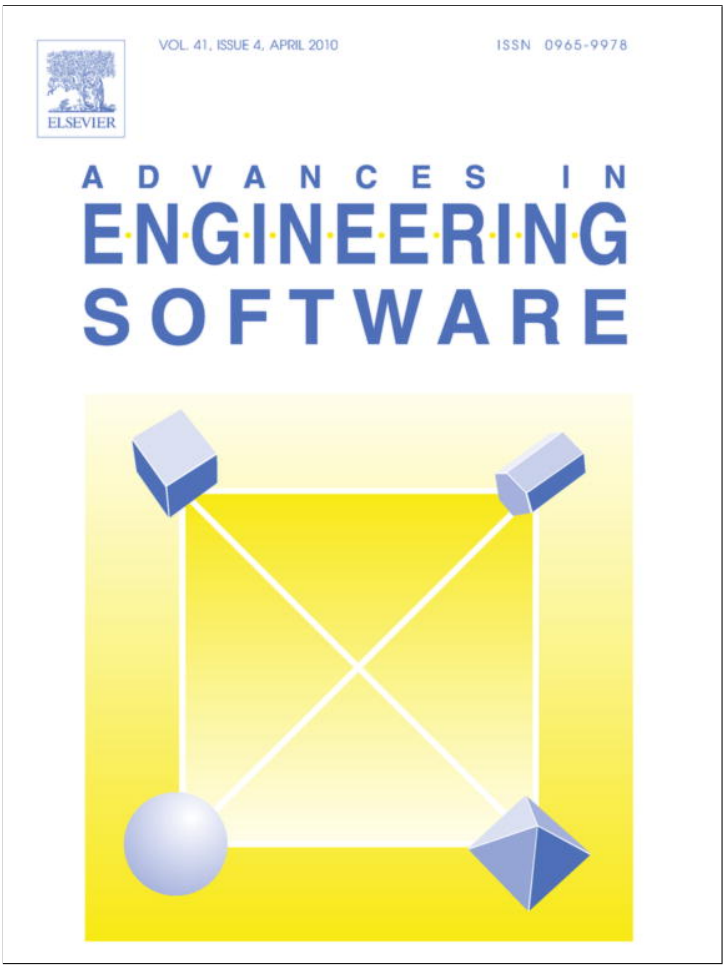

Advances in Engineering Software 41 (2010) 619–626

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

Neuro–fuzzy modeling tools for estimation of torque in Savonius rotor wind turbine

J. Sargolzaei [a,*], A. Kianifar [b]

[a] Department of Chemical Engineering, Ferdowsi University of Mashhad, P.O. Box 9177948944, Mashhad, Iran
[b] Department of Mechanical Engineering, Ferdowsi University of Mashhad, P.O. Box 9177948944, Mashhad, Iran

ARTICLE INFO

Article history:
Received 19 May 2009
Received in revised form 20 July 2009
Accepted 25 November 2009
Available online 4 January 2010

Keywords:
Savonius rotor
Torque
Modeling
ANFIS
RBF
FIS

ABSTRACT

In the present paper, the ability and accuracy of an adaptive neuro–fuzzy inference system (ANFIS) has been investigated for dynamic modeling of wind turbine Savonius rotor. The main objective of this research is to predict torque performance as a function of the angular position of turbine. In order to better understanding the present technique, the dynamic performance modeling of a Savonius rotor is an important consideration for the wind turbine design procedure. It could be difficult to derive the exact mathematical derivation for the input–output relationships because of the complexity of the design algorithm. In order to show the best fitted algorithm, an extensive comparison test was applied on the ANFIS (adaptive neuro–fuzzy inference system), FIS (fuzzy inference system), and RBF (radial basis function). Resulting from the extensive comparison test, the ANFIS procedure yields very accurate results in comparison with two alternate procedures. The results show that there is an excellent agreement between the testing data (not used in training) and estimated data, with average errors very low. Also FIS with threshold 0.05 and the trained ANFIS are able to accurately capture the non-linear dynamics of torque even for a new condition that has not been used in the training process (testing data). For the sake of comparison, the results of the proposed ANFIS model is compared with those of the RBF model, as well. For implementation of the present technique, the Matlab codes and related instructions are efficiently used, respectively.

1. Introduction

Wind turbine is a device to change wind energy into mechanical energy. These turbines are classified into two categories, horizontal and vertical axes. The horizontal axis wind turbines have complicated structures and are economically valuable only in areas where the permanent winds and high speeds are available and are mainly employed for generating electricity. The vertical axis wind turbines (VAWTs) such as Savonius turbines have a simple structure and are capable to operate at low wind speed [1,2]. Unlike horizontal axis turbines, in vertical axis turbines, the rotation speed is low and torque is high [3]. These turbines are independent of the wind direction [4,5]. In this research, for Savonius rotor which is a VAWT, the blades were tested in the wind tunnel. Therefore, there is no lift, so the drag force should be high. Actually, this high drag force causes a substantial reduction in the rate of power output. The neural networks and fuzzy systems have certain advantages over classical methods, especially when the vague data is existent or the prior knowledge is required. However, the applicability of these hybrid (neuro–fuzzy) modeling techniques could be very limited for modeling of some engineering problems [6].

Therefore, combinations of neural networks with fuzzy systems have been proposed, where both models complement each other. These so-called neural fuzzy or neuro–fuzzy systems allow of the individual weaknesses to overcome some and offer appealing features [7].

