Available online at www.sciencedirect.com
SCIENCE DIRECT
Annals of Nuclear Energy 33 (2006) 594–608
annals of NUCLEAR ENERGY
www.elsevier.com/locate/anucene

Neural network correlation for power peak factor estimation

Rose Mary G.P. Souza [a], João M.L. Moreira [b,*]

[a] Centro de Desenvolvimento da Tecnologia Nuclear – CDTN-CNEN/BH Caixa Postal 941, 30123-970 Belo Horizonte, MG, Brazil
[b] Centro Tecnológico da Marinha em São Paulo – CTMSP, Av. Prof. Lineu Prestes, 2468, 05508-900 Cidade Universitária, São Paulo, SP, Brazil

Received 6 May 2005; received in revised form 6 February 2006; accepted 6 February 2006
Available online 19 April 2006

Abstract

This paper proposes a method, based on the artificial neural network technique, to predict accurately and in real time the power peak factor in a form that can be implemented in reactor protection systems. The neural network inputs are the position of control rods and signals of ex-core detectors. The data used to train the networks were obtained in the IPEN/MB-01 zero-power reactor from especially designed experiments. The relative error for the power peak factor estimation ranged from 0.19% to 0.67%, an accuracy better than what is obtained performing a power density distribution map with in-core detectors. The networks were able to identify classes and interpolate the power peak factor values. It was observed that the positions of control rods bear the detailed and localised information about the power density distribution, and that the axial and the quadrant power differences, obtained from signals of ex-core detectors, describe its global variations in the axial and radial directions. In the power reactor environment, the neural networks would require in the input vector the position of control rods, and axial and quadrant power differences. The results showed that the RBF networks produced slightly better results than the MLP networks, but, for practical purposes, both can be considered of similar accuracy. The results indicate that they may allow decreasing the power peak factor safety margin by as much as 5%.

1. Introduction

The power density peak factor is an important operation and safety parameter in PWRs, and is usually monitored continuously and in real time by the reactor protection system (USNRC, 1995). The power density distribution cannot be directly measured and is usually described in terms of power peak factors, axial and quadrant power differences, or other schemes. All these parameters are usually inferred from primary system variables such as the signals from ex-core detectors, position of control rods and, in some cases, from in-core detectors (USNRC, 1995; Souza and Moreira, 2006). The current real time schemes to estimate the power density distribution and the departure from nucleate boiling ratio (DNBR) have uncertainties as high as 20% and, therefore, it is important to find alternatives to improve their accuracy.

A previous article reviewed the recent literature related to the problem of obtaining information about the power density distribution through artificial neural networks for protection system applications (Souza and Moreira, 2006). Artificial neural networks allow modelling complex systems without requiring an explicit knowledge or formulation of the relationship that exist among the variables, and constitute an alternative to structured models or empirical correlations (Haykin, 1999; Tsoukalas and Uhrig, 1997). The literature presents a wide variety of applications of neural networks to nuclear engineering such as for plant control (Boroushaki et al., 2004; Lin and Shen, 2000; Na and Upadhyaya, 1998), fuel cycle optimisation (Faria and Pereira, 2003), signal validation (Ikonomopoulos and Van der Hagen, 1997), systems diagnosis (Nabeshima et al., 1998), and for improving the accuracy of measured parameters (Kim et al., 1993). Specifically to our problem, there are applications for estab-

* Corresponding author. Tel.: +55 11 3817 7430; fax: +55 11 3817 7412.
E-mail address: jmoreira@ctmsp.mar.mil.br (J.M.L. Moreira).

0306-4549/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.anucene.2006.02.007

lishing a correlation among primary system variables to obtain information about the power density distribution for improving the estimate of the minimum DNBR (Guanghui et al., 2003; Lee and Chang, 2003; Seon et al., 2002; Na et al., 2004; Kim and Chang, 1997; Kim and Lee, 1993; Seong et al., 2002; Su et al., 2002). They attempted, using more sophisticated algorithms based on the neural network technique, to reduce the reactor thermal margins.

Kim and Chang (1997) developed neural networks to estimate the DNBR using as input core thermal-hydraulic variables and power distribution information (radial peak factor and axial power density). For constant and known axial power density distribution they predicted DNBR with 3.5% accuracy, and for those cases in which the power density distribution was unknown and could change, they predicted DNBR with 11.5% accuracy. They attempted to describe the axial power density distribution with 3, 4, 5 and 20 parameters, and the best result was obtained with 4 parameters. Na et al. (2004) developed fuzzy neural networks for estimating the DNBR distribution in the hot channel. As input, they used thermal-hydraulic state variables, simulated axial core power differences, and position of control rods to describe the radial power density distribution in the core. The most influential input variable was the axial power difference, and the minimum DNBR was estimated with 13.6% accuracy.

