Design of Neural Networks Model for Transmission Angle of a Modified Mechanism

Şahin Yıldırım*, Selçuk Erkaya, Şükrü Su, İbrahim Uzmay
Erciyes University, Engineering Faculty,
Department of Mechanical Engineering Kayseri/TURKEY

This paper discusses Neural Networks as predictor for analyzing of transmission angle of slider-crank mechanism. There are different types of neural network algorithms obtained by using chain rules. The neural network is a feedforward neural network. On the other hand, the slider-crank mechanism is a modified mechanism by using an additional link between connecting rod and crank pin. Through extensive simulations, these neural network models are shown to be effective for prediction and analyzing of a modified slider-crank mechanism's transmission angle.

Key Words : Transmission Angle, Neural Network, Slider-crank Mechanism, Learning Algorithms

Nomenclature

TDC : Top dead center
BDC : Bottom dead center
μ : Transmission angle
r_c : Crank arm length
l : Connecting rod length
r_p : Radius of pinion gear
e : Distance of eccentricity
θ : Crank rotation angle
φ_i : Weighted sum
$q(.)$: Non-linear dynamic function
ydi : i^{th} desired outputs
yi : i^{th} outputs of the network
Θ : Unknown parameters
w : Network weight
η : Learning rate
α : Momentum constant
δ_i^m : Error signal of the i^{th} neuron in the m^{th} layer
b_i^{m-1} : Bias input to neuron i in layer $m-1$.
n_i : Number of neurons in the input layer
n_j : Number of neurons in the hidden layer
n_k : Number of neurons in the output layer
N : Training numbers
RMSE : Root Mean Square Error

* Corresponding Author,
E-mail : sahiny@erciyes.edu.tr
Erciyes University, Engineering Faculty, Department of Mechanical Engineering Kayseri/TURKEY. (Manuscript **Received** December 8, 2004; **Revised** August 18, 2005)

1. Introduction

There has been considerable interest in the past few years in exploring the applications of artificial neural networks (ANNs) for analyzing and prediction mechanisms. Also, modeling applications of neural networks received increasing attention due to their versatility, such as non-linear mapping, linear adaptability and parallel processing.

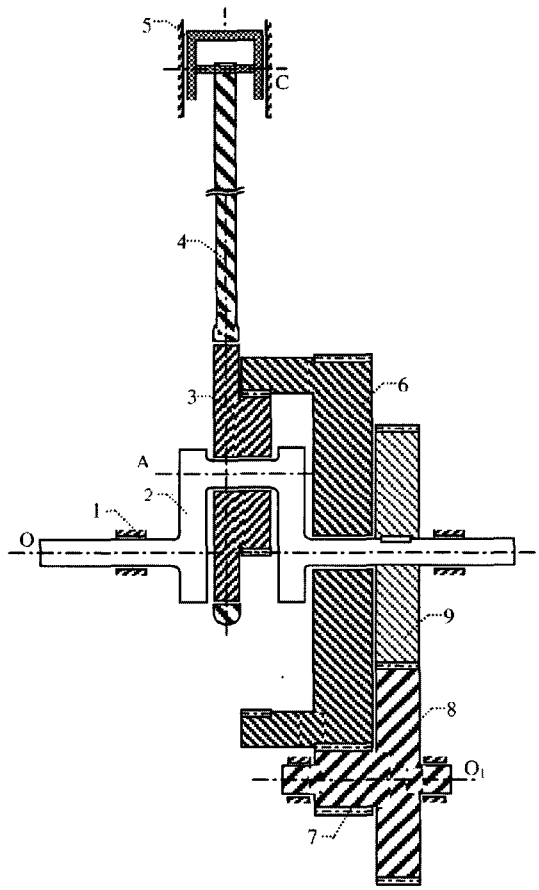
A neural network has been employed as a case-based approach for analyzing dimensions of a planar linkage (Vasiliu and Yannou, 2001). In addition to neural network approach, various classical method schemes have been studied to achieve good tracking performance of the slider-crank mechanism. Soylemez has analyzed slider-crank mechanism using complex algebra method (Söylemez, 2002). The complex algebra has employed to solve classical problem. The solution was obtained as the root of cubic equation within

a defined range. Shrinivas and Satish have proposed importance of the transmission angle for most effective force transmission. In their paper, 4, 5, 6 and 7 bar linkages of the mechanisms were investigated (Shrinivas and Satish, 2002). Kim and Jung have investigated the theoretical mechanism for driving the tapered pistons by using of the geometric method (Kim and Jung, 2003). In their research, the driving area of the tapered pistons was analyzed by measuring the strain of a cylinder forced against a tapered piston using an electric strain gauge and a slip ring. The forces applied to tapered pistons were also investigated with the change of discharge pressure and the rotational speed. Attia has analyzed a numerical algorithm for kinematic analysis of a multi-link five-point suspension mechanism (Attia, 2003). In his paper, geometric constraints for the system were introduced to fix the relative positions between the points belonging to the same rigid body. Position, velocity and acceleration analyses were carried out and the presented results were discussed. Choi has also investigated kinematic analysis and optimal design of a 3-PPR planar parallel manipulator, which consisted of three active prismatic joints, three passive prismatic joints, and three passive rotational joints (Choi, 2003). In his research, for the kinematic analysis, direct and inverse kinematics, and inverse Jacobian of the manipulator were derived. Also, for the optimal design of the manipulator, an optimal design procedure was carried out using Min-Max theory. Lee et al. have investigated a constraint operator for the kinematic calibration of a parallel mechanism (Lee et al., 2003). They constrained the movement between two poses by adopting the concept of a constraint operator. Also, a cost function was derived by the errors between the theoretical movement and the actual movement. Finally, the parameters that minimize the cost function were estimated and substituted into the kinematic model for a kinematic calibration.