Takagi and Sugeno [8] first presented numeric analysis approach of fuzzy system and then many studies have been made [9–14]. Since the systems using fuzzy theory can express rules or knowledge as "if–then" form, they have advantages such as they do not need mathematical analysis for modeling. However, they need the appropriate model construction and parameter selection [9–14]. The use of expert systems for natural-language interaction distinguishes the present framework presented in [15–17]. This kind of fuzzy modeling problem is a trouble some work in general. On the other hand, studies of fuzzy neural networks that combine both advantages of the fuzzy systems and the learning ability of

Abbreviations: ANFIS, adaptive neuro–fuzzy inference system; BP, backpropagation; RBF, radial basis function; ANN, artificial neural network; FIS, fuzzy inference system; RMSE, root mean square-error; MSE, mean square-error; LSE, least square-error; Min, minimum; Max, maximum; Prob, probor; Prod, product; SD, standard deviation.

* Corresponding author. Tel.: +98 915 512 4591; fax: +98 511 8816840.
E-mail addresses: sargolzaei@um.ac.ir (J. Sargolzaei), a-kiani@um.ac.ir (A. Kianifar).

0965-9978/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.advengsoft.2009.12.002

Nomenclature

$x_1, x_2, \ldots, x_6$	input variables	v	speed vector
$\{p_i, q_i, \ldots, k_i, r_i\}$	consequent parameters	r	rotating speed of blade
$\{x^*, \sigma\}$	premise parameters	$\sum$	sigma (sum)
$A_1, B_1, \ldots, C_1$	antecedent	N	number
f	apodosis	c^*	centre
θ	angular position of turbine	σ^*	width
T	torque of vertical force to blade's surface		

the neural networks have been carried out. These techniques can alleviate the matter of fuzzy modeling by learning ability of neural networks and have been reported since around the beginning of 1990s [8,10]. Fuzzy neural networks can be applied not only for simple pattern classification but also for also meaningful fuzzy if–then rules creation; therefore, they can be put into practice for various applications. In the early stage of fuzzy neural network researches, Lin et al. [9] proposed one of the current prima models that decide the initial fuzzy model by Kohonen's self-organizing algorithm [18] and carry out parameter adjustment by back propagation algorithm. Also as a representative example, Jang et al. proposed ANFIS [19] in 1993. ANFIS applies a neural network in determination of the shape of membership functions and rule extraction. However, since it needs to divide the input data space in advance, accuracy of the system depends much on the achievement of this pre-processing. Wang et al. [20] reported an approach to acquire fuzzy rules by dividing input space. These techniques, however, do not consider the output data space, so the obtained rules should not be always reasonable. Since the architecture and behavior of ANFIS are very applicable [21], it has been adopted as a basic component for interpretation researches [22,23]. However, its fuzzy modeling for the target task is not always sufficient. In this paper, an adaptive fuzzy inference neural network (ANFIS) is used that alleviates these shortcomings of the conventional models.

2. Structure of ANFIS in wind turbine

An ANFIS can divide input–output data space and provide appropriate rules automatically. Fig. 1 shows the structure of ANFIS for wind turbine which consists of three intermediate layers. The first layer is the input (I) layer, second is the intermediate (hidden layer) or rule-layer, and last is output (O) layer. The input and output layers consist of the output-part and the input-part. Each node in the rule-layer represents one fuzzy rule. Weights from the input-part to the rule-layer and those from the rule-layer to the output-part are fully connected and they store fuzzy if–then rules. Membership functions as premise part are expressed in the weights. Each weight from the rule-layer to the output-part corresponds to the estimated value of each rule. In short, the weights from the input-part to the rule-layer indicate if-parts of fuzzy if–then rules and those from the rule-layer to the output-part indicate then parts [24]. The shapes of membership functions are adjusted automatically in the learning phase.

3. Adaptive neuro–fuzzy inference systems (ANFIS)

From Sugeno fuzzy model, adaptive neural–fuzzy inference system (ANFIS) was proposed by Roger Jang in 1992 [7]. The architecture of a one-input seven-rule ANFIS is shown as Fig. 1. In ANFIS architecture, a FIS is described in a layered, feed-forward network structure where some of the parameters are represented by adjustable nodes and the others as fixed nodes. The raw inputs are fed into the layer one nodes that represent the membership functions (MF) that it is one MF in this study for each input.

For a first-order Sugeno fuzzy model, a common rule set with two fuzzy if–then rules is the following:

$$\begin{aligned}\text{Rule 1}: &\ \text{If } x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } B_1 \text{ and} \ldots x_6 \text{ is } C_1,\\ &\ \text{then } f_1 = p_1x_1 + q_1x_2 + \ldots + k_1x_6 + r_1,\\ \text{Rule 2}: &\ \text{If } x_1 \text{ is } A_2 \text{ and } x_2 \text{ is } B_2 \text{ and} \ldots x_6 \text{ is } C_2,\\ &\ \text{then } f_2 = p_2x_1 + q_2x_2 + \ldots + k_2x_6 + r_2.\end{aligned}$$

The ANFIS has five layers, in which node functions of the same layer have the same function type as described below: (Note that O_{ij} denotes the output of the ith node in the jth layer.)

Layer 1: Every node i in this layer is an adaptive node with node function:

$$\mu_{A'}(x) = \exp\left(-((x - x^*)/\sigma)^2\right) \tag{1}$$

where $\{x^*, \sigma\}$ are premise parameters updated through hybrid learning algorithm and x is input variable. At least in the basic ANFIS method these parameters are not adjustable.

