Energies **2012**, *5*, 101-118; doi:10.3390/en5010101

OPEN ACCESS

energies

ISSN 1996-1073

www.mdpi.com/journal/energies

Article

Modeling of Turbine Cycles Using a Neuro-Fuzzy Based Approach to Predict Turbine-Generator Output for Nuclear Power Plants

Yea-Kuang Chan [1,2,*] and Jyh-Cherng Gu [1]

[1] Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei 106, Taiwan; E-Mail: jcgu@ee.ntust.edu.tw

[2] Institute of Nuclear Energy Research (INER), No.1000, Wenhua Rd., Longtan Township, Taoyuan County 325, Taiwan

* Author to whom correspondence should be addressed; E-Mail: ykchan@iner.gov.tw; Tel.: +886-3-4711400; Fax: +886-3-4711404.

Received: 15 December 2011; in revised form: 16 January 2012 / Accepted: 16 January 2012 / Published: 19 January 2012

Abstract: Due to the very complex sets of component systems, interrelated thermodynamic processes and seasonal change in operating conditions, it is relatively difficult to find an accurate model for turbine cycle of nuclear power plants (NPPs). This paper deals with the modeling of turbine cycles to predict turbine-generator output using an adaptive neuro-fuzzy inference system (ANFIS) for Unit 1 of the Kuosheng NPP in Taiwan. Plant operation data obtained from Kuosheng NPP between 2006 and 2011 were verified using a linear regression model with a 95% confidence interval. The key parameters of turbine cycle, including turbine throttle pressure, condenser backpressure, feedwater flow rate and final feedwater temperature are selected as inputs for the ANFIS based turbine cycle model. In addition, a thermodynamic turbine cycle model was developed using the commercial software PEPSE® to compare the performance of the ANFIS based turbine cycle model. The results show that the proposed ANFIS based turbine cycle model is capable of accurately estimating turbine-generator output and providing more reliable results than the PEPSE® based turbine cycle models. Moreover, test results show that the ANFIS performed better than the artificial neural network (ANN), which has also being tried to model the turbine cycle. The effectiveness of the proposed neuro-fuzzy based turbine cycle model was demonstrated using the actual operating data of Kuosheng NPP.

Furthermore, the results also provide an alternative approach to evaluate the thermal performance of nuclear power plants.

Keywords: adaptive neuro-fuzzy inference system (ANFIS); neural network; turbine cycle; turbine-generator; nuclear power plant

---

1. Introduction

Nuclear power plants (NPPs) consist of very complex sets of component systems and interrelated thermodynamic processes. Creating accurate simulations to optimize performance is correspondingly difficult. Up until now, to simulate the turbine cycle, the fundamental principle used has been steady-state mass and energy balance equations, which have been well studied worldwide and several solutions have been developed to model the turbine cycle and to evaluate plant performance. PEPSE® is a commercial software application developed by Scientech Inc., and widely used to develop turbine cycle models for power plants under normal operation conditions and provide performance analyses for major components [1]. System modeling and performance evaluation must proceed step by step with PEPSE® to construct a turbine cycle model based on a thermal kit provided by the turbine vendor. Chang *et al*. [2] developed an on-line thermal efficiency monitoring and analysis system to calculate generator output, heat rate, and component operating conditions for the Kuosheng Nuclear Power Plant (NPP) in Taiwan. Heo *et al*. [3] developed a need-oriented turbine cycle simulation toolbox, and Kim and Choi [4] developed a performance upgrade system to aid on-line turbine cycle performance analysis for nuclear power plants in Korea. In addition, Nakao *et al*. [5] developed a general purpose software application to analyze the static thermal characteristics of the power generation system.

In developing the turbine cycle model, a number of researchers have used fundamental steady-state mass and energy balance equations, while others have adopted commercial tools to model the turbine cycle and analyze performance. However, these approaches all have the same drawback. They depend on system models that may deviate from ideal conditions, often involving empirical relationships, approximations of actual processes, and linearization of nonlinear phenomena. Moreover, conventional models usually include a large number of parameters supplied by turbine vendors for modeling the turbine cycle.

A practical alternative to overcome these problems is soft computing, which can be used to solve computationally complex and mathematically intractable problems. The main components of soft computing, fuzzy logic and neural networks, have been shown to be very capable at solving complex nonlinear system identification problems [6]. These methods are the basis of the artificial intelligence concept, which has been widely applied in most fields involving computational studies. The main features of these two methods are the ability to self-learn and self-predict particular desired outputs.

The adaptive neuro-fuzzy inference system (ANFIS) combines these two methods and uses the advantages of both methods [7,8]. Since Jang [7] proposed ANFIS, it has been widely adopted in many real world applications and all achieved high accuracy rates [9–13]. The ANFIS architecture can be used to construct an input-output mapping based on human knowledge and stipulated input-output data pairs. Guo and Uhrig [14] proposed a 3-layer hybrid artificial neural network (ANN) approach to study heat rate and thermal performance in nuclear power plants. This hybrid neural network, combining self-organization and backpropagation neural networks, analyzed plant data and extracted some useful information to help operate the plant more efficiently.

In this study, ANFIS is used to develop a turbine cycle model for Unit 1 of the Kuosheng NPP to estimate turbine-generator output with key parameters. This ANFIS based turbine cycle model is used to estimate turbine-generator output without any prior system knowledge pertaining to the exact structure of the mathematical model. To assess the performance of the neuro-fuzzy based turbine cycle model, we adopted a commercial software program, PEPSE®, for developing the thermodynamic turbine cycle of the Kuosheng NPP. In addition, a neural network based turbine cycle model was also been developed to compare the performance and effectiveness of the ANFIS based turbine cycle model.

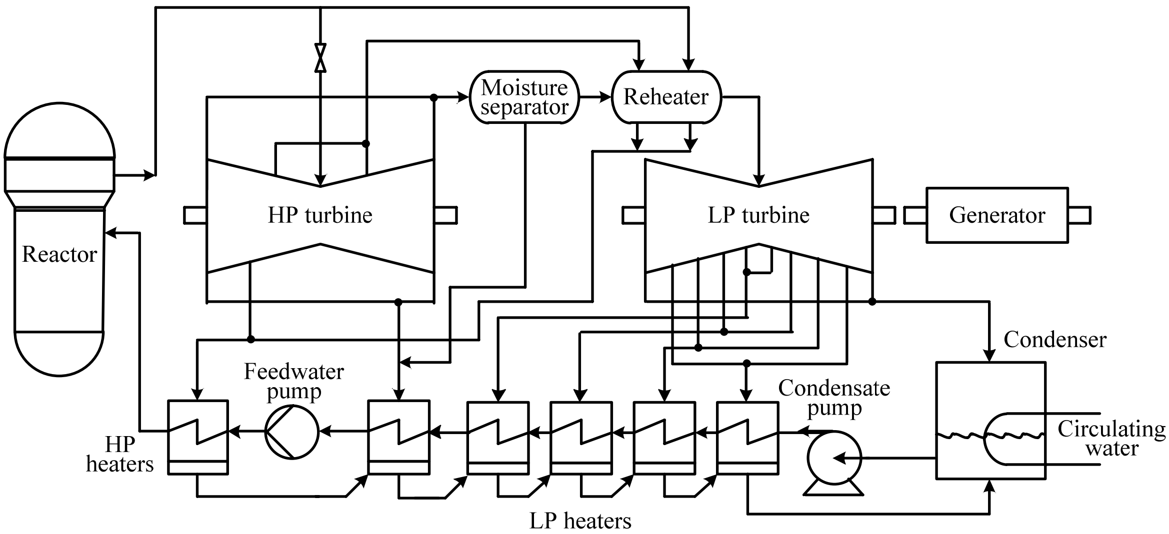
Measurement data for the model was obtained from Unit 1 of the Kuosheng NPP. As this data needs to be validated and verified, a linear regression model is adopted to detect sensor failure or degradation as a reference method. The data collected and validated as the baseline performance data set is the plant's operating data when the plant was operating above a 95% load during the past three fuel cycles. Then, signal errors for new operating data were detected to compare with the baseline data set and their allowable range of variations. The rest of this paper is organized as follows: Section 2 briefly describes the Kuosheng NPP. Section 3 addresses the development of turbine cycle model using the PEPSE® software. The ANFIS and ANN are introduced in Section 4. Section 5 details the development of the ANFIS based turbine cycle model for the Kuosheng NPP. Section 6 presents the results to validate the effectiveness of the proposed ANFIS based turbine cycle model. Finally, Section 7 summarizes this paper.