The paper is organized in the following manner. Section 2 describes the theory of transmission angle of the modified slider-crank mechanism. Some details of the neural networks and learning algorithms are outlined in section 3. Simulation results are given in section 4 and the paper is concluded within section 5.

2. Modified Slider-crank Mechanism

Modified slider-crank mechanism, as shown in Fig. 1, has an additional extra link between



O : Rotation center of the crank shaft
A : Crank-pin center
C : Piston-pin center
1 : Crank shaft bearing
2 : Crank arm
3 : Eccentric connector
4 : Connecting rod
5 : Piston
6-7-8-9 : Elements of the epicyclic gear mechanism
1st transmission line : 4-3-2
2nd transmission line : 4-3-6-7-8-9-2

Fig. 1 Schematic representation of the modified slider-crank mechanism

connecting rod and crank pin as distinct from well-known slider-crank mechanism. The new extra link may be called eccentric connector and transmits gas forces to the crank and also drives a planetary gear mechanism. In order to drive planetary gear train, a pinion fixed to the eccentric connector in a parallel plane is used. So, there are two transmission lines in this new system. One of them called direct transmission line consists of connecting rod-eccentric connector-crank arm and the other called indirect transmission line consists of connecting rod-eccentric connector-gear mechanism.

When the motion characteristic of the mechanism in Fig. 1 is outlined carefully, a kinematic-based scheme in Fig. 2 is obtained.

Referring to Fig. 2, it can be seen that the modified mechanism has one degree of freedom, that is, this model is a constrained mechanism. The eccentric connector has a curvilinear translation because of the particular choice of gear ratio

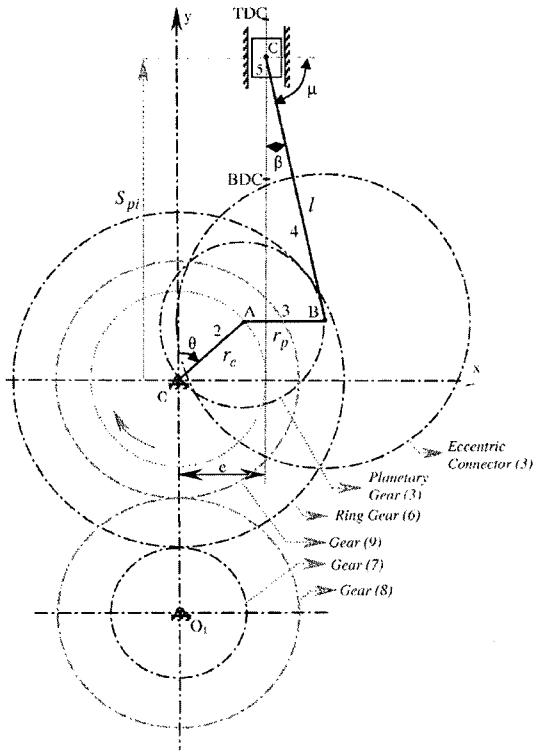


Fig. 2 Working scheme of the modified mechanism on XY plane

Table 1 Kinematic parameters of the modified mechanism

Parameters	Values (mm)
r_c	50
l	269
r_p	50
e	35~65

between pinion and ring gear. So, it has no a relative motion with respect to crank pin. The kinematic parameters and values of the modified mechanism are given in Table 1.

2.1 Transmission angle of mechanism

The transmission angle (μ) is an important criterion for design of the mechanisms, which denotes the quality of motion transmission in a mechanism. It helps to decide the "Best" among a family of possible mechanisms for most effective force transmission (Shrinivas and Satish, 2002). Usually, transmission angle is used to obtain better results for various linkage applications. The mechanism designed with cosine of maximum transmission angle criterion will have minimum force acting along the coupler and on the bearings. Although a good transmission angle is not a cure-all for every design problem, for many mechanical applications it can guarantee for the performance of linkage at higher speed without unfavorable vibrations. When μ=90°, most effective force transmission takes place and the accuracy of output motion is less sensitive to manufacturing tolerances of link lengths and clearance between joints and change of dimensions due to thermal expansion. Mechanisms having transmission angle too much deviated from 90°, exhibit poor operational characteristics like noise and jerk at high speeds. If μ=0°, self-locking takes place. Transmission angel in a mechanism provides a very good of the quality of motion, expected noise output and its costs in general. In other words, it is a simple and useful coefficient of performance in mechanisms for non-uniform motion transmission.