Layer 2: Every node i in this layer is a fixed node labeled $\prod$, whose output is the product of all the incoming signals:

$$O_{2,i} = \omega_i = \prod_{i=1}^{n} \mu_h(x_i) = \mu_{Ai}(x_1) \times \ldots \times \mu_{Bi}(x_n) \tag{2}$$

where $x, \ldots, x_n$ are input variables and n is nodes number.

Layer 3: Every node i in this layer is a fixed node labeled N. The ith node calculates the ratio of the ith rule's firing strength to the sum of all rules' strengths.

$$O_{3,i} = \varpi_i = \omega_i \Big/ \sum_{i=1}^{n} \omega_i \tag{3}$$

This layer implements a normalization function to the firing strengths producing normalized firing strengths.

Layer 4: The single node in this layer is a fixed node labeled $\sum$, which computes the overall output as the summation of all incoming signals:

$$\text{Overall output} = O_4 = \sum_i \varpi_n f_i = \sum_i \omega_i f_i \Big/ \sum_i \omega_i \tag{4}$$

where $f_i = p_ix_1 + q_ix_2 + \ldots + k_ix_6 + r_i$, $x_1, x_2, \ldots, x_6$ are input variables, $\{p_i, q_i, \ldots, k_i, r_i\}$ are consequent parameters updates through recursive least-squares estimation (LSE). The fifth layer represents the aggregation of the outputs performed by weighted summation. It is not adjustable.

4. Hybrid learning rule: combing BP and LSE

A hybrid learning algorithm [25,26] was proposed as follow:

In the forward pass, node outputs go forwards until layer 3 and the consequent parameters are identified by the least-squares

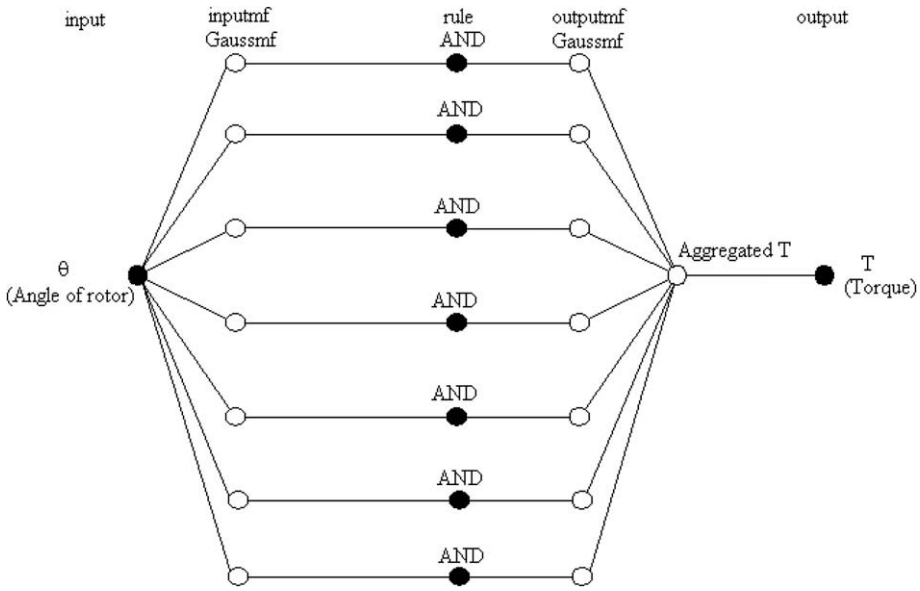

Fig. 1. ANFIS architecture of a single-input–single-output with seven rules in wind turbine system.

method; in the backward pass, the error signals propagate backward and the premise parameters are updated by gradient descent. These procedures are summarized as Table 1.

5. Materials and methods

5.1. Calculating power of wind force

Following equation is useful to calculate power produced by turbine:

$$P_t(\theta) = F(\theta) \cdot v(\theta) = T(\theta)r(\theta) \tag{5}$$

θ is angular position of turbine, T is torque of vertical force to blade's surface (force of air pressure), v is speed vector in force point of F, and r is rotating speed of blade. Product of dot multiply in Eq. (5) shows that only the factor of the force with the same direction of rotation is effective to produce power. Therefore, blade's curve in vertical axis turbine is very important [1–3].

5.2. Produced samples

Savonius rotor has been tested with six different blade's curves in a square section wind tunnel to dimension $0.4 \times 0.4 \times 14$ m. In rotors I–V each blade is a semicircle to the diameter value 16 cm. Values of S distances (gap) are 0, 3.2, 3.8, 6.4, and 7.2 cm for rotors I–VI, respectively. These gap distance change amount of drag force on back and front of blade in different angles in proportion to blowing wind. The blade's curve in rotor VI is Savonius curve which is similar to rotor IV in dimensions. Height (H) in all produced models is about 30 cm, thickness of blade is 1 mm, and it is made of aluminum. Fig. 2 shows rotor shape.

Table 1
Two passes in the hybrid learning procedure for ANFIS.

Forward pass		Backward pass
Premise parameters	Fixed	Gradient descent
Consequent parameters	Least-squares estimator	Fixed
Signals	Node outputs	Error signals

5.3. Experimentation of different blades in wind tunnel

Blade's power factor is calculated by measuring rotating speed of rotor round axis and outlet torque which is measured by two special dynamometer connected to the end of each blade. All experiments are done in the same situation and wind speed varies from 8 to 14 m/s. In the first experiment, rotation speed and torque of each rotor are measured in a complete rotation and results are compared with other rotors.