2. The Kuosheng Nuclear Power Plant

The Kuosheng NPP, owned by the Taiwan Power Company, was the second nuclear power plant constructed in Taiwan. It has two identical GE-designed Boiling Water Reactor (BWR) units. Unit 1 and Unit 2 each had an original licensed thermal power (OLTP) of 2894 MWt, and began commercial operations in December 1981 and March 1983, respectively [15]. Through the implementation of the Measurement Uncertainty Recapture Power Uprate (MUR PU) program, the core thermal power of each unit was uprated to 2,943 MWt (101.7% OLTP) in November 2007 and July 2007, respectively. MUR PU is achieved by using state-of-the-art feedwater flow measurement devices, *i.e.*, ultrasonic flow meters (UFMs), which reduce the degree of uncertainty associated with feedwater flow measurement and in turn, provide a more accurate calculation of core thermal power. The increases in generator output for Kuosheng Unit 1 and Unit 2 due to the MUR PU are approximately 15 MWe and 5 MWe, respectively. Currently, the Kuosheng Stretch Power Uprate (SPU) is being conducted to further increase the core thermal power of both units to 3030 MWt (104.7% OLTP) [16].

Figure 1 shows the simplified schematics of the overall BWR nuclear power plant. The turbine-generator is the primary component that converts the thermal energy produced by the reactor and primary system into electrical power. Each turbine-generator is a Westinghouse Electric Company Model TC4F unit. The turbine for each Kuosheng unit consists of three sections; a double-flow high pressure (HP) section and two double-flow low pressure (LP) sections. The generator is a directly driven, three-phase, 60-Hz, 22,000-volt, 1800-rpm, hydrogen inner-cooled, synchronous generator rated at 1095 MVA [16].

Figure 1. Simplified schematics of the overall BWR nuclear power plant.



3. Modeling of Turbine Cycles Using the PEPSE® Software

The PEPSE® computer code, a turbine cycle simulation tool developed by Scientech Inc., is a steady-state energy balance software program that calculates the performance of electric generating plants. It is widely used the world over for fossil-fired plants, nuclear plants, gas turbine plants, combined cycle plants, and plants with typical fluid systems. A plant analysis model can be constructed by a user developing a plant schematic that mimics the actual plant component connections.

The authors used PEPSE® to build the turbine cycle model for Unit 1 of the Kuosheng NPP. The thermodynamics model of the turbine cycle performs plant's secondary system (turbine cycle) heat balance calculation. The turbine cycle model was developed using the performance mode of PEPSE® on the basis of the 100% power heat balance diagram which was submitted by the turbine cycle vendor, Siemens Inc. The PEPSE® turbine cycle model for Unit 1 of the Kuosheng NPP was developed in a Windows interfaced setting by dragging and dropping plant component icons onto the screen from a component library. The developed turbine cycle model for the Kuosheng NPP is shown in Figure 2. The developed model was validated in two steps as follows:

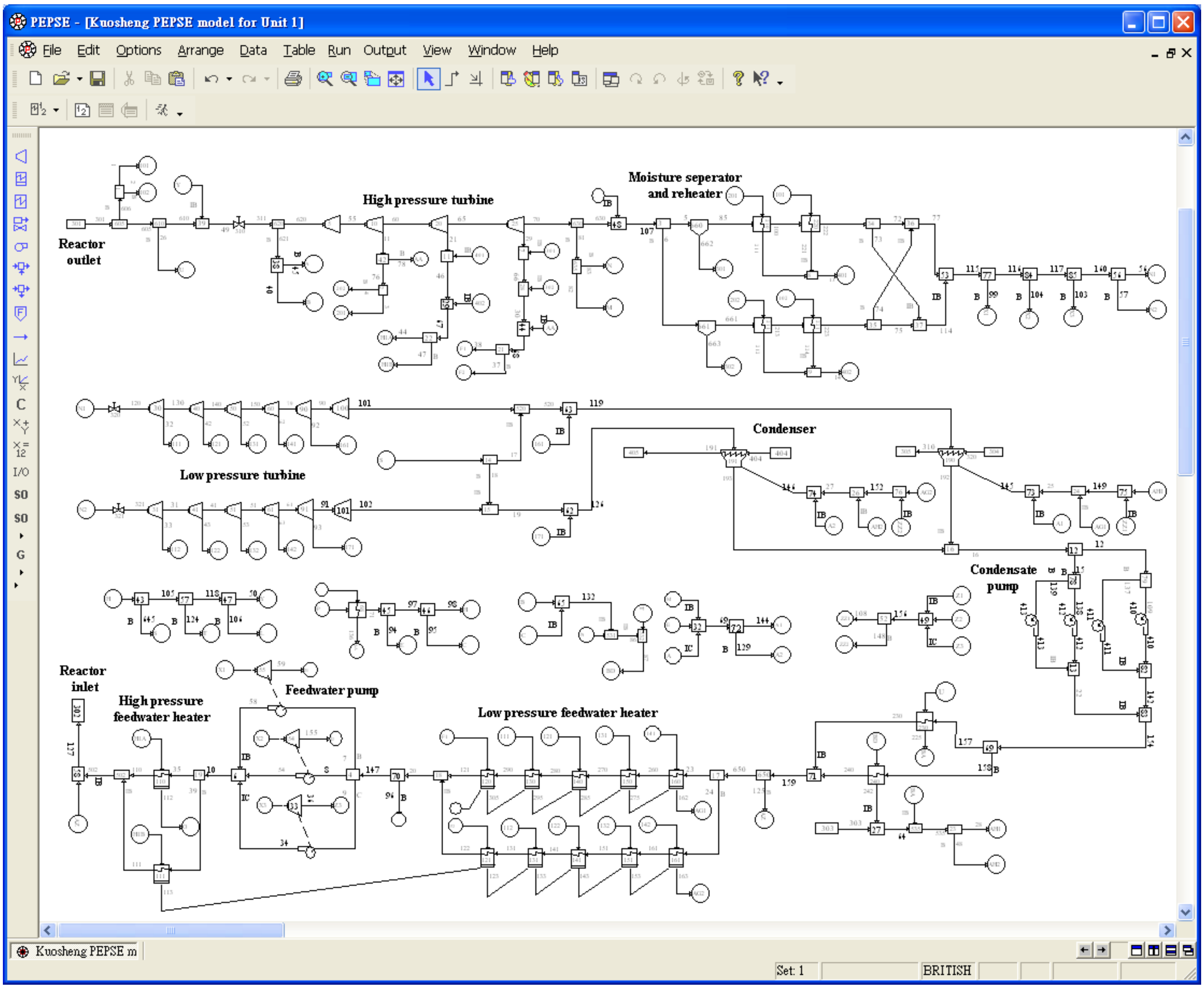
(1) Benchmark to vendor thermal kit [17]:

The developed PEPSE® turbine cycle model (100% load) was checked against the Seimens thermal kit, 100% load heat balance diagram. For the analysis to be more reliable, the PEPSE® turbine cycle model was adjusted so that the results from the PEPSE® model calculation match well with the Siemens thermal kit heat balance calculation and the model was then run under the MUR PU conditions.

(2) Benchmark to plant operating data:

The Kuosheng PEPSE® turbine cycle model was then tuned to the selected plant operating parameters for winter and summer operations to better reflect the plant's actual conditions. The tuned model was then rerun at 100% load and under the MUR PU thermal power conditions.

Figure 2. PEPSE® turbine cycle model for Unit 1 of the Kuosheng NPP.



4. Adaptive Neuro-Fuzzy Inference System (ANFIS) and Artificial Neural Network (ANN)

4.1. Adaptive Neuro-Fuzzy Inference System

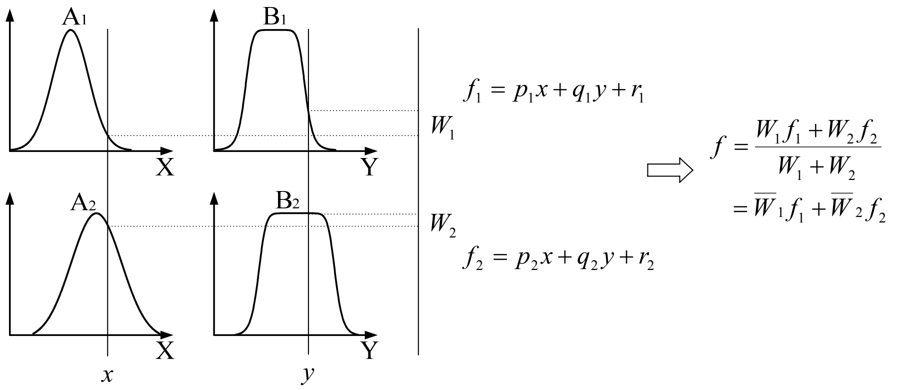
Neural network models are based on data, whereas fuzzy logic models are based on expert knowledge. In situations in which both data and knowledge of the underlying system are available, a neuro-fuzzy approach can exploit both sources of information. This study employs an adaptive neuro-fuzzy system (ANFIS). The system is an adaptive network, functionally equivalent to a first-order Sugeno fuzzy inference system [18]. The ANFIS uses a hybrid learning rule combining backpropagation (gradient-descent) and a least-squares algorithm to identify and optimize the parameters of the Sugeno system.