The transmission angle does not consider the dynamic forces due to velocity and acceleration.

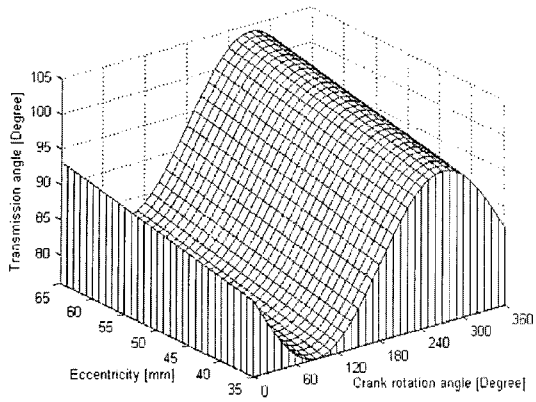


Fig. 3 The transmission angle variations of the modified system

Because of this reason, it is widely used in kinematic synthesis stage during which the lengths and mass properties of the links are unknown. Kinematically expressed transmission angle does not reflect the action of gravity or dynamic forces. So, for the determination of transmission characteristics of the linkage, it is not necessary to analyze the forces and torque acting at each joint of the whole mechanism. Referring to Fig. 2, the transmission angle equation for the modified mechanism can be described as ;

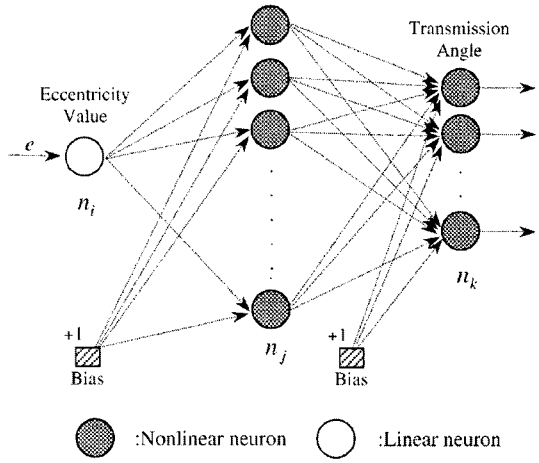
$$l\cos\mu = r_c\sin\theta + r_p - e \tag{1}$$

$$\mu = \cos^{-1}\left(\frac{r_c\sin\theta + r_p - e}{l}\right) \tag{2}$$

From Eq. (2), the transmission angle depends on mainly two variables : eccentricity value and crank rotation angle, that is, $\mu = f(\theta_i,\ e_i)$. The other parameters ; r_c, r_p and l are constant terms. The relationship between eccentricity value and transmission angle during one cycle is outlined in Fig. 3.

3. Feedforward Neural Network

The architecture of the neural network is shown in Figure 4. The structure is the same as a feed-forward ANN except for the learning algorithms. As it can be seen from Figure 4, neural network consists of three layers, which are input, hidden and output layers. The input layer with



e : Eccentricity value of mechanism
n_i : Number of neurons in the input layer (i=1)
n_j : Number of neurons in the hidden layer (j=1, ···, 10)
n_k : Number of neurons in the output layer (k=1, ···, 37)

Fig. 4 A Feedforward NN for kinematic analysis of modified mechanism

one neuron, the hidden layer with 10 neurons and the output layer with 37 neurons are employed as a predictor of the mechanism.

The network architecture is defined by the basic processing elements and the way in which they are interconnected.

The basic processing element of the connectionist architecture is often called a neuron by analogy with neurophysiology. Many of the basic processing elements may be considered to have three components ;

(1) A weighted sumer
(2) A linear dynamic single input single output (SISO)
(3) A non-linear dynamic function

These elements are considered in turn in the following section. The weighted sumer is described by

$$\varphi_i(t) = \sum_{j=1}^{N} a_{ij}y_i(t) = \sum_{k=1}^{M} b_{ik}u_k(t) + w_i \tag{3}$$

giving a weighted sum φ_i in terms of the outputs of all elements y_i, external inputs u_k and corresponding weights a_{ij} and b_{ik} together with constants w_i. N of these weighted sumer elements

can be conveniently expressed in vector-matrix notation.

Stacking N weighted sums φ_i into a column vector φ, the N outputs y_i into a vector y and M inputs u_k into a vector u and the N constants w_i into a vector w, equation (3) may be rewritten in vector matrix form as:

$$\varphi(t)=Ay(t)+Bu(t)+w \tag{4}$$

where the ij^{th} element of the $N\times N$ matrix A is and the element of the $N\times M$ matrix B is b_{ik}. The constants w_i could be incorporated with the inputs u_k, but it is useful to represent them explicitly.

