets

The data were divided in two sets, one for training and one for testing. The training set for K.F.-points suspension is represented by $((t_i, \Theta_i), (p_1, p_2, p_3, p_4, p_5)_i)$ for $i = 1$ to 36, and the testing set for $i = 37, \ldots, 82$ with $p_m = (x_m, y_m, z_m)$ in the points. We constructed networks of two inputs, which are represented by (t, Θ), and fifteen outputs correspond to the coordinates of the five points. The weights were initialized randomly. The learning rates were chosen as 0.09. Table 1 compares the coordinates of the 5 points obtained using GNNR with those evaluated using classical numerical method. The results ensure the validity and accuracy of the suggested GNNR method (see in Figs. 5–8).

Table 1. Comparison between GNNR and a classical numerical solution (N. S.)

GNNR $\Theta = 0.71$	N. S. $\Theta = 0.71$
−1.2590	−1.2051
0.3080	0.2009
0.1534	0.2517
−1.2090	−1.1220
0.2172	0.0163
0.3225	0.3226
−1.1892	−1.1129
0.1672	0.0075
0.2742	0.2452
−1.2523	−1.2122
0.2418	0.1872
0.0906	0.1535
−1.1089	−1.0363
0.2766	0.1199
0.2516	0.2841

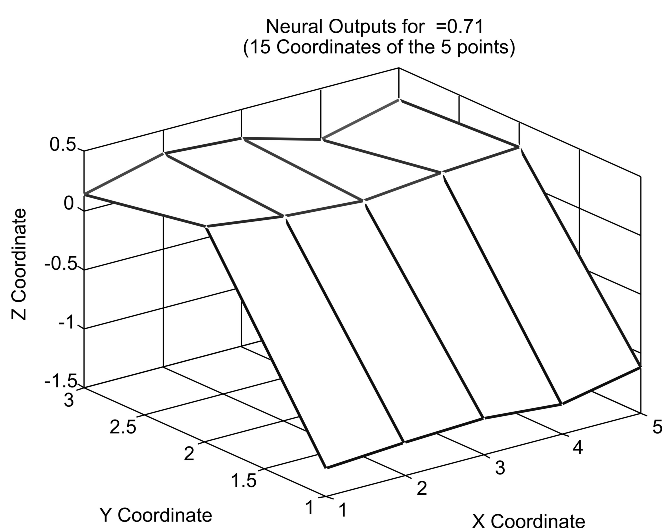


Fig. 5. The neural outputs for $\Theta = 0.71$ (15 coordinates of the 5 points).