For simplicity, we assume that the fuzzy inference system under consideration has two inputs, *x* and *y,* and one output, *f*. For a first-order Sugeno fuzzy model, a common rule set with two fuzzy if-then rules is stated as follows [8]:

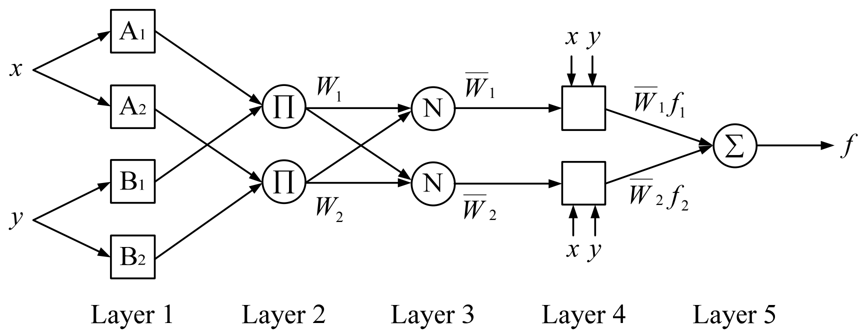
Rule 1: IF x is A_1 and y is B_1, then $f_1 = p_1x + q_1y + r_1$
Rule 2: IF x is A_2 and y is B_2, then $f_2 = p_2x + q_2y + r_2$

Figure 3(a) illustrates the reasoning mechanism for this Sugeno model. The corresponding ANFIS architecture is shown in Figure 3(b). The model has five layers and every node in a given layer has a similar function. In the fuzzy if-then rule set, the outputs are linear combinations of their inputs.

Figure 3. (**a**) Two-input first-order Sugeno fuzzy model with two rules. (**b**) Equivalent ANFIS architecture [8].



(**a**)



(**b**)

Layer 1 consists of adaptive nodes that generate linguistic-label membership grades based on premise parameters, using any appropriate parameterized membership function, such as the generalized bell function:

$$O_{1,i} = \mu_{Ai}(x) = \frac{1}{1 + \left| \dfrac{x - c_i}{a_i} \right|^{2b_i}} \tag{1}$$

where $O_{1,i}$ is the output of the i^{th} node in the first layer, x is the input to node i, A_i is a linguistic label (such as small or large) from the fuzzy set $A = \{A_1, A_2, B_1, B_2\}$ associated with the node, and $\{a_i, b_i, c_i\}$ is the premise parameter set used to adjust the shape of the membership function.

The nodes in layer 2 are fixed nodes, labeled Π, which represent the firing strength of each rule. The output of each node is the fuzzy AND (product or MIN) of all the input signals:

$$O_{2,i} = W_i = \mu_{Ai}(x)\,\mu_{Bi}(y), \qquad i = 1, 2 \tag{2}$$

The outputs of layer 3 are the normalized firing strengths. Each node is a fixed rule labeled *N*. The output of the i^{th} node is the ratio of the i^{th} rule's firing strength to the sum of the firing strengths of all rules:

$$O_{3,i} = \overline{W}_i = \frac{W_i}{W_1 + W_2}, \qquad i = 1, 2 \tag{3}$$

The adaptive nodes in layer 4 calculate the rule outputs based upon consequent parameters using the following function:

$$O_{4,i} = \overline{W}_i f_i = \overline{W}_i (p_i x + q_i y + r_i) \tag{4}$$

where $\overline{W}_i$ is the normalized firing strength from layer 3, and $\{p_i, q_i, r_i\}$ is the consequent parameter set of the node.

The single node in layer 5, labeled Σ, calculates the overall ANFIS output from the sum of the node inputs, as follows:

$$O_{5,i} = \sum_i \overline{W}_i f_i = \frac{\sum_i W_i f_i}{\sum_i W_i} \tag{5}$$

Training the ANFIS is a two-pass process over a number of epochs. During each epoch, node outputs are calculated up to layer 4. At layer 5, the consequent parameters are calculated using a least-squares regression method. The ANFIS output is calculated and the errors are propagated back through the layers in order to determine the premise parameter (layer 1) updates.

4.2. Artificial Neural Network (ANN)

Artificial neural networks (ANN) have been successfully employed in solving complex problems in various fields of applications including power systems, pattern recognition, identification, classification, speech, vision, prediction and control systems [8,19–21]. Today ANNs can be trained to solve problems that are difficult for conventional computers or human beings. ANNs overcome the limitations of the conventional approaches by extracting the desired information directly from the experimental (measured) data. The fundamental processing element of a neural network is a neuron. Basically, a neuron receives inputs from other neurons, combines them in some way, performs a generally non-linear operation and then outputs the final results. The network usually consists of an input layer, some hidden layers and an output layer. Figure 4 shows a typical feedforward ANN that has three layers of neurons. In a neuron, the weighted sum of the neuron's input is processed through a transfer function to produce an output signal. This general activation process in a neuron is also display in Figure 4.

A simplified procedure for the learning process of an ANN is as follows:

- Provide the network with training data consisting of patterns of input variables and target outputs.
- Assess how closely the network output matches the target outputs.
- Adapt the connection strength (*i.e.*, weights) of the various neurons.
- Continue the process of adjusting the weights until the desired accuracy level is achieved.

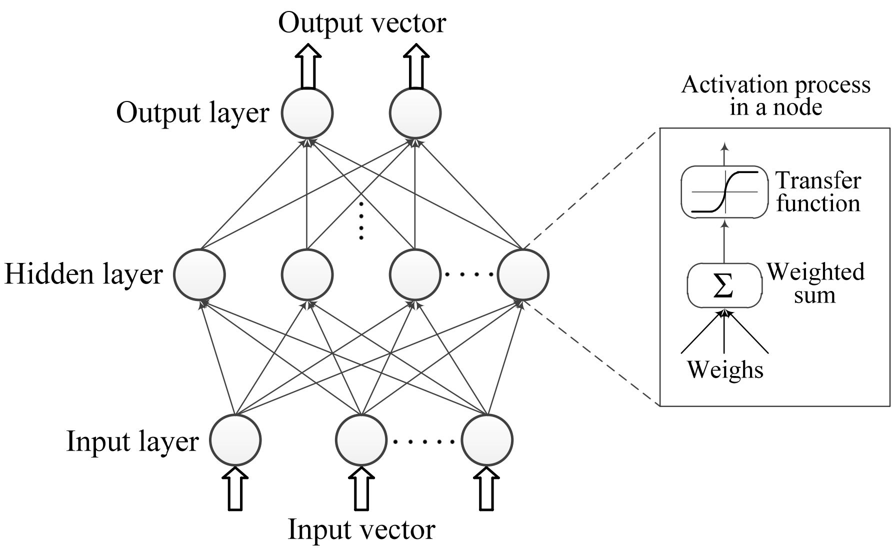
Usually a backpropagation learning algorithm is used.

5. Development of an ANFIS Based Turbine Cycle Model for the Kuosheng NPP

5.1. Operating Data Processing System

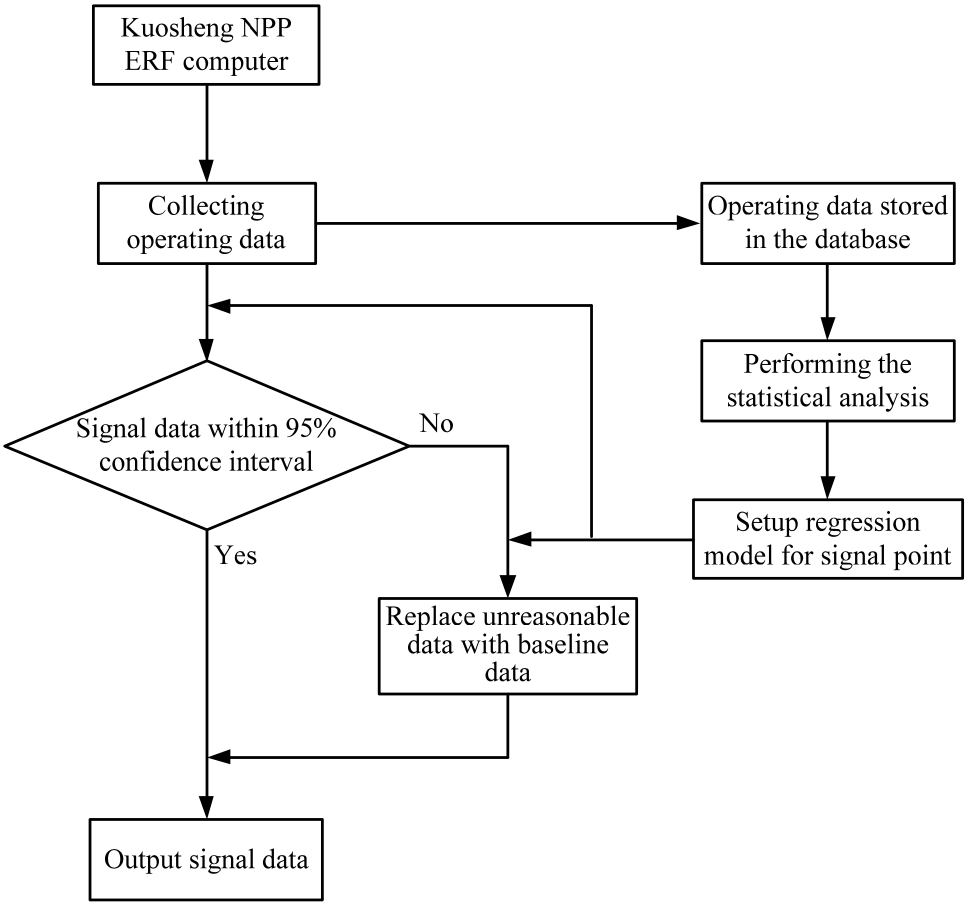
The actual operating data of Unit 1 was obtained from the plant's emergency response facility (ERF) computer from May 2006 to August 2011. Daily routine data was usually collected by the ERF computer over a 40 min period in the morning. The operating data of the plant in the turbine cycle had to be validated and verified to create data used to develop the turbine cycle model to predict the turbine-generator electrical output. To do this, an operating data processing system was developed and Figure 5 shows the overall structure for processing the operating data of the plant.