Lee and Chang (2003) used radial basis function neural networks to estimate the DNBR in which the power distribution was estimated from the signals of ex-core detectors and position of control rods. For problems with varying power density distribution the DNBR was estimated with 10% accuracy. Seon et al. (2002) attempted, with artificial neural networks, to obtain the axial power density distribution in 20 nodes from signals of ex-core detectors. It was used simulated data to represent the ex-core detector signals and the actual power density distribution. They remarked that for an actual application to power reactors it would be necessary accurate three-dimensional power density distribution data to reduce the uncertainty margins during the generalisation process.

Some interesting points can be drawn from the literature (Guanghui et al., 2003; Lee and Chang, 2003; Seon et al., 2002; Na et al., 2004; Kim and Chang, 1997). The common approach considered was to use one neural network to bear all the knowledge involved in the problem, namely, how to determine the power density distribution and how to estimate the critical heat flux. The neural networks had large input vectors including variables to describe the coolant thermal-hydraulic properties and the power density distribution. Most of the applications considered one hidden layer sufficient to solve the problem, and its number of neurons varied according to the number of inputs and output considered in the model. The number of training sets varied from 50 to 200, and the networks with smaller number of neurons appeared better trained and provided the best results. The training data of several applications consisted of many simulated parameters, and it was remarked that for power reactors it would be necessary to obtain accurate values for the inputs and outputs in order to attain a reduction in the thermal margins (Guanghui et al., 2003; Lee and Chang, 2003; Na et al., 2004; Kim and Chang, 1997; Seon et al., 2002).

In this work, we sought to establish a correlation to obtain the power peak factor using experimental data especially obtained for this purpose (Souza and Moreira, 2006). A series of experiments were performed in the IPEN/MB-01 nuclear reactor to provide actual measured data suitable for artificial neural network applications. In the data set, the power peak factor was correlated to signals of ex-core detector and position of control rods (Souza and Moreira, 2006). We considered two different types of neural networks to model the problem, namely, the multilayer perceptron, MLP, and the radial basis function networks, RBF. The main difference between them is that, for estimating the output result, the MLP networks map the entire input space, while the RBF networks map only part of it identifying clusters in the input space (Haykin, 1999; Braga et al., 2000). We also considered input vectors with different reactor state variables. Section 2 summarises the experimental data used to design, train and validate the neural networks, and Section 3 introduces the RBF and MLP neural networks. Section 4 presents the several artificial neural networks developed, and Section 5, the results and the discussions. Finally, Section 6 presents the conclusions.

2. The training data to estimate the power peak factor

The data used to train the neural networks were obtained in especially designed experiments correlating the power density peak factor to signals of ex-core detectors and position of control rods (Souza and Moreira, 2006). The experiments were performed in the IPEN/MB-01 reactor, a zero-power light water reactor designed for PWR core studies (Maiorino et al., 1989). The experiments were based on the fact that the control rods usually determine the shape of the power distribution in the reactor core. The different power density distributions in the IPEN/MB-01 reactor core were obtained by positioning the control rods in different heights and establishing critical states. The procedure adopted to establish the several states was based on moving the control rods in pre-determined patterns: insertion of diagonal control rods, insertion of parallel control rods, and insertion of two control rods together. This procedure allowed the data to be divided into 10 different classes, according to the control rod movement pattern, as shown in Table 1. In order to avoid data extrapolation, the states were such that they encompassed the maximum control rod motion possible for each class. Altogether, 56 different critical states were obtained constituting in 56 examples relating control rod position and ex-core detector signals with the power peak factors, PF.

Table 1 shows the following data: the position of four control rods, BS1, BS2, BC1 and BC2; the axial power differences obtained from ex-core detectors in the north and

Table 1
Data for training the neural networks: BS1, BS2, BC1 and BC2, APD^N, APD^W, QPD_B, QPD_T, and PF values