The result of experiments for rotors I and IV are presented. Results of previous experiments make it possible to calculate and compare average power factor in a complete rotation in a specific wind speed. This comparison could be a good standard to select the rotor with the best efficiency.

6. Results

In this work, the applications of ANFIS for prediction of torque were tested for wind turbine at different operation conditions. The modeling results are presented for two option (θ, torque) follow.

6.1. Method of applications

The ANFIS information and errors are shown in Table 2 that used for all rotors. The generation fuzzy inference system (FIS) method is Subtractive Clustering. There are 0.5, 1.25, 0.5, 0.15 values for ranges of influence, squash factor, accept ratio, and reject

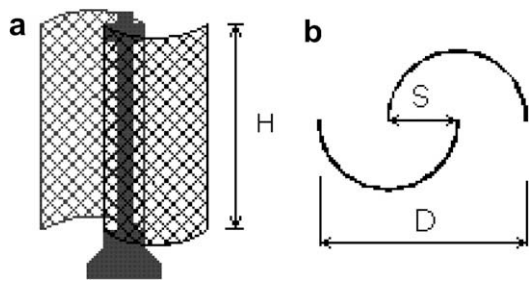

Fig. 2. Schematic of a Savonius rotor: (a) front view and (b) semicircle shape.

Table 2
The ANFIS information used in this study by hybrid optimum method.

Number	Rotor I	Rotor II	Rotor IV	Rotor V
Nodes	20	20	24	20
Linear parameters	8	8	10	8
Non-linear parameters	8	8	20	8
Training data pairs	10	10	10	10
Fuzzy rule	4	4	5	4
Epochs optimum	1000	2000	20,000	5000
Training error	0.023018	0.02821	0.01857	0.01983
Testing error	0.02497	0.02799	0.01868	0.02053
Checking error	0.02452	0.027996	0.0184	0.020144

ratio. Also the optimum method is hybrid. In this research, two methods hybrid and backpropagation tested for generation FIS that the results is presented in Tables 2 and 3. Also the best threshold of the generation FIS is 0.05 for all rotors simulation. The convergence threshold is effectiveness factor to checking error of FIS modeling. Changing the threshold of convergence reduces error rate and the best model of FIS is built. The results show the training error in the hybrid method is lower of backpropagation method. Therefore, the hybrid method has used for simulations.

Table 3
The ANFIS information used in this study by BP optimum method.

Number	Rotor I	Rotor II	Rotor IV	Rotor V
Nodes	16	16	16	16
Linear parameters	6	6	6	6
Non-linear parameters	6	6	6	6
Training data pairs	24	24	24	24
Fuzzy rule	3	3	3	3
Epochs optimum	1000	1000	1000	1000
Training error	0.001933	0.002941	0.005636	0.005169
Testing error	0.002160	0.00336	0.00790	0.007536
Checking error	0.002145	0.003304	0.00792	0.00754

6.2. Model training and testing

The model was trained with part of the database derived from the experimental work described above. The database was first split into training data and testing data. Table 4 presents the training and testing data. The training data set was also split into two parts, a training set and a checking set. The use of checking sets in ANFIS learning beside the training set is a recommended technique to guarantee model generalization and to avoid over-fitting the model to the training data set. The Gaussian membership function is bounded between zero and one, so the input and output data should be normalized. Therefore, the database has normalized. Normalization of data leads to avoidance of numerical overflows due to very large or very small weights [27,28].

The successful training process was accomplished using different training epochs (iterations) for rotors. The ANFIS network was able to achieve training and checking the lowest RMSE for rotors. The Fig. 3 shows training plot achieve with ANFIS for all rotors. The results of modeling using ANFIS for the torque at data set (testing data) are shown in Fig. 4. It can be seen that the magnitudes of torque vary significantly with index (data values). The figures also show that the complex behavior (non-linearity) of torque profile is well reproduced by the ANFIS. As shown in Fig. 4, there is agreement between the output ANFIS behavior with testing data.

During the ANFIS training, the training set up foresaw the analytical forms of prod and probor operators for the connectors AND and OR, respectively, the min for the if–then implication, the max for the ELSE aggregation, and the defuzzification method Wtaver produced the crisp output [7]. The whole procedure was implemented on a Pentium IV 330 MHz, using Matlab 7.0 (Mathworks Inc.). The ANFIS is the best manufacturer of FIS. Note that the basic

Table 4
The training and testing data for modeling of Savonius rotors.