The linear dynamic SISO system has input φ and output x_i. In transfer function form, it is described by

$$x_i(s)=H(s)\,\varphi_i(s) \tag{5}$$

In the time domain; above equation becomes:

$$x_i(t)=\int_{-\infty}^{t} h(t-t')\,\varphi_i(t')\,dt' \tag{6}$$

where $H(s)$ and $h(t)$ form are a Laplace transform pair.

The non-linear dynamic function $g(.)$ gives the element output y_i in terms of the transfer function output x_i:

$$y_i=g(x_i) \tag{7}$$

3.1 Learning algorithms

The learning algorithm topology, which was employed for the neural network updating the weight can be described as follows; define the error function as (Canbulut et al., 2004)

$$J=\frac{1}{2}\sum_{i=1}^{n_O}(y_{di}(t)-y_i(t))^2 \tag{8}$$

where $ydi(t)$ are the i^{th} desired outputs and $yi(t)$ are the i^{th} outputs of the network. This error function is to be minimized with respect to all the unknown parameters Φ. In the steepest descent approach the parameter vector $\Phi=[\theta_1, \theta_2, \cdots, \theta_n]^{\mathrm{T}}$ is adjusted using the increment vector $[\Delta\theta_1, \Delta\theta_2, \cdots, \Delta\theta_n]^{\mathrm{T}}$ defined along the negative gradient direction of J

$$\Delta\theta_i=-\eta\frac{\partial J}{\partial\theta_i} \tag{9}$$

Although the one-hidden layer model is used in the present application, it is useful to derive the gradient of J for the general case, and the result for the one-hidden-layer model can readily be obtained as a special case.

Starting from the output layer m of the network and setting $\theta_i=W_{ij}^m$, the application of the chain rule gives rise to

$$\frac{\partial J}{\partial W_{ij}^m}=\frac{\partial J}{\partial y_i}\frac{\partial y_i}{\partial W_{ij}^m} \tag{10}$$

From equation (8)

$$\frac{\partial J}{\partial y_i}=-(y_{di}-y_i)=-\delta_i^m \tag{11}$$

where δ_i^m is called the error signal of the i^{th} neuron in the m^{th} layer. From Equation (10)

$$\frac{\partial y_i}{\partial W_{ij}^m}=\mathbf{x}_j^{m-1} \tag{12}$$

Thus,

$$\frac{\partial J}{\partial W_{ij}^m}=-\delta_i^m x_j^{m-1} \tag{13}$$

Next consider the $(m-1)^{\text{th}}$ layer. Using the chain rule yields:

$$\frac{\partial J}{\partial W_{ij}^{m-1}}=\sum_{k=1}^{n_O}\frac{\partial J}{\partial y_k}\times\frac{\partial y_k}{\partial x_j^{m-1}}\times\frac{\partial x_i^{m-1}}{\partial z_i^{m-1}}\times\frac{\partial z_i^{m-1}}{\partial W_{ij}^{m-1}} \tag{14}$$

Then

$$\frac{Sx_i^{m-1}}{\partial z_i^{m-1}}=g'(z_i^{m-1}) \tag{15}$$

and

$$\frac{z_i^{m-1}\partial}{\partial W_{ij}^{m-1}}=x_i^{m-2} \tag{16}$$

$$g'(z)=\frac{\partial g(z)}{\partial z} \tag{17}$$

and $g(z_i)$ is the activation of neuron i.

By defining the error signal for the i^{th} neuron of the $(m-1)^{\text{th}}$ layer as:

$$\delta_i^{m-1}=g'(z_i^{m-1})\sum_{k=1}^{n_O}\delta_k^m W_{ki}^m \tag{18}$$

Equation (8) can be rewritten as:

$$\frac{\partial J}{\partial W_{ij}^{m-1}}=-\delta_i^{m-1}x_j^{m-2} \tag{19}$$

Similarly it can be shown that

$$\frac{\partial J}{\partial b_i^{m-1}} = -\delta_i^{m-1} \tag{20}$$

where b_i^{m-1} is the bias input to neuron i in layer $m-1$.

By carrying on this procedure, Equations (18)-(20) can be used as a general algorithm for updating weights in other layers.

Equations (18)-(20) indicate how the error signals propagate backwards from the output layer of the network through the hidden layer to the input layer, hence the name "BP". The steepest-descent minimization of the error function defined in Equation (8) produces the following increments for updating Θ:

$$\Delta W_{ij}^m(t)\,\eta_w \delta_i^m(t)\, x_j^{m-1}(t) \tag{21}$$

$$\Delta b_i^m(t) = \eta_b \delta_i^m(t) \tag{22}$$

where in the output layer

$$\delta_i^m(t) = y_{di}(t) - y_i(t) \tag{23}$$

and in other layers

$$\delta_i^m(t) = g'(z_i^m(t)) \sum_j \delta_j^{m+1}(t)\, W_{ji}^{m+1}(t-1) \tag{24}$$