State/class	BS1 (%)	BS2 (%)	BC1 (%)	BC2 (%)	APD^N (%)	APD^W (%)	QPD_B (%)	QPD_T (%)	PF
1/[a]	67.10	67.10	67.10	67.10	17.10	17.07	−0.01	−0.05	2.247
2/1	59.82	77.10	67.10	67.10	16.01	17.37	−1.43	−0.03	2.261
3/1	55.86	87.10	67.10	67.10	16.16	17.17	−1.33	−0.29	2.258
4/1	53.66	100.00	67.10	67.10	16.53	17.86	−1.06	0.31	2.228
5/1	77.10	59.81	67.10	67.10	15.28	17.01	−1.42	0.35	2.261
6/1	87.10	55.92	67.10	67.10	14.46	16.22	−1.03	0.77	2.258
7/1	100.00	53.71	67.10	67.10	13.94	15.43	−1.14	0.38	2.228
8/2	67.10	67.10	77.10	60.00	15.55	18.68	0.58	3.80	2.259
9/2	67.10	67.10	87.10	56.30	14.53	18.37	1.85	5.79	2.249
10/2	67.10	67.10	100.00	54.33	14.04	17.84	3.49	7.37	2.237
11/2	67.10	67.10	59.81	77.10	17.38	17.02	−3.49	−3.86	2.259
12/2	67.10	67.10	55.97	87.10	19.05	15.83	−3.20	−6.51	2.249
13/2	67.10	67.10	53.88	100.00	18.65	15.35	−4.31	−7.70	2.237
14/3	67.10	77.10	67.10	59.13	18.28	18.36	1.09	1.17	2.243
15/3	67.10	87.10	67.10	55.08	17.90	18.00	1.95	2.05	2.228
16/3	67.10	100.00	67.10	52.85	16.82	17.45	1.79	2.43	2.202
17/3	67.10	61.01	67.10	77.10	18.30	16.88	−1.35	−2.82	2.247
18/3	67.10	57.87	67.10	87.10	16.88	15.82	−3.08	−4.17	2.243
19/3	67.10	56.18	67.10	100.00	16.98	15.49	−3.26	−4.80	2.221
20/4	61.00	67.10	67.10	77.10	18.21	17.19	−1.98	−3.02	2.247
21/4	57.87	67.10	67.10	87.10	18.03	16.52	−2.45	−4.01	2.242
22/4	56.17	67.10	67.10	100.00	18.13	15.94	−2.89	−5.14	2.220
23/4	77.10	67.10	67.10	59.18	17.05	17.33	0.75	1.03	2.243
24/4	87.10	67.10	67.10	55.01	15.27	16.06	2.45	3.27	2.228
25/4	100.00	67.10	67.10	52.79	15.17	15.12	3.64	3.59	2.202
26/5	77.10	67.10	58.96	67.10	16.86	16.37	−0.44	−0.95	2.243
27/5	87.10	67.10	54.91	67.10	16.08	15.36	−1.28	−2.01	2.228
28/5	100.00	67.10	52.61	67.10	14.97	14.36	−1.74	−2.37	2.202
29/5	61.02	67.10	77.10	67.10	16.87	17.51	1.53	2.19	2.247
30/5	57.94	67.10	87.10	67.10	15.62	17.22	2.06	3.70	2.243
31/5	56.33	67.10	100.00	67.10	15.44	17.30	3.05	4.97	2.221
32/6	67.10	61.05	77.10	67.10	16.72	17.19	1.34	1.82	2.247
33/6	67.10	57.99	87.10	67.10	15.81	16.51	2.53	3.25	2.242
34/6	67.10	56.30	100.00	67.10	14.82	16.38	2.92	4.51	2.220
35/6	67.10	77.10	58.97	67.10	17.97	17.35	−0.91	−1.56	2.243
36/6	67.10	87.10	54.92	67.10	17.82	16.69	−1.85	−3.01	2.228
37/6	67.10	100.00	52.68	67.10	17.52	16.63	−2.00	−2.92	2.202
38/7	77.10	59.86	77.10	59.86	16.03	16.52	0.04	0.54	2.247
39/7	87.10	55.80	87.10	55.80	13.68	15.62	1.14	3.12	2.227
40/7	100.00	53.30	100.00	53.30	11.90	14.38	2.38	4.90	2.178
41/7	59.75	77.10	59.75	77.10	17.96	16.41	−5.04	−6.63	2.247
42/7	55.62	87.10	55.62	87.10	17.73	15.79	−6.93	−8.91	2.227
43/7	53.05	100.00	53.05	100.00	17.48	14.78	−7.82	−10.57	2.178
44/8	59.85	77.10	77.10	59.85	16.85	17.95	−0.50	0.65	2.248
45/8	55.75	87.10	87.10	55.75	15.57	17.92	1.03	3.45	2.228
46/8	53.23	100.00	100.00	53.23	14.10	17.78	5.47	9.23	2.179
47/8	77.10	59.77	59.77	77.10	17.20	15.88	−1.84	−3.19	2.248
48/8	87.10	55.63	55.63	87.10	15.76	14.25	−3.41	−4.95	2.228
49/8	100.00	53.05	53.05	100.00	14.52	12.78	−4.49	−6.26	2.179
50/9	61.22	61.22	77.10	77.10	17.04	17.16	0.41	0.53	2.215
51/9	58.27	58.27	87.10	87.10	16.31	16.20	0.76	0.64	2.181
52/9	56.66	56.66	100.00	100.00	14.96	15.50	0.45	0.99	2.142
53/9	77.10	77.10	59.35	59.35	17.15	17.37	0.45	0.66	2.199
54/9	87.10	87.10	55.50	55.50	15.70	15.77	0.73	0.80	2.151
55/9	100.00	100.00	53.28	53.28	14.54	14.93	0.22	0.63	2.088
56/10	64.30	77.10	64.30	64.30	17.93	17.76	0.88	0.71	2.248