Training		Testing		Training		Testing	
θ (°)	Torque (N m)	θ (°)	Torque (N m)	θ (°)	Torque (N m)	θ (°)	Torque (N m)
Rotor I				*Rotor II*			
0.0500	0.0557	0.0528	0.0688	0.0500	0.0614	0.0528	0.0643
0.0556	0.0815	0.0569	0.0859	0.0556	0.0732	0.0569	0.0757
0.0611	0.0898	0.0597	0.0888	0.0611	0.0857	0.0597	0.0821
0.0667	0.1000	0.0639	0.0956	0.0667	0.1000	0.0639	0.0929
0.0722	0.0792	0.0681	0.0958	0.0722	0.0893	0.0681	0.0982
0.0778	0.0531	0.0708	0.0893	0.0778	0.0732	0.0708	0.0929
0.0833	0.0500	0.0750	0.0604	0.0833	0.0500	0.0750	0.0775
0.0889	0.0549	0.0792	0.0500	0.0889	0.0643	0.0792	0.0679
0.0944	0.0607	0.0819	0.0500	0.0944	0.0661	0.0819	0.0554
0.1000	0.0708	0.0861	0.0500	0.1000	0.0661	0.0861	0.0571
		0.0903	0.0560			0.0903	0.0661
		0.0931	0.0589			0.0931	0.0661
		0.0972	0.0661			0.0972	0.0661
Rotor IV				*Rotor V*			
0.0500	0.0500	0.0528	0.0582	0.0500	0.0512	0.0528	0.0613
0.0556	0.0656	0.0569	0.0736	0.0556	0.0702	0.0569	0.0750
0.0611	0.0847	0.0597	0.0813	0.0611	0.0887	0.0597	0.0833
0.0667	0.1000	0.0639	0.0933	0.0667	0.1000	0.0639	0.0970
0.0722	0.0851	0.0681	0.0946	0.0722	0.0810	0.0681	0.0940
0.0778	0.0675	0.0708	0.0885	0.0778	0.0643	0.0708	0.0857
0.0833	0.0521	0.0750	0.0767	0.0833	0.0500	0.0750	0.0732
0.0889	0.0644	0.0792	0.0595	0.0889	0.0643	0.0792	0.0583
0.0944	0.0750	0.0819	0.0566	0.0944	0.0744	0.0819	0.0524
0.1000	0.0503	0.0861	0.0571	0.1000	0.0551	0.0861	0.0565
		0.0903	0.0669			0.0903	0.0667
		0.0931	0.0718			0.0931	0.0720
		0.0972	0.0702			0.0972	0.0702

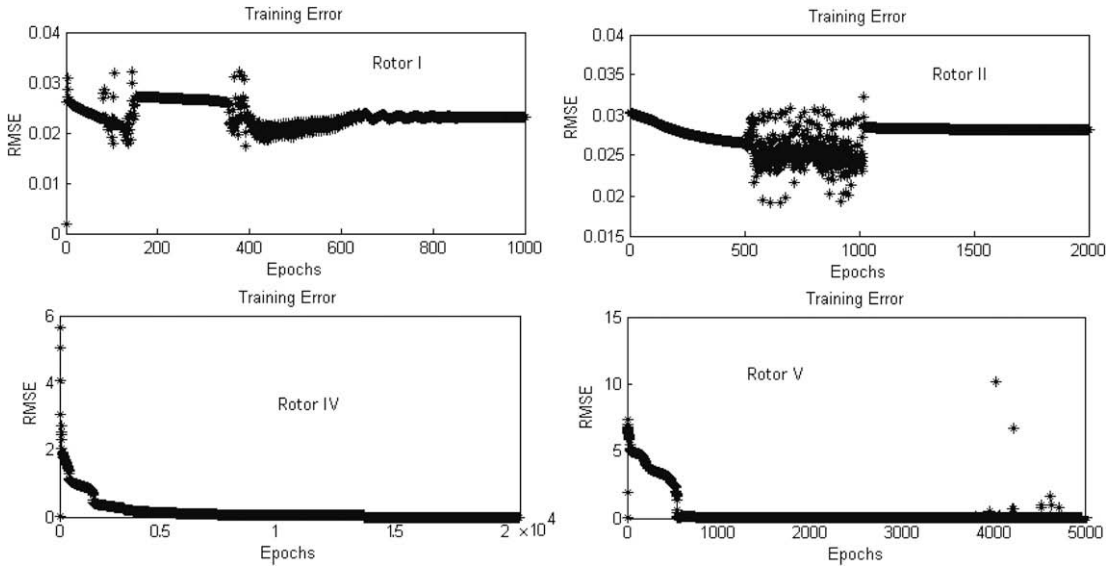

Fig. 3. Shows training RMSE achieve with ANFIS for torque of rotors.

fuzzy inference system (FIS) can take either fuzzy inputs or crisp inputs (which are viewed as fuzzy singletons), but outputs it produces are almost always fuzzy sets. Sometimes it is necessary to have a crisp output, especially in a situation where a fuzzy inference system is used as a controller. Therefore, we need a method of defuzzification to extract a crisp value that best represents a fuzzy set. Defuzzification refers to the way a crisp value is extracted from a fuzzy set as a representative value. In general, there are several methods for defuzzifying a fuzzy set: centroid of area (COA), bisector of area (BOA), mean of maximum (MOM), smallest of maximum (SOM), largest of maximum (LOM), and weighted average (Wtaver). These defuzzification operations are not easily subject to rigorous mathematical analysis, so most of the studies are based on experimental results [7]. Since each rule has a crisp output, the overall output is obtained via weighted average (Wtaver), thus avoiding the time consuming process of defuzzification required in a Mamdani model. In practice, the Wtaver operator is sometimes replaced with the weighted sum (Wsum) operator to reduce computation further, especially in the training of a fuzzy inference system.

6.3. Comparison between ANFIS, fuzzy, RBF

The ANFIS compared with experimental values, RBF, and FIS predicted of torque in wind turbine in the special angle of rotor position range 0–0.1 (normalized data), as shown Fig. 5.