The constants η_w $(0<\eta_w<1)$ and η_b $(0<\eta_b<1)$ represent the learning rates for the weights and biases respectively. In practice, a large value of the learning rate would be preferable, because this would result in rapid learning. Unfortunately, a large value of the learning rate can also lead to oscillation or even divergence. To help speed up learning but avoid undue oscillations, a momentum term is usually included so that Equations (21) and (22) become

$$\Delta W_{ij}^m = \eta_w \delta_i^m(t)\, x_j^{m-1}(t) + \alpha_w \Delta W_{ij}^m(t-1) \tag{25}$$

$$\Delta b_i^m(t) = \eta_b \delta_i^m(t) + \alpha_b \Delta b_i^m(t-1) \tag{26}$$

where α_w and α_b are momentum constants, which determine the effect of past changes of $W_{ij}^m(t)$ and $\Delta b_i^m(t)$ on the current updating direction in the weight and the bias space respectively. This effectively filters out high frequency variations in the error surface. To summarize, the BP algorithm updates the weights and thresholds of the networks according to

$$W_{ij}^m(t) = W_{ij}^m(t-1) + \Delta W_{ij}^m(t) \tag{27}$$

and

$$b_i^m(t) = b_i^m(t-1) + \Delta b_i^m(t) \tag{28}$$

where the increments $\Delta W_{ij}^m(t)$ and $\Delta b_i^m(t)$ are given in equations in (25) and (26).

The neural network was trained and tested by using five types of learning algorithms. The algorithms can be described in the following forms.

3.1.1 Online-backpropagation algorithm (Case 1)

Online Backpropagation which updates the weights after each pattern is presented to the network. Back-propagation is the most commonly used training algorithm for neural networks. The weights are updated as follows

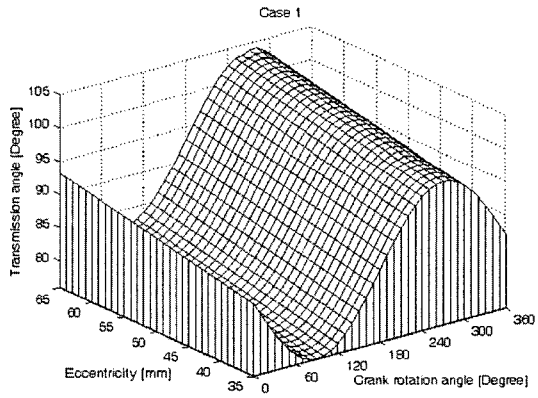


Fig. 5(a) The actual transmission angle variations using online BP learning algorithm

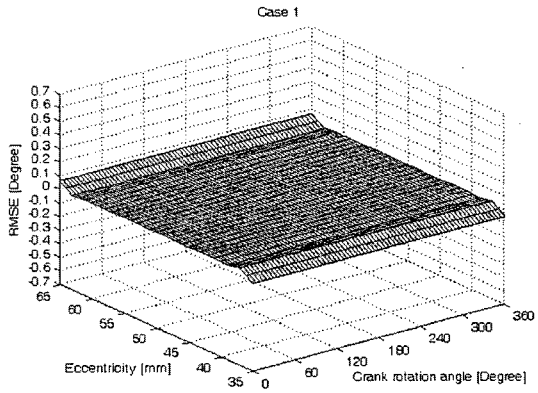


Fig. 5(b) The RMSEs using online BP learning algorithm

$$\Delta w_{ij}(t) = -\eta \frac{\partial E(t)}{\partial w_{ij}(t)} + \alpha \Delta w_{ij}(t-1) \quad (29)$$

where η is the learning rate, and α is the momentum constant. The actual results for "Online Backpropagation Learning Algorithm" can be outlined in Fig. 5(a).

From Fig. 3 and Fig. 5(a), the deviations between desired and actual results (RMS Error) using Online BP Learning Algorithm is shown in Fig. 5(b).

3.1.2 Batch backpropagation algorithm (Case 2)

Batch Backpropagation with weight updates occurring after each epoch. The actual results for "Batch Backpropagation Learning Algorithm" can be outlined in Fig. 6(a).

From Fig. 3 and Fig. 6(a), the deviations between desired and actual results using Batch BP Learning Algorithm is shown in Fig. 6(b).

3.1.3 Delta-Bar-Delta algorithm (Case 3)

Delta-Bar-Delta is an adaptive learning rate method in which every weight has its own learning rate. The learning rates are updated based on the sign of the gradient. If the gradient does not change signs on successive iterations then the step size is increased linearly. If the gradient changes signs, the learning rate is decreased exponentially. In some cases this method seems to learn much faster than non-adaptive methods. Learning rates $\eta(t)$ are updated as follows:

$$\Delta\eta(t) = \begin{cases} K & \text{if } \bar{\delta}(t-1)\,\delta(t) > 0 \\ -\phi\eta(t) & \text{if } \bar{\delta}(t-1)\,\delta(t) < 0 \\ 0 & \textit{else} \end{cases} \quad (30)$$