The uncertainty in the position of control rods is ±0.01%; the uncertainties in the APDs and QPDs are smaller than 0.32%; the uncertainty in the power peak factor is smaller than 10^{-4}.

[a] State 1 belongs to classes 1–9.

west sides of the reactor core, APD^N and APD^W, respectively; the quadrant power differences from detectors located in the bottom part, QPD_B, and top part QPD_T; and finally the power peak factors values, PF. Except for the power peak factor, all other variables were measured in the IPEN/MB-01 reactor. The power peak factor for

each of the 56 states was obtained from three-dimensional, four-energy groups, pin by pin, core calculations using the CITATION code (Fowler et al., 1971). The cross-sections were generated with the unit cell HAMMER/TECHNION code (Barhen et al., 1978).[1]

The footnotes in Table 1 present the average uncertainty for each of the parameters in their respective units (Souza and Moreira, 2006). The power peak factor were calculated with the CITATION code with convergence criteria of 10^{-4} indicating the accuracy of its first three decimal points.

3. Artificial neural networks

Artificial neural networks can be defined as a parallel distributed processor consisting of a great number of processing elements, the neurons, connected to each other with different connection strengths. The strength of a connection between neurons is called weight. A feed-forward multilayered network consists of a layer of input, a layer of output neurons, and one or more hidden layers of neurons. Fig. 1 shows schematically the architecture of a typical artificial neural network with one hidden layer, where "x_m", $m = 1, \ldots, M$ represent the input, "φ_n", $n = 1, \ldots, N$, represent non-linear functions associated to each neuron and called activation functions, and "y", is the output. The summation signal above the output neuron means that its result is the summation of the contributions of all neurons in the previous layer.

In artificial neural networks, the knowledge lies in the interconnection weights between the neurons. In the learning process of the neural network technique, representative examples of the knowledge are interactively presented to the network, so that it can integrate this knowledge within its structure (the weights).

As mentioned before, the data presented in Table 1 could be clustered in classes of characteristic behaviour. This fact indicated that radial basis function (RBF) neural networks could be more suitable to develop the model for obtaining the power peak factor (Haykin, 1999; Braga et al., 2000). Networks that use linear discriminant techniques, such as the multilayer perceptrons (MLP) are global in nature and map the entire input space to an output space. Networks that use clustering techniques, such as the radial basis function (RBF) networks, are local in nature and map only part of the input space to the output (Haykin, 1999; Braga et al., 2000).

The MLP and RBF networks have their advantages and limitations. A limitation of the MLP networks is that an input from an untrained region results in an arbitrary output. They require a comprehensive set of training examples and, should the input space change, the network output may be in great error. Conversely, if an input of an untrained region is presented to a RBF network, its output can be identified as an unknown result and the user be informed about it. The RBF networks train faster than back-propagation MLP networks but, after the training, they are generally slower to use, requiring more computation to perform a function approximation. For small neural networks, such as those developed in this work, these computation time differences are not significant. In this study, both MPL and RBF neural networks are considered for estimating the power peak factor.