The radial basis function (RBF) structure used for artificial neural network (ANN) prediction that it consists: three layers (the input layer, the hidden layer and the output layer) that the hidden layer is composed of a determined number of nodes or basis functions. These basis functions, also called kernel, can be selected among several types of functions, but for most applications they are chosen to be Gaussian functions. Each node is a Gaussian function, characterized by a centre c^* and a width σ^*

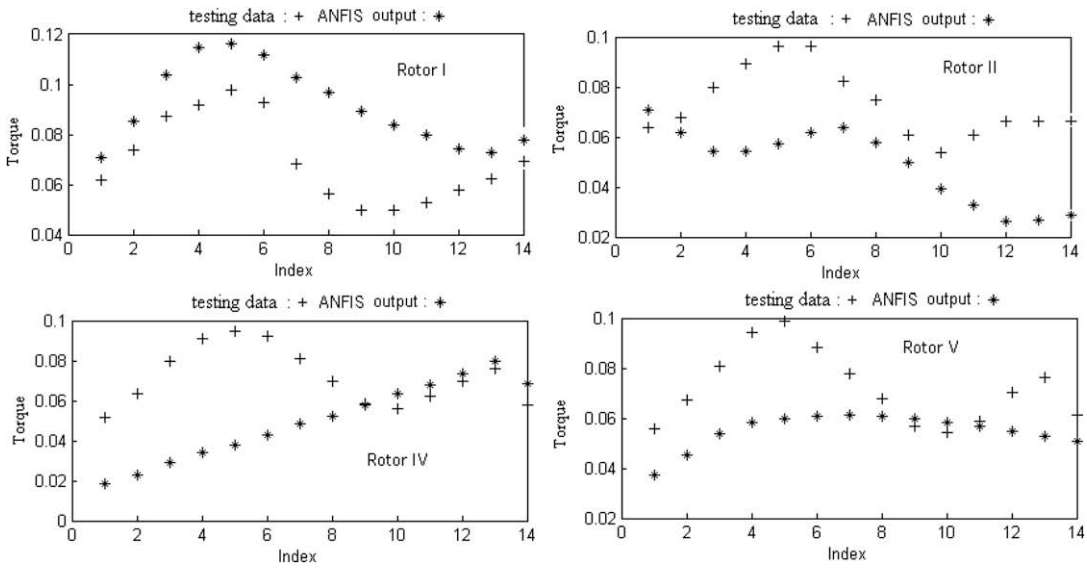

Fig. 4. The results of modeling using ANFIS for the torque at data set (testing data).

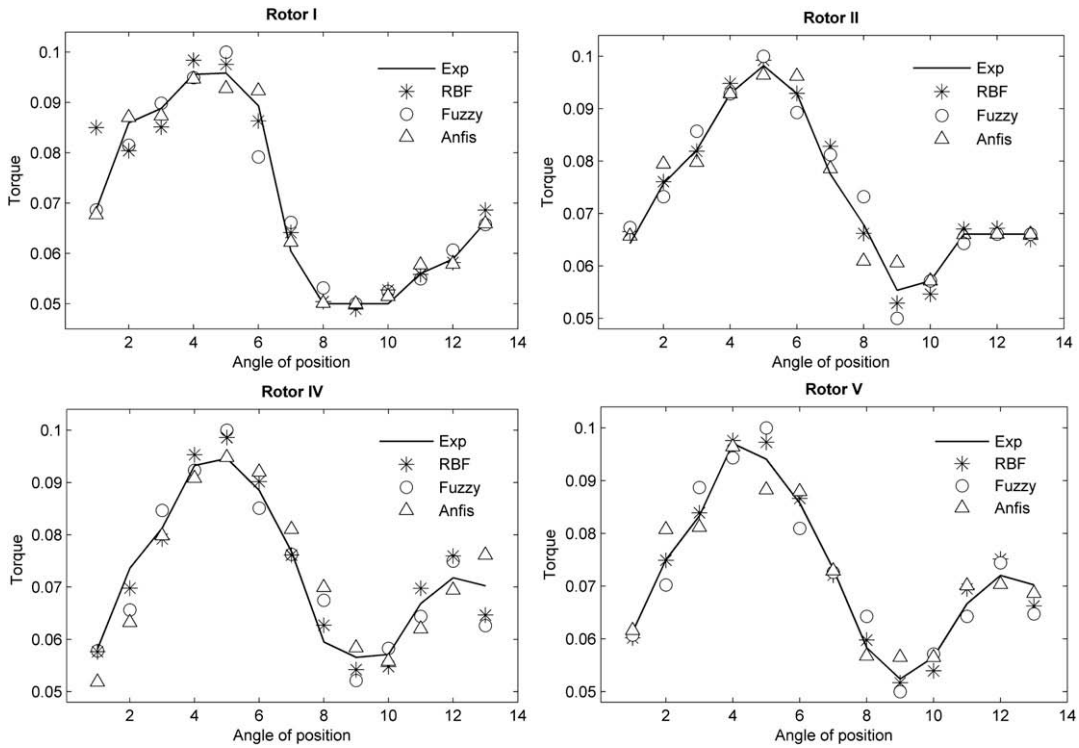

Fig. 5. Comparison between values of desired and ANFIS (hybrid method), RBF (the best spread 0.005), and FIS (the best threshold 0.05) predicted of Savonius rotor wind turbine for torque versus of angle of rotors position.

that produces a non-linear output, being its maximum value when the input corresponds to c and decreasing as the input moves away, that in this research width of the activation functions fixed to a standard value, $\sigma^* = 1$. In step of defining the radial basis layer (number of neurons, centers and bias) and the calculus of the output layer weights, many randomly selected patterns of each training test (N 10 torque) data are used to design the net what we have called the training data set [29]; and the patterns corresponding to maximum, minimum and median values of input and output variables of each training test (N 27