Figure 4. A typical architecture of multilayered feedforward ANN.



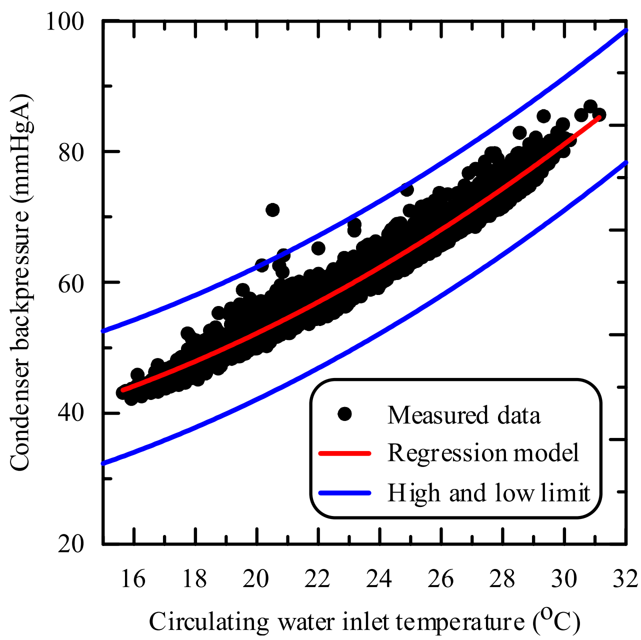
The baseline performance of the turbine cycle was established based on specific key parameters, adjusting for the seasonal effects of circulating water temperature at condenser inlets using the past three fuel cycles. Through a statistical analysis, outliers were determined as those signals lying outside the confidence interval and only data inside the confidential interval were averaged (we used a 95% confidential level). Then, signal errors were detected by comparing them with the reference baseline data and their allowable range of variations. Signal data that deviated from the allowable range were recognized as sensor failures and replaced by the reference baseline data. Linear regression was adopted due to its ease of use, clearly derived process, and effectiveness in estimating important signals.

Figure 5. Overall structure for plant operating data acquisition and processing system.



The solid curve (red color) shown in Figure 6 provides the curve fitting regression model representing condenser backpressure as a function of circulating water inlet temperature.

Figure 6. Regression model of condenser backpressure versus circulating water inlet temperature.



5.2. Determining the Input and Output Variables

The operating limit of a nuclear power plant is directly related to its core thermal power production. The energy balance equation can be expressed as [22]:

$$P_t = W_{fw}(h_s - h_{fw}) \pm P_{loss} \tag{6}$$

where P_t is core thermal power, W_{fw} is feedwater flow rate, h_s and h_{fw} are enthalpies of main steam and feedwater, respectively, and P_{loss} is system losses. The enthalpies, h_s and h_{fw}, are influenced by the turbine throttle pressure and final feedwater temperature, respectively. In addition, the circulating water system of the Kuosheng NPP takes water from the sea and the temperature of the sea water directly influences condenser backpressure. When the circulating water inlet temperature increases, condenser backpressure increases, which in turn reduces the turbine-generator output. The circulating water system provides the cooling water for condensing steam in the main condenser. The main condenser provides the heat sink for the turbine exhaust, turbine bypass steam, and other cycle flows to be returned to the reactor. Maintaining an adequately low condenser backpressure ensures efficient operation of the turbine-generator and minimizes wear on the turbine blades. The heat transferred by condenser (BTU/h) can be expressed as:

$$Q = W_c \cdot c_p \cdot (T_o - T_i) \tag{7}$$

where T_o is circulating water outlet temperature (°F), T_i is circulating water inlet temperature (°F), W_c is circulating water flow (lb_m/h), and c_p is specific heat of salt water (BTU/°F-lb_m). On this basis, the variables that most strongly influence turbine-generator output are the input variables, including turbine throttle pressure, condenser backpressure, feedwater flow rate, and final feedwater temperature. The output variable is the turbine-generator electrical output.

The operating data used for this study were obtained from Unit 1 of the Kuosheng NPP and the method stated in Section 5.1 was applied to verify the data. The turbine throttle pressure, condenser backpressure, feedwater flow rate, final feedwater temperature, and turbine-generator output data were collected for the past three fuel cycles and the current operating cycle from May 2006 to August 2011. As shown in Table 1, ten signal points were used in this study.

Table 1. Selected operating parameters.

ANFIS	**Compute r point**	**Unit**	**Signal description**	**Note**
Input 1	AV 164	kg/cm^{2}G	Turbine throttle pressure	Input 1 (turbine throttle pressure) is the reading of AV 164.
Input 2	AV 203 AV 204	mmHgA	Condenser A backpressure Condenser B backpressure	Unit 1 has two condensers called A and B. Input 2 (condenser backpressure) is the average reading of AV 203 and AV 204.
Input 3	AV 687 AV 688	Ton/h	Feedwater flow rate for the two loops (each loop equipped with one sensor)	Input 3 (total feedwater flow rate) is the summation reading of AV 687 and AV 688.

Table 1. *Cont.*

Input 4	AV 548 AV 549 AV 550 AV 551	°C	Final feedwater temperature for the two loops (each loop equipped with two sensors)	There are totally 4 sensors for measuring the final feedwater temperature. Input 4 (final feedwater temperature) is the average reading of AV 548, AV 549, AV 550, and AV 551.
Output	AV 228	MWe	Turbine-generator electrical output	

The trends of selected operating parameters are shown in Figure 7. The relationship between the turbine-generator output and condenser backpressure are shown in Figure 8. As can be seen in Figure 8, the turbine-generator output was influenced mainly by condenser backpressure under normal operating conditions and the turbine-generator output is inversely proportional to condenser backpressure.

Figure 7. Trend data for generator output, turbine throttle pressure, condenser backpressure, feedwater flow rate, and final feedwater temperature.