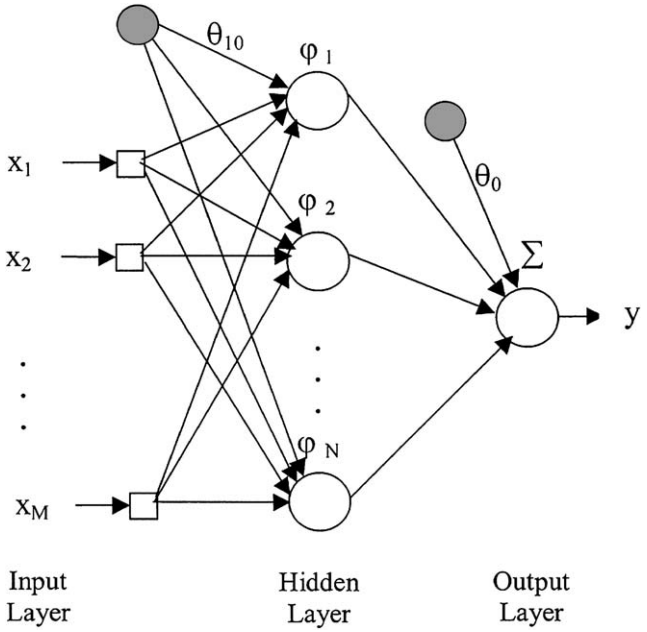

Fig. 1. Typical artificial neural network architecture with one hidden layer.

3.1. Multilayer perceptron – MLP

The MLP neural networks are the most widely used. As shown in Fig. 1, the information is transferred from layer to layer. In most applications of MLP, the weights are determined through the back-propagation algorithm, which minimises a quadratic cost function by a gradient descent method (Haykin, 1999). During the training phase, the inputs are presented to the network and propagated forward to determine the resulting signal at the output neuron. The difference between the computed output and the desired output represents an error that is back-propagated through the network in order to adjust the weights. This process is repeated and the learning continues until the desired degree of accuracy is achieved (Haykin, 1999).

Considering the one hidden layer network presented in Fig. 1, we can present in a simplified manner the mathematical formulation of MLP neural networks. A detailed description for the MLP neural networks can be found

[1] The CITATION code solves the three-dimensional, multi-energy-group neutron diffusion equation through the finite-difference method. The HAMMER/TECHNION code solves the one-dimensional, multi-energy-group neutron integral transport equation through the collision probability method and generates cross-sections in few energy groups which can be used in the CITATION code.

elsewhere (Haykin, 1999; Tsoukalas and Uhrig, 1997). According to Fig. 1, the input of the n_{th} neuron in the hidden layer is

$$v_n = \sum_{m=1}^{M} w_{nm} x_m + \theta_{n0} \tag{1}$$

where M is the total number of inputs applied to the n_{th} neuron, w_{nm} denotes the weight characterising the connection between the m_{th} input to the n_{th} hidden neuron, and θ_{n0} is a threshold offset. The output of the n_{th} neuron is given by

$$y_n = \varphi(v_n) \tag{2}$$

The activation function φ adopted for the present study was the hyperbolic tangent function $\varphi(v_n) = \tan h(v_n)$. The neural network output is then given by

$$y = \sum_{n=1}^{N} w_n y_n + \theta_0 \tag{3}$$

where w_n is the weight from the connection between the n_{th} hidden neuron and the output "y", and θ_0 is another threshold offset. The training procedure will yield at the end the values of w_n, w_{nm}, θ_{n0} and θ_0, as well as the number of hidden layers and the number of neurons in each layer.

3.2. Radial basis function network – RBF

The RBF network topology in its most basic form involves three layers, the input, the hidden and the output layer. Fig. 1 is also a schematic of the typical architecture of a RBF network with the output layer having just one output. In this case there is only the threshold offset for the output. The hidden layer consists of non-linear processing units which are connected directly to all input nodes, and each hidden neuron computes a radial basis function. The one adopted in this work is

$$\varphi_n(\mathbf{x}) = \exp\left(-\frac{1}{\sigma^2}\|\mathbf{x} - \boldsymbol{\mu}_n\|^2\right) \tag{4}$$

where $\|\ \|$ is the Euclidean norm of a vector, $\boldsymbol{x}$ is the input vector to the neural network, $\boldsymbol{\mu}_n$ is the centre of the radial basis function for the n_{th} hidden neuron, and σ is the radius or spread of all radial basis functions. When $\mathbf{x} = \boldsymbol{\mu}_n$, $\varphi_n(\mathbf{x})$ is 1 and determines the value of $\mathbf{x}$ which the neuron produces its maximum output. $\varphi_n(\mathbf{x})$ drops quickly as $\mathbf{x}$ deviates from $\boldsymbol{\mu}_n$, becoming negligible when $\boldsymbol{x}$ is far from $\boldsymbol{\mu}_n$. The neuron output has a significant response only over a defined range of values of $\mathbf{x}$, called the neuron's receptive field, which is determined by σ.