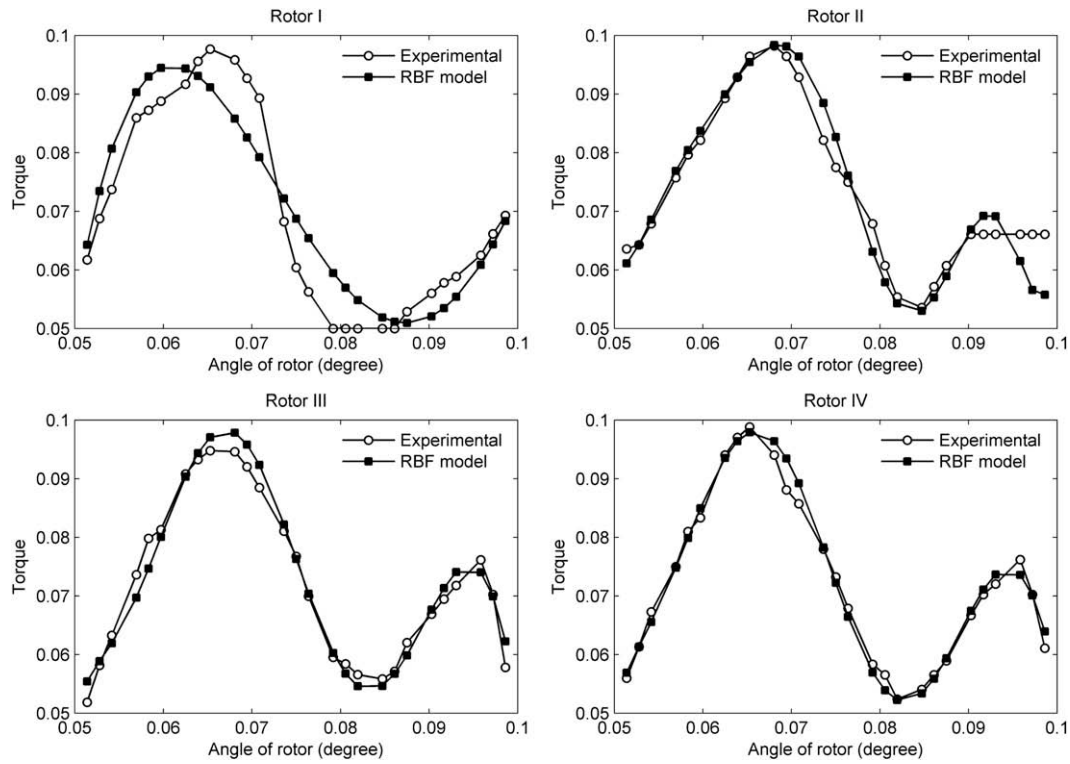

Fig. 6. Comparing different charts obtained from the best RBF nets with experiment data for torque (rotor I: SD = 0.0155, rotor II: SD = 0.0153, rotor III: SD = 0.0082, rotor IV: SD = 0.0069).

Table 5
A common rule set with fuzzy if–then rules for FIS model.

Rotor I
If (angle of position is high) then (torque is low I)
If (angle of position is medium) then (torque is medium I)
If (angle of position is very high) then (torque is high I)
If (angle of position is low) then (torque is low I)
Rotor II
If (angle of position is high) then (torque is low II)
If (angle of position is medium) then (torque is medium II)
If (angle of position is very high) then (torque is high II)
Rotor IV
If (angle of position is high) then (torque is low IV)
If (angle of position is medium) then (torque is medium IV)
If (angle of position is very high) then (torque is high IV)
If (angle of position is low) then (torque is low IV)
Rotor V
If (angle of position is high) then (torque is low V)
If (angle of position is medium) then (torque is medium V)
If (angle of position is very high) then (torque is high V)

Table 6
The RMSE of the three models of ANFIS, RBF, and FIS.

	ANFIS	FIS	RBF
Rotor I	0.005455	0.004008	0.006481
Rotor II	0.003807	0.004484	0.005077
Rotor IV	0.001036	0.003287	0.002561
Rotor V	0.002393	0.003591	0.004911

torque) are used to measure the net model accuracy what we have called the checking data set. The performance of the neural network model evaluated with the root mean square-error (RMSE) and determination coefficient (R^2) between the modeled output and measures of the training data set. Fig. 6 presents a comparison between the testing data and predicted values for the rotors in optimum spread of RBF model.

A set of 'if–then' rules made in the fuzzy system design stage. The result demonstrates that the centroid defuzzification and the guss2mf membership function for Mamdani and the Wtaver defuzzification for Mamdani models produces the best model performance in terms of the drop in the quantity of root mean square-error (low RMSE). Also, the Mamdani model will be able to predict this process with a high degree of accuracy when the threshold is modified. It is apparent that the rules (made of linguistic statement) in FIS structure had a significant effect on its decline. The results are obtained in wind velocity 12 $\frac{m}{s}$ with the threshold modified for the best torque values (the best threshold is 0.05). A common rule set with fuzzy if–then rules is the Table 5.

As shown Fig. 6, fuzzy inference system (FIS) with threshold 0.05 and the ANFIS approximation gave advantages over the RBF method. The results showed that there is an excellent agreement between the ANFIS and FIS with desired data. The ability to predict torque could significantly reduce the computation time and the amount of practical work required before designing a wind turbine. It is studies that the hybrid learning approach is supposed to converge better and faster than BP approach.

The results show that the best threshold in FIS design can be lead to output values corresponding to ANFIS values. The RMSE of the three models have compared in Table 6.