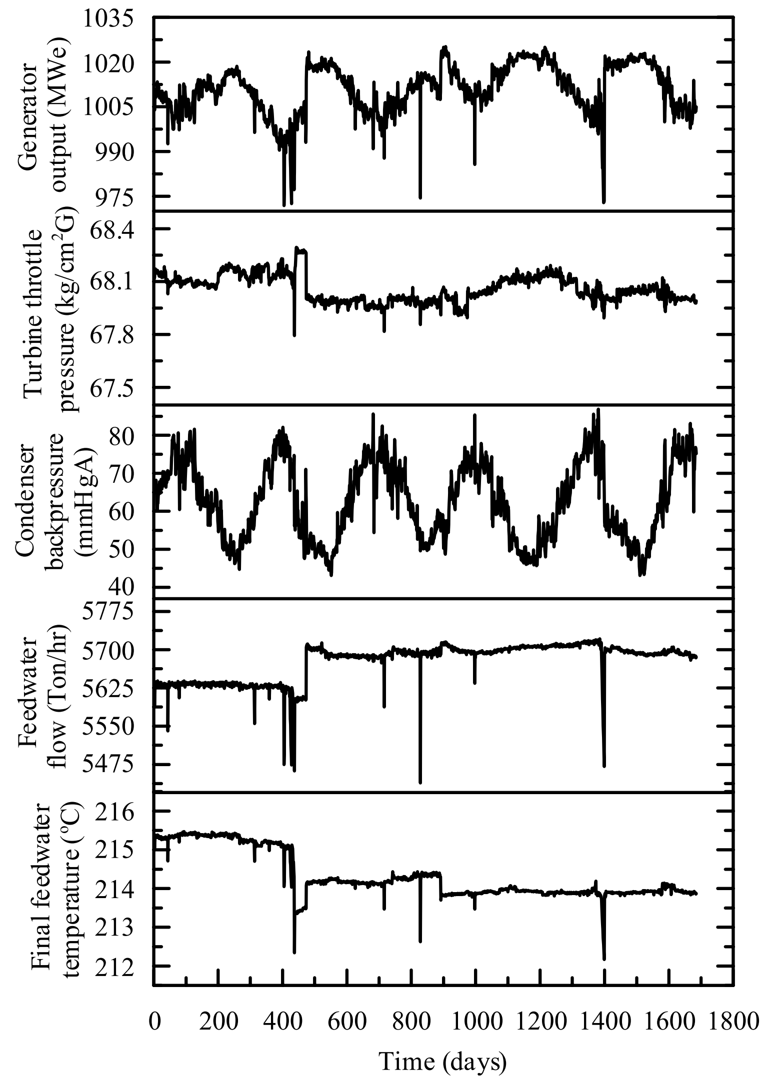
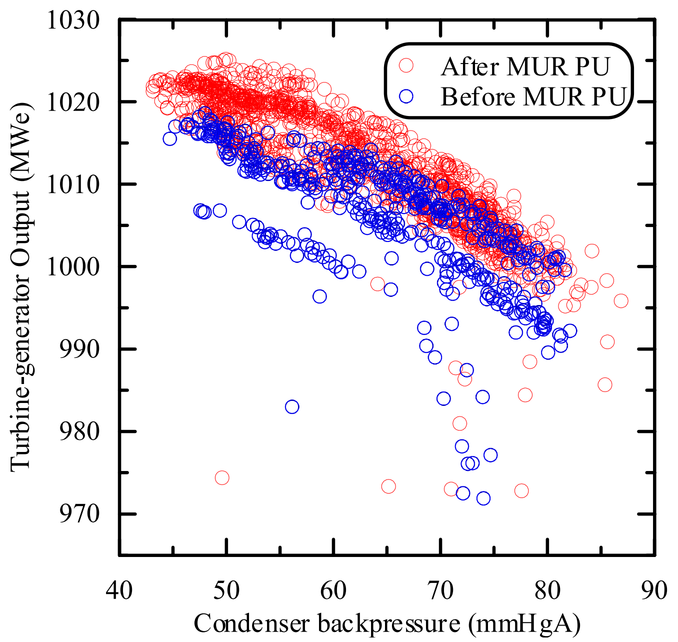


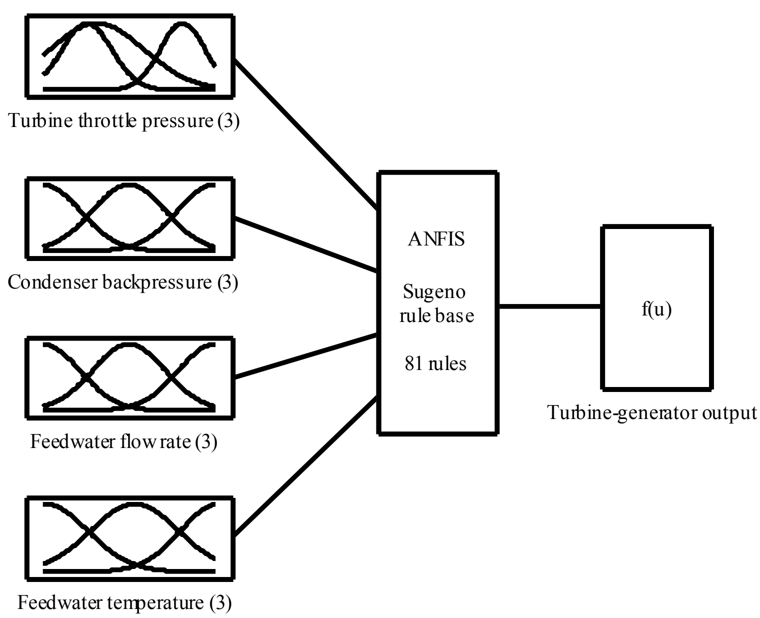
Figure 8. Turbine-generator output versus condenser backpressure.



5.3. ANFIS Structure

To ensure the accuracy of the results and simplify the computation and structure for the ANFIS based turbine cycle model, researchers assembled the membership function of the four inputs in the form of a Gaussian function with three membership functions for each input, as shown in the brackets following the name of the input variable (Figure 9). The 12 Gaussian membership functions were initialized by grid-partitioning of the training data and each rule has unity weight. The rule base comprises 81 Sugeno-type rules. The training process changes the parameters of the initial membership functions to optimize the representation of the input and output mappings. The resulting 81 fuzzy if-then rules are listed in the Appendix.

Figure 9. Structure of the ANFIS based turbine cycle model for the Kuosheng NPP.



5.4. Training, Checking, and Testing Data

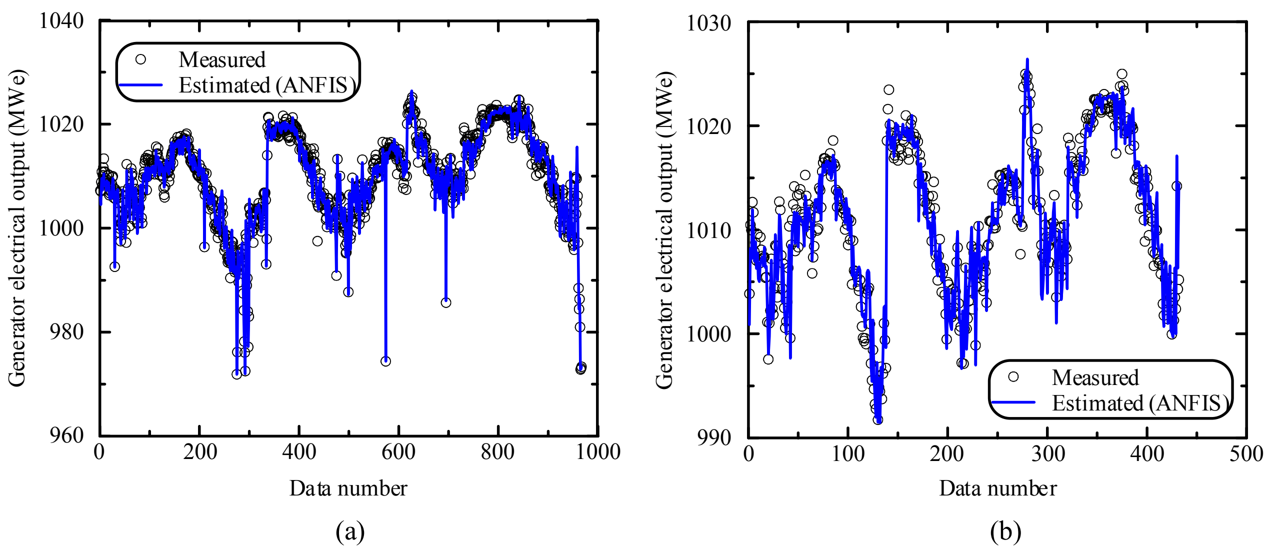
The operating data for the five parameters (including the 10 signal points) listed in Table 1 were collected from Unit 1 of the Kuosheng NPP Plant from May 2006 to August 2011, as shown in Figure 7. The 1685 available input patterns were subdivided into three subsets for neuro-fuzzy modeling: a training dataset of 967 patterns, a checking dataset of 432 patterns, and a testing dataset of 286 patterns. Each pattern contained values related to turbine throttle pressure, condenser backpressure, feedwater flow rate, feedwater temperature, and the target value (turbine-generator electrical output).

The training dataset was used for training the ANFIS in the input-output mappings. The checking dataset was used together with the training dataset in the learning process to prevent model overfitting. The testing dataset was used to validate the model and determine whether the developed ANFIS based turbine cycle model could be generalized. There is no overlap or duplicate data samples among these three datasets, *i.e.*, no data sample can exist in more than one dataset. The data selected for the training process have a significant impact on the predictive power of the fuzzy inference model developed by ANFIS; the modeling accuracy will be high if the training data presented to ANFIS for training is fully representative of the features of the data space being modeled. Therefore, the data windows for the training data should be kept as wide as possible. For training purposes, the authors selected the data that covered the entire range of values for each parameter so that it fully represents all the operating conditions.