The input to each neuron in the hidden layer is just the input vector of the neural network. The output of the n_{th} neuron in the hidden layer is given by

$$y_n = \varphi_n(\mathbf{x}) \tag{5}$$

The output layer consists of a single linear unit fully connected to the hidden layer. The output of the RBF neural network is given by

$$y = \sum_{n=1}^{N} w_n y_n + \theta_0 \tag{6}$$

where w_n is the weight defining the strength of the connection between the n_{th} hidden neuron and the output.

During the training process of RBF neural networks, σ, $\boldsymbol{\mu}_n$ and w_n are adjusted to minimise the average error between the network output and the desired output.

4. Generation of data for training the neural networks

In this research, the data presented in Table 1 were used to develop neural networks for estimating the power peak factor. The data, comprising 56 different examples, were divided into two subsets: the first subset was to design and train the artificial neural networks and the second one, to validate the trained networks.

The selection of a suitable set of input examples is important for the performance of the artificial neural network. In order to be general, 80% of the data of each class (Souza and Moreira, 2006) were drawn randomly to constitute the training set for the neural network. The remaining 20% of the data was separated to constitute the validation set, and used to verify the neural networks generalisation capability. Since these examples were not presented to the network during the learning process, they became an adequate set for validation.

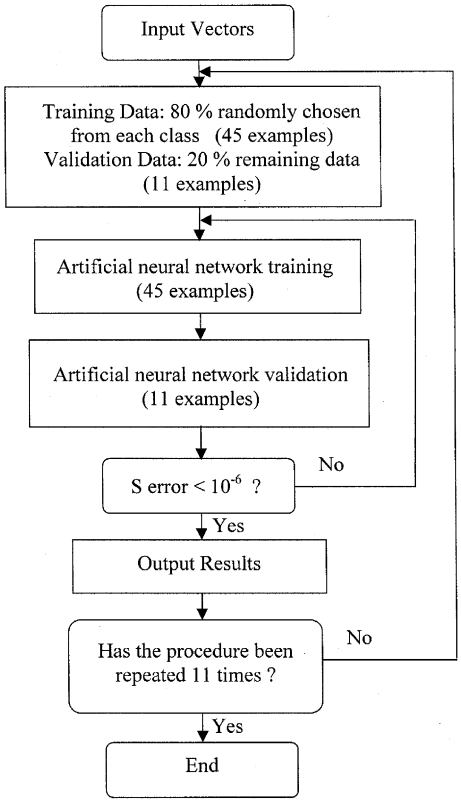

Fig. 2. Flow chart showing how the artificial neural networks were trained.

As the number of examples was limited to 56, this procedure was repeated 11 times so that 11 different training sets could be obtained. The increased number of training sets allowed a more general validation and avoided choosing particular sets that by chance could give good or bad results. Fig. 2 presents a flow chart showing how the neural networks were trained, and Table 2 presents the 11 random training sets and their corresponding validation sets.

The neural network validation was carried analysing two different types of errors, the root mean square error, S,

$$S = \left[\frac{1}{I}\sum_{i=1}^{I}\left(\mathrm{PF}_i - \mathrm{PF}_i^{NN}\right)^2\right]^{1/2} \tag{7}$$

and the average relative error, E,

$$E = \frac{1}{I}\sum_{i=1}^{I}\left|\frac{\mathrm{PF}_i^{NN} - \mathrm{PF}_i}{\mathrm{PF}_i}\right| \tag{8}$$

where I is the number of data in the subset, PF_i^{NN} is the power peak factor estimated by the neural network and PF_i is the target value presented in Table 1. The S error gives the actual magnitude of the peak factor discrepancy while the E error is a measure of the discrepancy with respect to the peak factor value. For two results with similar S errors, the one with smaller E error would be preferred.

4.1. Neural networks training

The MLP networks were designed using the resilient back-propagation algorithm available in the Matlab Neural Network Toolbox (Dermuth and Beale, 2001), because it converged faster than the standard back-propagation algorithm. During the training procedure the weights and biases were interactively adjusted to minimise the mean square error between the networks outputs and the target outputs. The training was done using the procedure referred to as *early stopping method* (Haykin, 1999; Braga et al., 2000). The training session was stopped periodically and the network was tested with the validation subset after each period of training. The decision criterion to choose the better network model was the one which minimised the mean square error of validation. The training stopped when the number of iterations exceeded the number of epochs (in our case, 10^5), when the average mean square error dropped below 10^{-6}, or when the magnitude of the gradient was less than 10^{-6}. This training procedure was repeated for networks with different numbers of neurons in the hidden layer, and the chosen network topology was the one which presented the best validation result.