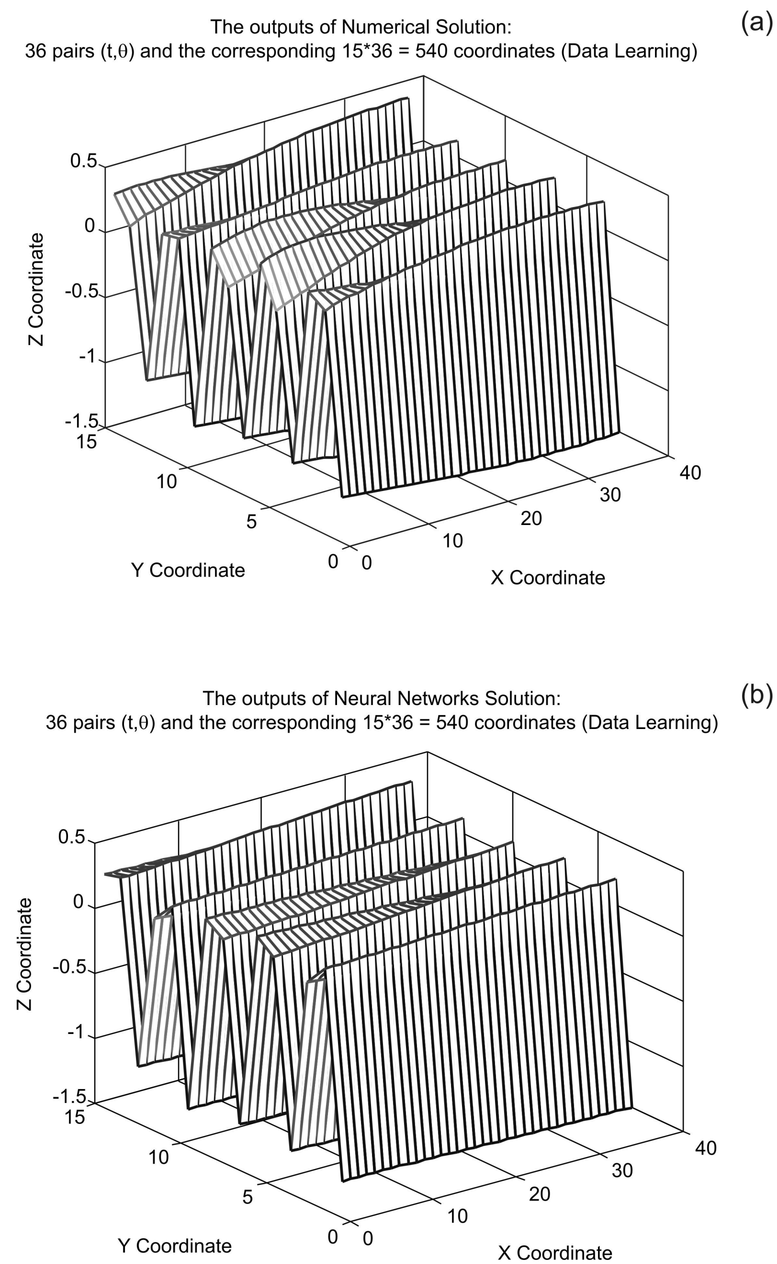
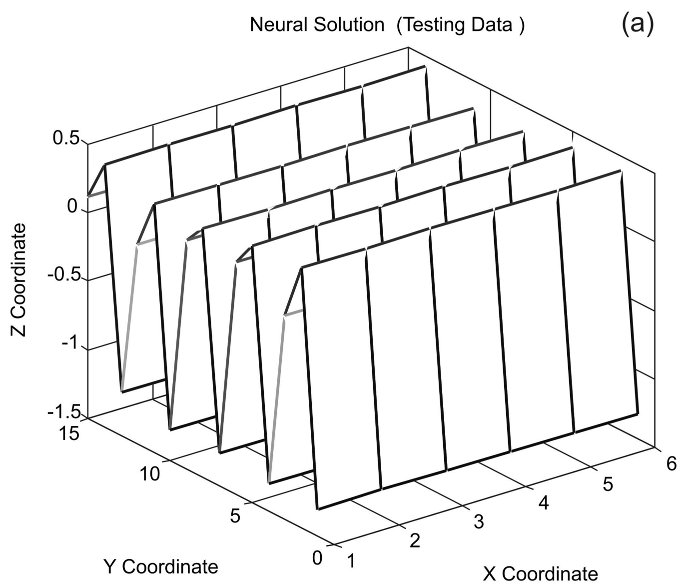


Fig. 6. The outputs of numerical solution (a) and neural solution (b): 36 pairs (t, θ) and the corresponding $15 * 36 = 540$ coordinates.



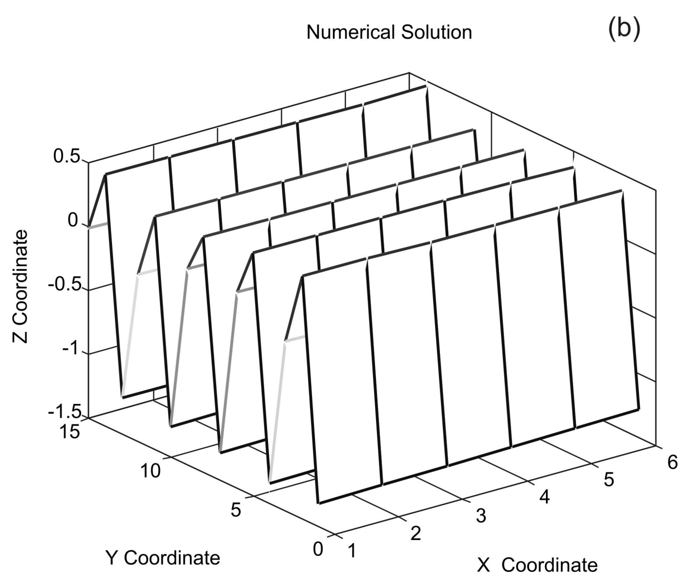


Fig. 7. The outputs of neural solution (a) and numerical solution (b): 6 pairs (t, θ) and the corresponding $15 * 6 = 90$ coordinates.