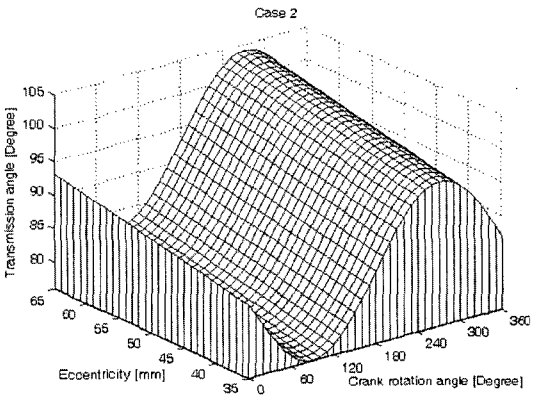


Fig. 6(a) The actual transmission angle variations using batch BP learning algorithm

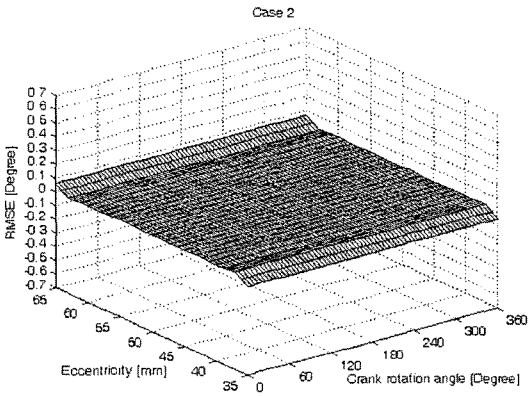


Fig. 6(b) The RMSEs using batch BP learning algorithm

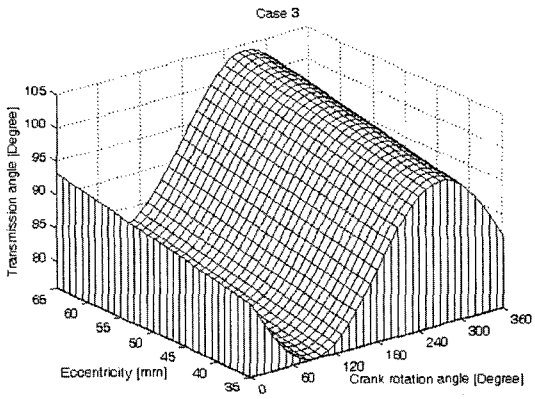


Fig. 7(a) The actual transmission angle variations using delta-bar-delta learning algorithm

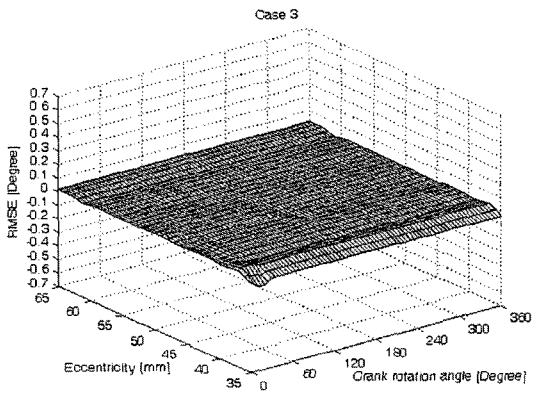


Fig. 7(b) The RMSEs using delta-bar-delta learning algorithm

where $\delta(t)=\frac{\partial E}{\partial w}$ at time t and $\bar{\delta}$ is the exponential average of past values of δ.

$$\bar{\delta}(t)=(1-\theta)\,\delta(t)+\theta\bar{\delta}(t-1) \tag{31}$$

The actual results for "Delta-Bar-Delta Learning Algorithm" can be outlined in Fig. 7(a).

From Fig. 3 and Fig. 7(a), the deviations between desired and actual results using Delta-Bar-Delta Learning Algorithm is shown in Fig. 7 (b)

3.1.4 Random backpropagation algorithm (Case 4)

Random Backpropagation Algorithm stands for "resilient propagation". This is an adaptive learning rate method where weight updates are based only on the sign of the local gradients, not their magnitudes. Each weight, w_{ij}, has its own step size or update value, Δ_{ij} which varies with time t according to :

$$\Delta_{ij}(t)=\begin{cases}\eta^{+}.\Delta_{ij}(t-1), & \text{if } \frac{\partial E}{w_{ij}}(t-1).\frac{\partial E}{w_{ij}}(t)>0\\ \eta^{-}.\Delta_{ij}(t-1), & \text{if } \frac{\partial E}{w_{ij}}(t-1).\frac{\partial E}{w_{ij}}(t)<0\\ \Delta_{ij}(t-1), & \textit{else}\end{cases} \tag{32}$$

where $0<\eta^{-}<1<\eta^{+}$. The weights are updated according to :

$$\Delta w_{ij}(t)=\begin{cases}-\Delta_{ij}(t), & \text{if } \frac{\partial E}{w_{ij}}(t)>0\\ +\Delta_{ij}(t), & \text{if } \frac{\partial E}{w_{ij}}(t)<0\\ 0 & \textit{else}\end{cases} \tag{33}$$

The actual results for "Random Backpropagation Learning Algorithm" can be outlined in Fig. 8 (a).

From Fig. 3 and Fig. 8(a), the deviations between desired and actual results using Random BP Learning Algorithm is shown in Fig. 8(b).