The RBF network was designed with the function *newrb* from the Matlab Neural Network Toolbox. With a fixed

Table 2
States belonging to the training and validation subsets

Set	Training subset	Validation subset
1	1, 2, 4, 5, 6, 8, 9, 11, 12, 13, 14, 15, 16, 17, 18, 20, 21, 23, 25, 26, 27, 28, 30, 31, 32, 34, 35, 36, 37,38, 40, 41, 42, 43, 45, 46, 47, 48, 49, 50, 51, 53, 54, 55, 56	3, 7, 10, 19, 22, 24, 29, 33, 39, 44, 52
2	1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 13, 14, 15, 17, 18, 19, 20, 22, 23, 24, 26, 27, 29, 30, 32, 33, 34, 35, 36, 37, 38, 39, 42, 43, 44, 45, 46, 47, 48, 49, 51, 52, 53, 55, 56	5, 12, 16, 21, 25, 28, 31, 40, 41, 50, 54
3	1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 13, 14, 15, 17, 18, 19, 20, 22, 23, 24, 26, 27, 29, 30, 31, 32, 33, 34, 36, 37, 38, 39, 40, 42, 43, 44, 46, 47, 48, 49, 51, 52, 53, 55, 56	5, 12, 16, 21, 25, 28, 35, 41, 45, 50, 54
4	1, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 15, 16, 17, 19, 21, 22, 23, 24, 25, 26, 27, 28, 29, 31, 32, 33, 34, 35, 37, 39, 40, 41, 42, 43, 45, 46, 48, 49, 50, 51, 52, 53, 54, 56	2, 9, 14, 18, 20, 30, 36, 38, 44, 47, 55
5	1, 2, 3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 16, 17, 18, 19, 20, 21, 22, 24, 25, 27, 28, 29, 30, 31, 32, 33, 35, 37, 39, 40, 41, 42, 44, 45, 46, 47, 48, 49, 50, 52, 54, 55, 56	4, 8, 15, 23, 26, 34, 36, 38, 43, 51, 53
6	1, 2, 3, 4, 5, 7, 8, 9, 10, 12, 14, 15, 16, 18, 19, 20, 21, 22, 23, 25, 26, 28, 29, 30, 31, 33, 34, 35, 36, 38, 39, 40, 41, 43, 44, 45, 47, 48, 49, 50, 51, 52, 54, 55, 56	6, 11, 13, 17, 24, 27, 32, 37, 42, 46, 53
7	1, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15, 16, 18, 19, 20, 21, 22, 23, 25, 27, 28, 30, 31, 32, 33, 34, 35, 36, 38, 39, 41, 42, 44, 45, 47, 48, 49, 50, 51, 52, 53, 55, 56	2, 11, 17, 24, 26, 29, 37, 40, 43, 46, 54
8	1, 3, 4, 6, 7, 8, 9,10, 11, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 34, 35, 36, 38, 41, 43, 44, 45, 46, 48, 49, 50, 51, 52, 53, 54, 55, 56	2, 5, 14, 20, 32, 33, 37, 39, 40, 42, 47
9	1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 14, 16, 17, 20, 21, 23, 24, 25, 26, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 51, 52, 53, 56	8, 10, 15, 18, 19, 22, 27, 49, 50, 54, 55
10	1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 14, 15, 16, 18, 20, 21, 23, 24, 25, 27, 28, 29, 30, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 44, 45, 46, 47, 49, 50, 51, 53, 54, 55, 56	6, 13, 17, 19, 22, 26, 31, 42, 43, 48, 52
11	1, 2, 3, 4, 5, 6, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 25, 26, 27, 28, 30, 31, 32, 34, 35, 36, 38, 39, 41, 42, 43, 44, 45, 47, 48, 49, 50, 51, 52, 53, 54, 56	7, 8, 10, 23, 24, 29, 33, 37, 40, 46, 55

spread value, this function interactively created one neuron at a time, up to a maximum number equal to the number of examples available in each training set, which in our case it was 45. Neurons were added to the network until the sum-squared error fell under the error goal or until the maximum number of neurons was reached (45). The network which produced the smaller validation error was the one considered for this spread value. Then, the spread value was varied and the procedure was repeated. The maximum spread value considered was such that it covered the entire range of variation of the input variables. The network with smaller validation error was the one considered for that training set.