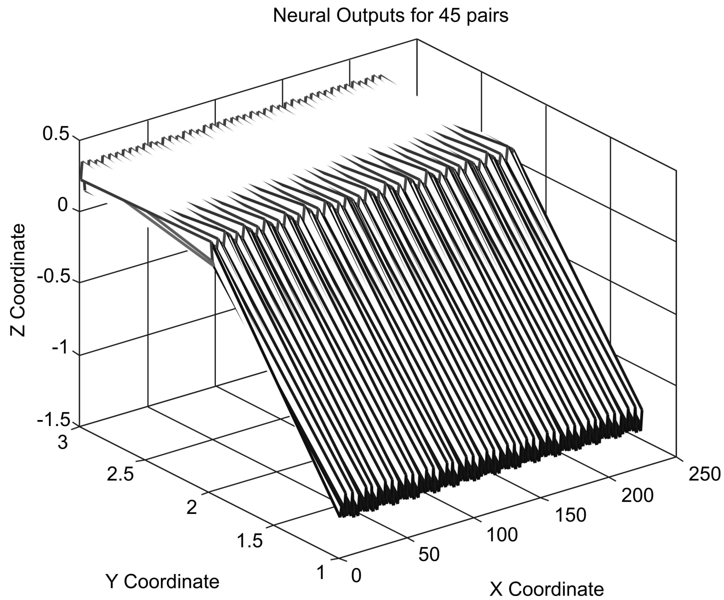


Fig. 8. The neural outputs for 45 pairs (t, Θ), (15 coordinates of the 5 points).

5. Discussion and conclusions

The kinematical analysis of spatial mechanisms is classically done by numerical methods which can be classified according to the type of coordinates chosen. The algebraic constraint equations are introduced to represent the kinematical joints that connect the rigid bodies and are expressed in terms of the system of coordinates chosen. These constraint equations are algebraic non-linear equations whose non--linearity depends on the complexity of the mechanism and the coordinates used. The numerical solution of this system of non-linear equations, known as the finite displacement problem, has many essential problems. The most important problem is the convergence problem which depends generally on the initial guess of the unknown variables and the nature of nonlinearity. It is known that in the case of complex spatial mechanisms, if the initial guess is not properly chosen, it is very difficult to reach convergence. Even in the case of convergence, the complexity of the mechanism affects greatly the number of iterations and consequently, the computations needed, which determines the efficiency. The lengthy computations affect to a great extent the accuracy as well as computational time.

Utilizing the neural networks solves entirely the convergence problem and gets rid of its disadvantages which guarantees the possibility of solving the nonlinear problem without large time consuming.

In this paper, a General Neural Networks Regression (GNNR) is presented. The position analysis of multi-link five-point suspension system is solved using

GNNR. The position of the different links in the suspensions is carried out in terms of the rectangular Cartesian coordinates of some defined points in the links and at the joints. GNNR is first trained with coordinates of the defined points and input driving variables. After training, performance is measured by having the network to generate the coordinates of the defined points in terms of driving input variables. It is found that GNNR provides a simple and effective way to model the spatial kinematics of the five point's suspension and eliminates the convergence problems associated with algorithmic solution methods.

REFERENCES

[1] ZURADA, J. M.: Introduction to Artificial Neural Systems. Bombay, India, Jaico Pub. House 1997.
[2] FAUSELT, L.: Fundamentals of Neural Networks. Englewood Cliffs, NJ, Prentice--Hall 1994.
[3] HARVEY, R. L.: Neural Network Principals. Englewood Cliffs, NJ, Prentice-Hall 1994.
[4] WEHAGE, R. A.—HAUG, E. J.: ASME Journal of Applied Mechanical Design, *104*, 1982, p. 247.
[5] NIKRAVESH, P. E.: Computer-Aided Analysis of Mechanical Systems. NJ, Prentice--Hall 1988.
[6] DENAVIT, J.—HARTENBERG, R. S.: ASME Journal of Applied Mechanics, *22*, 1955, p. 215.
[7] PAUL, B.—KRAJCINOVIC, D.: ASME Journal of Applied Mechanics, *37*, 1970, p. 697.
[8] ADLER, U.: Automotive Handbook. Bosch, FDI-Verlag, Approved Edition Under Licence SAE, ISBN 089283-518-6.
[9] MOLIAN, S.: Journal of Mechanical Engineering Science, *10*, 1968, p. 360.
[10] WASSERMAN, P. D.: Advanced Methods in Neural Computing. New York, Van Nostrand Reinhold 1993.
[11] HAZEM ALI ATTIA: KSME International Journal, *17*, 2003, p. 1133.

Received: 26.6.2006
Revised: 11.12.2007