3.1.5 Quick propagation algorithm (Case 5)

Quick Propagation Algorithm is a training method based on the following assumptions :

(1) $E(w)$ for each weight can be approximated by a parabola that opens upward

(2) The change in slope of $F(w)$ for this weight is not affected by all other weights that change at the same time.

The weight update rule is :

$$\Delta w(t)=\frac{S(t)}{S(t-1)-S(t)}\Delta w(t-1)-\eta S(t) \tag{34}$$

where $S(t)=\frac{\partial E}{\partial w}(t)$. The numerator is the derivative of the error with respect to the weight and $[S(t-1)-S(t)]/\Delta w(t-1)$ is a finite difference approximation of the second derivative. Together these approximate Newton's method for minimizing a one-dimensional function $f(x)$: $\Delta x=-f'(x)/f''(x)$. To avoid taking an infinite backward step, or a backward uphill step, a maximum growth factor parameter ψ is introduced.

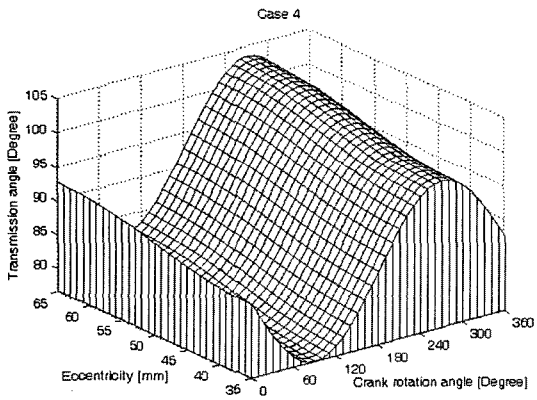


Fig. 8(a) The actual transmission angle variations using random BP learning algorithm

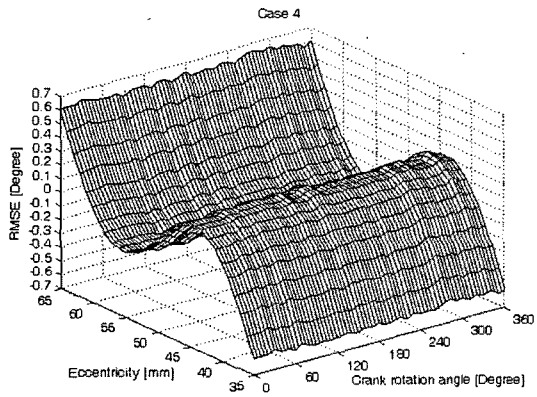


Fig. 8(b) The RMSEs using random BP learning algorithm

No weight change is allowed to be larger than ψ times the previous weight change. The actual results for "Quick Propagation Learning Algorithm" can be outlined in Fig. 9(a).

From Fig. 3 and Fig. 9(a), the deviations between desired and actual results using Quick Propagation Learning Algorithm is shown in Fig. 9 (b).

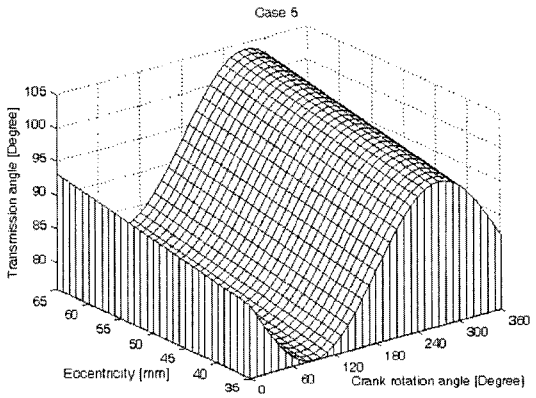


Fig. 9(a) The actual transmission angle variations using quick propagation learning algorithm

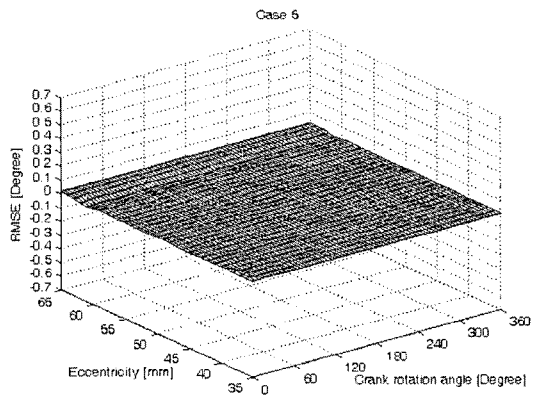


Fig. 9(b) The RMSEs using quick propagation learning algorithm

4. Simulation Results and Discussion

The results obtained using NNs for a modified slider-crank mechanism have been described in this section. The general structure of the neural predictor is shown in Figure 4. Training of the neural network proceeds as follows. Firstly, neural network was trained by using CASE 1 algorithm for 1000000 iteration numbers. The neural network was trained by using all cases as described in section 3.1. The structural and training algorithm of the neural network is outlined in Table 2.