6. Results

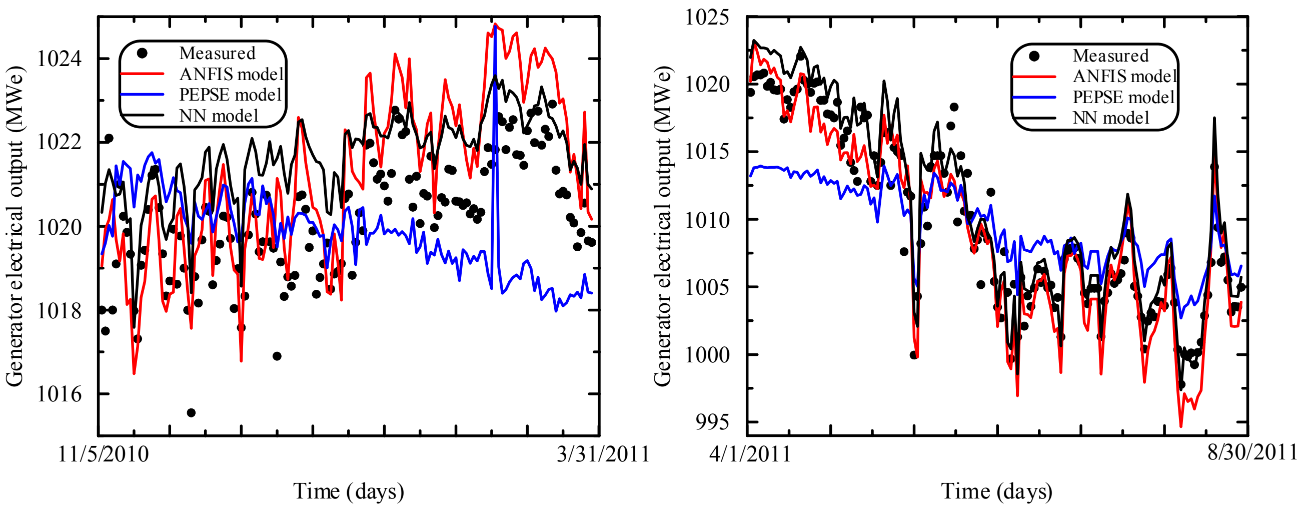
MATLAB software package with its associated Fuzzy Logic Toolbox and Neural Network Toolbox were used to develop the ANFIS based turbine cycle model. Figure 10(a), the training results, shows that the ANFIS based turbine model performed well with a satisfactory accuracy. Figure 10(b), the validated results between the measured and estimated output of the ANFIS model, shows that the output of the ANFIS model accurately matches the measured generator output well.

Figure 10. (**a**) Training results for the ANFIS based turbine cycle model; (**b**) Comparison between the measured values and the validated results of the ANFIS model.



After the ANFIS model had been trained and validated, it was used to predict turbine-generator output. In addition, we adopted a widely used commercial software program, PEPSE®, to develop the thermodynamic turbine cycle model for Unit 1 of the Kuosheng NPP in order to compare the performance of the ANFIS based turbine cycle model. Furthermore, the artificial neural network has also being tried to model the turbine cycle model. In this study, a feedforward backpropagation neural network was adopted due to its simplicity and maturity. The number of hidden layers was set to 1 to simplify the model building process since a one hidden layer network is typically sufficient for modeling industrial tasks [23]. A satisfactory result has been met with 20 hidden layer neurons and hyperbolic tangent sigmoid transfer function is used. The ANN model was trained with the Levenberg-Marquardt algorithm as a training algorithm. ANFIS and ANN design parameters are optimized in terms of the number of training epochs for both models, the number of MFs of ANFIS per input, the number of ANN neurons, and the type of transfer function for ANN. The values of these parameters are obtained by over 50 runs. A comparison of measured and estimated output of the turbine-generator among the ANFIS, ANN, and PEPSE® models under winter conditions with lower condenser backpressure that varied between 43.1 and 61.8 inHgA is shown in Figure 11(a) (Case 1). The results show that the ANFIS based turbine cycle model can be used to accurately estimate turbine-generator output, providing accurate estimation and clearly defined trends.

Figure 11. (**a**) Estimated results under winter operating conditions (Case 1); (**b**) Estimated results with wider variations in condenser backpressure (Case 2).



(a) (b)

Figure 11(b) (Case 2) compares the performance of the ANFIS, ANN, and PEPSE® models under wider variations in condenser backpressure (varied between 47.9 and 83.1 inHgA) from April 2011 to August 2011 covering the spring and summer operating conditions. This demonstrates that the ANFIS based turbine cycle model provides greater prediction accuracy than the ANN and PEPSE® models. From Figure 11 and the results, it is clear that the ANFIS based turbine cycle model is capable of accurately predicting the turbine-generator output with a large change in condenser backpressure. Figures 10 and 11 show that the ANFIS based turbine cycle model is more reliable than the ANN and PEPSE® models in predicting turbine-generator output, providing accurate estimation and clearly defined trends. In this study, model performance is measured by using the mean relative error (MRE) and root mean square error (RMSE), defined as:

$$MRE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - y_i^*}{y_i}\right| \tag{8}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - y_i^*\right)^2} \tag{9}$$

where y_i and y_i^* represent the measured and estimated electrical output of the turbine-generator, respectively, and n is the number of the values provided. The results of MRE and RMSE for ANFIS, ANN and PEPSE® models are summarized in Table 2, which shows that the proposed ANFIS based turbine cycle model can predict turbine-generator output very accurately. The values shown in Table 2 are the best one in a set of experimentations obtained from different approaches. In addition, the minimum error and mean absolute error for different approaches are also listed in Table 2.

Table 2. Statistical parameters for different models.

Models	Case 1				Case 2			
	MRE (%)	**RMSE (MWe)**	**Mean Absolute Error (MWe)**	**Minimum Error (MWe)**	**MRE (%)**	**RMSE (MWe)**	**Mean Absolute Error (MWe)**	**Minimum Error (MWe)**
ANFIS	0.129	1.521	1.312	0.015	0.158	1.995	1.596	0.029
ANN	0.145	1.688	1.482	0.027	0.165	2.095	1.665	0.002
PEPSE	0.162	1.959	1.651	0.001	0.328	3.813	3.319	0.002

Nuclear power plants are almost always operated at full load to supply the demanded load and to operate economically. Under this condition, we are unable to obtain partial load data, *i.e.*, below the 95% load level. Therefore, the present ANFIS based turbine cycle model was used to estimate the turbine-generator output above the 95% load level with higher accuracy. Fossil-fired, combined cycle, or power generation plants of other types may continually operate at partial load conditions. We anticipate that the ANFIS based technique may be used to develop the turbine cycle model for those power plants to predict the turbine-generator output as a thermal efficiency tool.

7. Conclusions

This paper presents an ANFIS based turbine cycle model for the Kuosheng NPP. The effectiveness of the proposed ANFIS based turbine cycle model to predict the turbine-generator output is demonstrated using plant operating data obtained from Unit 1 of the Kuosheng NPP. The plant operating data was verified using a linear regression model with a corresponding 95% confidence interval. The key variables that strongly affect the turbine-generator output are selected as the input variables of the ANFIS model, which is then used to predict the turbine-generator output above the 95% load level under normal operating conditions.

A comparison of measured data with estimated results shows that the ANFIS based turbine cycle model is reliable and effective. The results also show that this turbine cycle model can be used to accurately predict turbine-generator output. In addition, by comparing turbine-generator output from the ANFIS based turbine cycle model with that from a commercial simulation tool, the effectiveness and accuracy of ANFIS based turbine cycle model is validated. Furthermore, the results obtained by the ANFIS model present better performance than the ANN, which has also being tried to model the turbine cycle. The ANFIS based turbine cycle model can be used to predict the turbine-generator output for Kuosheng NPP in practice. The achievement of this study also provides an alternative approach to evaluate the thermal performance of NPPs.

Acknowledgments

The authors would like to present their sincere appreciation to Kuosheng NPP for providing the valuable plant operating data for this study.

Appendix

As suggested by one of the reviewers, to give the readers a concrete idea of the resulting fuzzy inference system, it would be better to list the fuzzy if-then rules explicitly. The final 81 fuzzy if-then rules for ANFIS based turbine cycle model would read as:

1. If (input1 is LOW_1) and (input2 is LOW_2) and (input3 is LOW_3) and (input4 is LOW_4) then (output is out1mf1)
2. If (input1 is LOW_1) and (input2 is LOW_2) and (input3 is LOW_3) and (input4 is $MEDIUM_4$) then (output is out1mf2)
3. If (input1 is LOW_1) and (input2 is LOW_2) and (input3 is LOW_3) and (input4 is $HIGH_4$) then (output is out1mf3)
4. If (input1 is LOW_1) and (input2 is LOW_2) and (input3 is $MEDIUM_3$) and (input4 is LOW_4) then (output is out1mf4)
5. If (input1 is LOW_1) and (input2 is LOW_2) and (input3 is $MEDIUM_3$) and (input4 is $MEDIUM_4$) then (output is out1mf5)
6. If (input1 is LOW_1) and (input2 is LOW_2) and (input3 is $MEDIUM_3$) and (input4 is $HIGH_4$) then (output is out1mf6)
7. If (input1 is LOW_1) and (input2 is LOW_2) and (input3 is $HIGH_3$) and (input4 is LOW_4) then (output is out1mf7)
8. If (input1 is LOW_1) and (input2 is LOW_2) and (input3 is $HIGH_3$) and (input4 is $MEDIUM_4$) then (output is out1mf8)
9. If (input1 is LOW_1) and (input2 is LOW_2) and (input3 is $HIGH_3$) and (input4 is $HIGH_4$) then (output is out1mf9)
10. If (input1 is LOW_1) and (input2 is $MEDIUM_2$) and (input3 is LOW_3) and (input4 is LOW_4) then (output is out1mf10)
11. If (input1 is LOW_1) and (input2 is $MEDIUM_2$) and (input3 is LOW_3) and (input4 is $MEDIUM_4$) then (output is out1mf11)
12. If (input1 is LOW_1) and (input2 is $MEDIUM_2$) and (input3 is LOW_3) and (input4 is $HIGH_4$) then (output is out1mf12)
13. If (input1 is LOW_1) and (input2 is $MEDIUM_2$) and (input3 is $MEDIUM_3$) and (input4 is LOW_4) then (output is out1mf13)
14. If (input1 is LOW_1) and (input2 is $MEDIUM_2$) and (input3 is $MEDIUM_3$) and (input4 is $MEDIUM_4$) then (output is out1mf14)
15. If (input1 is LOW_1) and (input2 is $MEDIUM_2$) and (input3 is $MEDIUM_3$) and (input4 is $HIGH_4$) then (output is out1mf15)
16. If (input1 is LOW_1) and (input2 is $MEDIUM_2$) and (input3 is $HIGH_3$) and (input4 is LOW_4) then (output is out1mf16)
17. If (input1 is LOW_1) and (input2 is $MEDIUM_2$) and (input3 is $HIGH_3$) and (input4 is $MEDIUM_4$) then (output is out1mf17)
18. If (input1 is LOW_1) and (input2 is $MEDIUM_2$) and (input3 is $HIGH_3$) and (input4 is $HIGH_4$) then (output is out1mf18)
19. If (input1 is LOW_1) and (input2 is $HIGH_2$) and (input3 is LOW_3) and (input4 is LOW_4) then (output is out1mf19)
20. If (input1 is LOW_1) and (input2 is $HIGH_2$) and (input3 is LOW_3) and (input4 is $MEDIUM_4$) then (output is out1mf20)
21. If (input1 is LOW_1) and (input2 is $HIGH_2$) and (input3 is LOW_3) and (input4 is $HIGH_4$) then (output is out1mf21)
22. If (input1 is LOW_1) and (input2 is $HIGH_2$) and (input3 is $MEDIUM_3$) and (input4 is LOW_4) then (output is out1mf22)
23. If (input1 is LOW_1) and (input2 is $HIGH_2$) and (input3 is $MEDIUM_3$) and (input4 is $MEDIUM_4$) then (output is out1mf23)
24. If (input1 is LOW_1) and (input2 is $HIGH_2$) and (input3 is $MEDIUM_3$) and (input4 is $HIGH_4$) then (output is out1mf24)
25. If (input1 is LOW_1) and (input2 is $HIGH_2$) and (input3 is $HIGH_3$) and (input4 is LOW_4) then (output is out1mf25)
26. If (input1 is LOW_1) and (input2 is $HIGH_2$) and (input3 is $HIGH_3$) and (input4 is $MEDIUM_4$) then (output is out1mf26)
27. If (input1 is LOW_1) and (input2 is $HIGH_2$) and (input3 is $HIGH_3$) and (input4 is $HIGH_4$) then (output is out1mf27)
28. If (input1 is $MEDIUM_1$) and (input2 is LOW_2) and (input3 is LOW_3) and (input4 is LOW_4) then (output is out1mf28)
29. If (input1 is $MEDIUM_1$) and (input2 is LOW_2) and (input3 is LOW_3) and (input4 is $MEDIUM_4$) then (output is out1mf29)
30. If (input1 is $MEDIUM_1$) and (input2 is LOW_2) and (input3 is LOW_3) and (input4 is $HIGH_4$) then (output is out1mf30)
31. If (input1 is $MEDIUM_1$) and (input2 is LOW_2) and (input3 is $MEDIUM_3$) and (input4 is LOW_4) then (output is out1mf31)
32. If (input1 is $MEDIUM_1$) and (input2 is LOW_2) and (input3 is $MEDIUM_3$) and (input4 is $MEDIUM_4$) then (output is out1mf32)
33. If (input1 is $MEDIUM_1$) and (input2 is LOW_2) and (input3 is $MEDIUM_3$) and (input4 is $HIGH_4$) then (output is out1mf33)
34. If (input1 is $MEDIUM_1$) and (input2 is LOW_2) and (input3 is $HIGH_3$) and (input4 is LOW_4) then (output is out1mf34)
35. If (input1 is $MEDIUM_1$) and (input2 is LOW_2) and (input3 is $HIGH_3$) and (input4 is $MEDIUM_4$) then (output is out1mf35)
36. If (input1 is $MEDIUM_1$) and (input2 is LOW_2) and (input3 is $HIGH_3$) and (input4 is $HIGH_4$) then (output is out1mf36)
37. If (input1 is $MEDIUM_1$) and (input2 is $MEDIUM_2$) and (input3 is LOW_3) and (input4 is LOW_4) then (output is out1mf37)
38. If (input1 is $MEDIUM_1$) and (input2 is $MEDIUM_2$) and (input3 is LOW_3) and (input4 is $MEDIUM_4$) then (output is out1mf38)
39. If (input1 is $MEDIUM_1$) and (input2 is $MEDIUM_2$) and (input3 is LOW_3) and (input4 is $HIGH_4$) then (output is out1mf39)
40. If (input1 is $MEDIUM_1$) and (input2 is $MEDIUM_2$) and (input3 is $MEDIUM_3$) and (input4 is LOW_4) then (output is out1mf40)