7. Conclusion

This paper presents application of a class of hybrid neuro–fuzzy network to the solution of a non-linear complex design. The primary objectives are both to investigate the capability of adaptive neuro–fuzzy networks and to justify their application to predict torque characteristic of wind turbine. Successful implementations of neuro–fuzzy predictors are described and their performances are illustrated using the results obtained from adaptive neuro–fuzzy networks and showed data in figures. Dynamic modeling of wind turbine performance is very important for designing and better understanding of the present process. In this paper, ANFIS, radial basis function (RBF), and FIS were applied to compare results. The fuzzy inference system (FIS) with threshold 0.05 and ANFIS approximation gave advantages over the other methods and these are able to accurately capture the non-linear dynamics of torque even for new conditions that have not been used in the training process (testing data). The results showed that there is an excellent agreement between the checked data (not used in training) and modeling data, with average errors very low. As well as ANFIS and FIS are compared to artificial neural networks (RBF).

References

[1] Bhatti TS, Kothari DP. Aspects of technological development of wind turbines. J Energy Eng 2003;129:81–95.
[2] Gourieres DL. Wind power plants. Oxford (England): Pergamon Press; 1982.
[3] Menet J-L. A double-step Savonius rotor for local production of electricity: a design study. J Renew Energy 2004;29:1843–62.
[4] Burton FL. Water and wastewater industries: characteristics and energy management opportunities (Burton Engineering) Los Altos, CA, Report CR-106941, In: Electric power research institute report (1996); ES-1.
[5] Menet J, Bourabaa N. Increase in the Savonius rotors efficiency via a parametric investigation. In: European wind energy conference & exhibition, London, UK; 2004.
[6] Ozger M, Yildirim G. Determining turbulent flow friction coefficient using adaptive neuro-fuzzy computing technique. J Adv Eng Soft 2009;40(4):281–7.
[7] Jang JSR, Sun CT, Mizutani E. Neuro–fuzzy and soft computing, International Edition. Prentice-Hall International, Inc.; 1997.
[8] Takagi T, Sugeno M. Structure identification of systems and its application to modeling and control. IEEE Trans Syst Man Cybern 1985;15:116–32.
[9] Lin C-T, Lee C. Neural-network-based fuzzy logic control and decision systems. IEEE Trans Comput 1991;40:1320–36 [special issue on Artificial Neural Networks].
[10] Amano A, Arisuka T. On the use of neural networks and fuzzy logic in speech recognition. In: Proceedings of the international joint conference on neural networks; 1989. p. 301–5.
[11] Lin Y, Cunningham GA. A new approach to fuzzy–neural system modeling. IEEE Trans Fuzzy Syst 1995;3:190–7.
[12] Wong C, Chen CC. A hybrid clustering and gradient descent approach for fuzzy modeling. IEEE Trans Syst, Man Cybern 1999;29:686–93.
[13] Linkens DA, Chen M-Y. Input selection and partition validation for fuzzy modeling using neural network. Fuzzy Sets Syst 1999;107:299–308.
[14] Chen M-Y, Linkens DA. A systematic neuro–fuzzy modeling framework with application to material property prediction. IEEE Trans Syst, Man Cybern 2001;31(5):781–90.
[15] Wasfy TM, Noor Ahmed K. Rule-based natural-language interface for virtual environments. Adv Eng Soft 2002;33:155–68.
[16] Wasfy A, Wasfy T, Noor Ahmed K. Intelligent virtual environment for process training. Adv Eng Soft 2004;35(6):337–55.
[17] Wasfy HM, Wasfy TM, Noor Ahmed K. An Interrogative visualization environment for large-scale engineering simulations. Adv Eng Soft 2004:805–13.
[18] Kohonen T. The self-organizing map. Proc IEEE 1990;78(9):1464–80.
[19] Jang Jyh-Shing R. ANFIS: adaptive-network-based fuzzy inference system. IEEE Trans Syst, Man Cybern 1993;23(3):665–85.
[20] Wang Li-Xin. Training of fuzzy logic systems using nearest neighborhood clustering. In: Proceedings of the second IEEE international conference on fuzzy systems, vol. 1; 1993. p. 93–100.
[21] Adaptive fuzzy inference neural network. <http://www.elsevier.com/locate/patcog>.
[22] Iyatomi H, Hagiwara M. Knowledge extraction from scenery images and the recognition using fuzzy inference neural networks. Trans IEICE (D-II) J82-D-II(4) 1999:685–93 [in Japanese].
[23] Iyatomi H, Hagiwara M. Scenery image recognition and interpretation using fuzzy inference neural networks. Pattern Recogn 2002;35(8):1793–806.
[24] Yildirim G, Ozger M. Neuro–fuzzy approach in estimating Hazen–Williams friction coefficient for small-diameter polyethylene pipes. J Adv Eng Soft 2009;40(8):593–9.
[25] Nauck D, Klawonn F, Kruse R. Foundation of neuro–fuzzy systems. New York: Wiley; 1997.
[26] Hiirsalmi M., Kotsakis E., Pesonen, Wolski A. Discovery of fuzzy models from observation data, VTT information technology, FUME project, version 1.0; December 2000.

[27] Sargolzaei J, Khoshnoodi M, Saghatroleslami N, Mosavi M. Fuzzy inference system to modeling of crossflow milk ultrafiltration. J Appl Soft Comput 2008;8:456–65.
[28] Sargolzaei J, Saghatroleslami N, Khoshnoodi M, Mosavi M. Comparative study of artificial neural nets (ANN) and statistical methods for predicting the performance of ultrafiltration process in the milk industry, Iran. J Chem Chem Eng 2006;25:67–76.
[29] Biedermann JD, Grierson DE. Training and using neural networks to present heuristic design knowledge. Adv Eng Soft 1996;27(1–2):117–28.