To verify the effectiveness of neural network, simulation results of modeling of a slider-crank mechanism are shown in Figure 5(a)~9(b) for all cases. Figure 5(a) represents variations of transmission angle when eccentricity and crank rotation angles changed. The RMSE results are given in Figure 5(b) for Case 1. Figure 6(a) and Figure 6(b) show the transmission angle tracking for Case 2 when eccentricity and crank rotation angles of mechanism changed. The results of transmission angle of modified slider-crank mechanism for Case 3 are plotted in Figure 7(a). RMSEs variation for this case is given in Figure 7(b). Figure 8(a) shows the output of the neural network predictor for case of 4. The results of this case are rather worse than the cases of 1, 2, 3 and 5. RMSEs for the case of 4 are depicted in Figure 8(b). Figure 9(a) indicates the results of Case 5. As can be depicted from figure, the case of Quick Propagation learning algorithm gives the best results of all cases. RMSEs for the case of 5 are given in Figure 9(b).

Most of neural network models used for prediction and modeling are feed-forward networks.

Table 2 Training parameters and RMSE of the network for the different learning algorithms

NN type	n_i	n_j	n_k	η	α	N	RMSE	Learning Algorithm
Case 1	1	10	37	0.05	0.001	1000000	0.01627	Online-Backpropagation
Case 2	1	10	37	0.05	0.001	1000000	0.01565	Batch Backpropagation
Case 3	1	10	37	0.05	0.001	1000000	0.01079	Delta-Bar-Delta
Case 4	1	10	37	0.05	0.001	1000000	0.24845	Random Backpropagation
Case 5	1	10	37	0.05	0.001	1000000	0.004388	Quick Propagation

Since feed-forward nets can only represent memoryless transformations, one needs to explicitly feed all the past inputs and outputs of the mechanisms.

5. Conclusions

The prediction of a proposed and modified slider-crank mechanism has been discussed in this paper. Neural networks have been employed to find exact transmission angle parameters of the slider-crank mechanism. The case of 5, Quick Propagation neural network predictor has given the best performance rather than the cases of 1, 2, 3 and 4.

Based on the simulation results, it is observed that the proposed NN is promising in the following aspects. First, the QP learning algorithm is very efficient. After the task is repeated only a few times, the neural network is well trained so that the desired tracking performance can be achieved. Second, the network structure is simple which is suitable for real time prediction purpose. Third, in comparison with the results obtained in all cases, this prediction technique does not require knowledge of the bounds of uncertainties of the mechanism model. Thus, it can be implemented to predict a more general class of dynamic system whose model is unknown.

Acknowledgments

This work is a part of the research project FBA-04-15. The authors wish to express their thanks for financial support being provided by the Scientific Research Project Fund of Erciyes University, in carrying out this study.

References

Attia, H. A., 2003, "Kinematic Analysis of the Multi-Link Five-Point Suspension System in Point Coordinates," *KSME International Journal*, Vol. 17, pp. 1133~1139.

Canbulut, F., Sinanoglu, C. and Yıldırım, Ş., 2004, "Analysis of Effects of Sizes of Orifice and Pockets on the Rigidity of Hydrostatic Bearing Using Neural Network Predictor System," *KSME International Journal*, Vol. 18, pp. 432~442.

Choi, K. B, 2003, "Kinematic Analysis and Optimal Design of 3-PPR Planar Parallel Manipulator," *KSME International Journal*, Vol. 17, pp. 528~537.

Hamilton, H. M. and Charles, F. R., 1987. *Mechanisms and Dynamics of Machinery*, John Wiley & Sons, Canada.

Kim, J. K. and Jung, J. Y., 2003, "Driving Mechanism of Tapered Pistons in Bent-Axis Design Axial Piston Pumps," *KSME International Journal*, Vol. 17, pp. 181~186.

Lee, M. K., Kim, T. S., Park, K. W. and Kwon, S. H., 2003, "Constraint Operator for the Kinematic Calibration of a Parallel Mechanism," *KSME International Journal*, Vol. 17, pp. 23~31.

Shrinivas, S. B. and Satish, C., 2002, "Transmission Angle in Mechanisms (Triangle in Mech)," *Mechanism and Machine Theory*, Vol. 37, pp. 175~195.

Söylemez, E., 2002, "Classical Transmission Angle Problem for Slider-crank Mechanisms," *Mechanism and Machine Theory*, Vol. 37, pp. 419~425.

Vasiliu, A. and Yannou, B., 2001, "Dimensional Synthesis of Planar Mechanisms Using Neural Networks : Application to Path Generator Linkages," *Mechanism and Machine Theory*, Vol. 36, pp. 299~310.

Yıldırım, Ş. and Uzmay, İ., 2003, "Neural network applications to vehicle's vibration analysis," *Mechanism and Machine Theory*, Vol. 38, pp. 27~41.

Yıldırım, Ş., 2004, "Vibration Control of Bus Suspensions Using a Proposed Neural Network," *Journal of Sound and Vibration*, Vol. 277, pp. 1059~1069.