41. If (input1 is $MEDIUM_1$) and (input2 is $MEDIUM_2$) and (input3 is $MEDIUM_3$) and (input4 is $MEDIUM_4$) then (output is out1mf41)
42. If (input1 is $MEDIUM_1$) and (input2 is $MEDIUM_2$) and (input3 is $MEDIUM_3$) and (input4 is $HIGH_4$) then (output is out1mf42)
43. If (input1 is $MEDIUM_1$) and (input2 is $MEDIUM_2$) and (input3 is $HIGH_3$) and (input4 is LOW_4) then (output is out1mf43)
44. If (input1 is $MEDIUM_1$) and (input2 is $MEDIUM_2$) and (input3 is $HIGH_3$) and (input4 is $MEDIUM_4$) then (output is out1mf44)
45. If (input1 is $MEDIUM_1$) and (input2 is $MEDIUM_2$) and (input3 is $HIGH_3$) and (input4 is $HIGH_4$) then (output is out1mf45)
46. If (input1 is $MEDIUM_1$) and (input2 is $HIGH_2$) and (input3 is LOW_3) and (input4 is LOW_4) then (output is out1mf46)
47. If (input1 is $MEDIUM_1$) and (input2 is $HIGH_2$) and (input3 is LOW_3) and (input4 is $MEDIUM_4$) then (output is out1mf47)
48. If (input1 is $MEDIUM_1$) and (input2 is $HIGH_2$) and (input3 is LOW_3) and (input4 is $HIGH_4$) then (output is out1mf48)
49. If (input1 is $MEDIUM_1$) and (input2 is $HIGH_2$) and (input3 is $MEDIUM_3$) and (input4 is LOW_4) then (output is out1mf49)
50. If (input1 is $MEDIUM_1$) and (input2 is $HIGH_2$) and (input3 is $MEDIUM_3$) and (input4 is $MEDIUM_4$) then (output is out1mf50)
51. If (input1 is $MEDIUM_1$) and (input2 is $HIGH_2$) and (input3 is $MEDIUM_3$) and (input4 is $HIGH_4$) then (output is out1mf51)
52. If (input1 is $MEDIUM_1$) and (input2 is $HIGH_2$) and (input3 is $HIGH_3$) and (input4 is LOW_4) then (output is out1mf52)
53. If (input1 is $MEDIUM_1$) and (input2 is $HIGH_2$) and (input3 is $HIGH_3$) and (input4 is $MEDIUM_4$) then (output is out1mf53)
54. If (input1 is $MEDIUM_1$) and (input2 is $HIGH_2$) and (input3 is $HIGH_3$) and (input4 is $HIGH_4$) then (output is out1mf54)
55. If (input1 is $HIGH_1$) and (input2 is LOW_2) and (input3 is LOW_3) and (input4 is LOW_4) then (output is out1mf55)
56. If (input1 is $HIGH_1$) and (input2 is LOW_2) and (input3 is LOW_3) and (input4 is $MEDIUM_4$) then (output is out1mf56)
57. If (input1 is $HIGH_1$) and (input2 is LOW_2) and (input3 is LOW_3) and (input4 is $HIGH_4$) then (output is out1mf57)
58. If (input1 is $HIGH_1$) and (input2 is LOW_2) and (input3 is $MEDIUM_3$) and (input4 is LOW_4) then (output is out1mf58)
59. If (input1 is $HIGH_1$) and (input2 is LOW_2) and (input3 is $MEDIUM_3$) and (input4 is $MEDIUM_4$) then (output is out1mf59)
60. If (input1 is $HIGH_1$) and (input2 is LOW_2) and (input3 is $MEDIUM_3$) and (input4 is $HIGH_4$) then (output is out1mf60)
61. If (input1 is $HIGH_1$) and (input2 is LOW_2) and (input3 is $HIGH_3$) and (input4 is LOW_4) then (output is out1mf61)
62. If (input1 is $HIGH_1$) and (input2 is LOW_2) and (input3 is $HIGH_3$) and (input4 is $MEDIUM_4$) then (output is out1mf62)
63. If (input1 is $HIGH_1$) and (input2 is LOW_2) and (input3 is $HIGH_3$) and (input4 is $HIGH_4$) then (output is out1mf63)
64. If (input1 is $HIGH_1$) and (input2 is $MEDIUM_2$) and (input3 is LOW_3) and (input4 is LOW_4) then (output is out1mf64)
65. If (input1 is $HIGH_1$) and (input2 is $MEDIUM_2$) and (input3 is LOW_3) and (input4 is $MEDIUM_4$) then (output is out1mf65)
66. If (input1 is $HIGH_1$) and (input2 is $MEDIUM_2$) and (input3 is LOW_3) and (input4 is $HIGH_4$) then (output is out1mf66)
67. If (input1 is $HIGH_1$) and (input2 is $MEDIUM_2$) and (input3 is $MEDIUM_3$) and (input4 is LOW_4) then (output is out1mf67)
68. If (input1 is $HIGH_1$) and (input2 is $MEDIUM_2$) and (input3 is $MEDIUM_3$) and (input4 is $MEDIUM_4$) then (output is out1mf68)
69. If (input1 is $HIGH_1$) and (input2 is $MEDIUM_2$) and (input3 is $MEDIUM_3$) and (input4 is $HIGH_4$) then (output is out1mf69)
70. If (input1 is $HIGH_1$) and (input2 is $MEDIUM_2$) and (input3 is $HIGH_3$) and (input4 is LOW_4) then (output is out1mf70)
71. If (input1 is $HIGH_1$) and (input2 is $MEDIUM_2$) and (input3 is $HIGH_3$) and (input4 is $MEDIUM_4$) then (output is out1mf71)
72. If (input1 is $HIGH_1$) and (input2 is $MEDIUM_2$) and (input3 is $HIGH_3$) and (input4 is $HIGH_4$) then (output is out1mf72)
73. If (input1 is $HIGH_1$) and (input2 is $HIGH_2$) and (input3 is LOW_3) and (input4 is LOW_4) then (output is out1mf73)
74. If (input1 is $HIGH_1$) and (input2 is $HIGH_2$) and (input3 is LOW_3) and (input4 is $MEDIUM_4$) then (output is out1mf74)
75. If (input1 is $HIGH_1$) and (input2 is $HIGH_2$) and (input3 is LOW_3) and (input4 is $HIGH_4$) then (output is out1mf75)
76. If (input1 is $HIGH_1$) and (input2 is HIGH) and (input3 is $MEDIUM_3$) and (input4 is LOW_4) then (output is out1mf76)
77. If (input1 is $HIGH_1$) and (input2 is HIGH) and (input3 is $MEDIUM_3$) and (input4 is $MEDIUM_4$) then (output is out1mf77)
78. If (input1 is $HIGH_1$) and (input2 is HIGH) and (input3 is $MEDIUM_3$) and (input4 is $HIGH_4$) then (output is out1mf78)
79. If (input1 is $HIGH_1$) and (input2 is HIGH) and (input3 is $HIGH_3$) and (input4 is LOW_4) then (output is out1mf79)
80. If (input1 is $HIGH_1$) and (input2 is HIGH) and (input3 is $HIGH_3$) and (input4 is $MEDIUM_4$) then (output is out1mf80)
81. If (input1 is $HIGH_1$) and (input2 is HIGH) and (input3 is $HIGH_3$) and (input4 is $HIGH_4$) then (output is out1mf81)

References

1. PEPSE®. *User Input Description*; Scientech Inc.: Idaho Falls, ID, USA, 2001.
2. Chang, C.J.; Chan, Y.K.; Tsai, C.H.; Lee, D.W. Development of a Thermal Efficiency Monitoring System for Kuosheng Nuclear Power Plant. In *Proceedings of the 6th International Conference on Nuclear Thermal-Hydraulics, Operations and Safety*, Nara, Japan, 4–8 October 2004.
3. Heo, G.; Chang, S.H.; Choi, S.S. Development of a need-oriented steam turbine cycle simulation toolbox. *IEEE Trans. Energy Convers.* **2005**, *20*, 859–869.
4. Kim S.; Choi, K. PERUPS (PERformance UPgrade System) for on-line performance analysis of a nuclear power plant turbine cycle. *Nuclear Eng. Technol.* **2005**, *37*, 167–176.
5. Nakao, Y.; Koda, E.; Takahashi, T. Development of general-purpose software to analyze the statistic thermal characteristic of nuclear power plant. *J. Power Energy Syst.* **2009**, *3*, 2–11.
6. Ubeyli, E.D. Automatic detection of electroencephalographic changes using adaptive neuro-fuzzy inference system employing Lyapunov exponents. *Expert Syst. Appl.* **2009**, *36*, 9031–9038.
7. Jang, J.S. ANFIS: Adaptive network-based fuzzy inference system. *IEEE Trans. Syst. Man Cybern.* **1993**, *23*, 665–685.

8. Jang, J.S.; Sun, C.T.; Mizutani, E. *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*; Prentice-Hall: Upper Saddle River, NJ, USA, 1997.
9. Chen, C.S.; Lai, Y.H. Rotor fault diagnosis system based on individual neural networks and fuzzy synthesized engine. *J. Chin. Inst. Eng.* **2010**, *33*, 975–986.
10. Hasiloglu, A.; Yilmaz, M.; Comakli, O.; Ekmekci, I. Adaptive neuro-fuzzy modeling of transient heat transfer in circular duct air flow. *Int. J. Therm. Sci*. **2004**, *43*, 1075–1090.
11. Awadallah, M.A.; Morcos, M.M.; Gopalakrishnan, S.; Nehl, T.W. A neuro-fuzzy approach to automatic diagnosis and location of stator inter-turn faults in CSI-fed PM brushless DC motors. *IEEE Trans. Energy Convers.* **2005**, *20*, 253–259.
12. Guimaraes, A.C.F.; Lapa, C.M.F. Adaptive fuzzy system for fuel rod cladding failure in nuclear power plant. *Ann. Nuclear Energy*. **2007**, *34*, 233–240.
13. Mellit, A.; Kalogirou, S.A. ANFIS-based modelling for photovoltaic power supply system: A case study. *Renew. Energy* **2011**, *36*, 250–258.
14. Guo, Z.; Uhrig, R.E. Nuclear power plant performance study by using neural networks. *IEEE Trans. Nuclear Sci.* **1992**, 39, 915–918.
15. Taiwan Power Company. Final Safety Analysis Report; Kuosheng Nuclear Power Station Units 1 & 2, Amendment No. 18, 2010.
16. Chang, C.J.; Kao, L.; Wang, T.; Lin, C.P.; Hong, L.C.; Chen, S.R. Measurement Uncertainty Recapture Power Uprates in Taiwan. In *Proceedings of the 8th International Conference on Nuclear Thermal-Hydraulics, Operations and Safety*, Shanghai, China, 10–14 October 2010.
17. Siemens AG Power Generation. *100% Load Heat Balance Diagram*; Siemens AG Power Generation Inc.: Orlando, FL, USA, 2006.
18. Takagi, T.; Sugeno, M. Fuzzy identification of systems and its applications to modeling and control. *IEEE Trans. Syst. Man Cybern.* **1985**, *15*, 116–132.
19. Pai, P.F.; Hong, W.C. An improved neural network model in forecasting arrivals. *Ann. Tourism Res.* **2005**, *32*, 1138–1141.
20. Kandananond, K. Forecasting electricity demand in Thailand with an artificial neural network approach. *Energies* **2011**, *4*, 1246–1257.
21. Amjady, N.; Keynia, F. A new neural network approach to short term load forecasting of electrical power systems. *Energies* **2011**, *4*, 488–503.
22. Chan, A.M.C.; Ahluwalia, A.K. Feedwater Flow Measurement in U.S. Nuclear Power Generation Station; EPRI TR-101388, Electric Power Research Institute: Charlotte, NC, USA, 1992.
23. Zhou, Q.; Wu, Y.; Chan, C.W.; Tontiwachwuthikul, P. Modeling of the carbon dioxide capture process system using machine intelligence approaches. *Appl. Artif. Intell.* **2011**, *24*, 673–685.