5. The neural network predictions and discussion of the results

The data presented in Table 1 show that the power peak factor can be correlated to the several variables: position of control rods, BS1, BS2, BC1, and BC2, and to the axial and quadrant power differences obtained from ex-core detectors, APD^N, APD^W, QPD_B and QPD_T. All neural networks we considered had only one output, the power peak factor. Different input vectors were studied in order to find the influence of these variables in the final result. Souza and Moreira (2006) indicated that the APDs and QPDs are complementary variables and should appear together in the input vectors in order to distinguish different peak factor values from distributions which present similar APDs or QPDs. The first input vector consisted of the four signals of position of control rods, that is, PF was considered to be given by

$$\mathrm{PF} = f(\mathrm{BS1}, \mathrm{BS2}, \mathrm{BC1}, \mathrm{BC2}) \tag{9}$$

where f is the neural network correlation, and BS1, BS2, BC1 and BC2 represent the position of the respective control rod. In the second input vector the position of the BS1 control rod was taken out and the PF was given by

$$\mathrm{PF} = f(\mathrm{BS2}, \mathrm{BC1}, \mathrm{BC2}) \tag{10}$$

The third input vector consisted of the two axial and two quadrant power differences,

$$\mathrm{PF} = f\left(\mathrm{APD^N}, \mathrm{APD^W}, \mathrm{QPD_B}, \mathrm{QPD_T}\right) \tag{11}$$

The fourth input vector consisted of all 8 variables available, and PF was given by

$$\mathrm{PF} = f\left(\mathrm{BS1}, \mathrm{BS2}, \mathrm{BC1}, \mathrm{BC2}, \mathrm{APD^N}, \mathrm{APD^W}, \mathrm{QPD_B}, \mathrm{QPD_T}\right) \tag{12}$$

The fifth input vector was similar to the fourth but the QPDs were taken out, and the input vector consisted of 6 variables

$$\mathrm{PF} = f\left(\mathrm{BS1}, \mathrm{BS2}, \mathrm{BC1}, \mathrm{BC2}, \mathrm{APD^N}, \mathrm{APD^W}\right) \tag{13}$$

and, finally, the sixth input vector which was similar to the fourth one but the BS1 control rod was taken out, and consisted of 7 variables

$$\mathrm{PF} = f\left(\mathrm{BS2}, \mathrm{BC1}, \mathrm{BC2}, \mathrm{APD^N}, \mathrm{APD^W}, \mathrm{QPD_B}, \mathrm{QPD_T}\right) \tag{14}$$

Initially, a sensitivity study was carried out to determine the best number of hidden layers to be considered for the networks. The results indicated that one hidden layer was sufficient to map the inputs to the output, e.g., a topology similar to that shown in Fig. 1. The following sections present the results obtained for the different forms of inputs vectors.

5.1. Power peak factor from signals of position of control rods

The input presented to the MLP and the RBF networks consisted of four signals of position control rods, Eq. (9), and the output was the corresponding PF value. The control rod position input data did not require any normalisation since their values were uniformly distributed over the range between 50% and 100% of position of control rods. Table 3 presents the results for the 11 subsets of training

Table 3
Topologies and root mean square errors obtained in the training of RBF and MLP neural networks using 4 positions of control rods as the input (Eq. (9))

Training subset	RBF			MLP	
	Topology		Training	Topology	Training
	Number of neurons	σ	Root mean square error	Number of neurons	Root mean square error
1	27	21.9	0.0040	5	0.0023
2	34	12.9	0.0030	5	0.0051
3	30	21.8	0.0030	5	0.0035
4	25	19.6	0.0041	5	0.0027
5	22	17.9	0.0042	5	0.0035
6	25	26.1	0.0036	5	0.0046
7	16	26.5	0.0059	5	0.0040
8	23	24.4	0.0044	5	0.0028
9	23	33.1	0.0044	5	0.0025
10	23	31.5	0.0048	5	0.0038
11	18	41.9	0.0054	5	0.0030
Average root mean square error $\overline{S}$			0.0043		0.0035

data. The topology and the final root mean square error, S, for each training subset are presented for the RBF and MLP neural networks.

For the RBF networks, the search for the optimum number of neurons in the hidden layer resulted in numbers varying from 16 to 34, showing that very different topologies are obtained when different training subsets are used. The spread, σ, varied from 12.9 to 41.9 which is comparable to the Euclidean distance between the several input vectors from the training subsets (<50). This indicates that the output of a neuron in the hidden layer has contribution of all inputs but, since they are weighted according to Eq. (4), the part of the input near the centre of the neuron contributes more to the neuron output. The training process converged well producing small final training errors. The average root mean square error considering the 11 training subsets was 0.0043.

For the MLP neural networks, the optimum number of neurons searched in the hidden layer was 5 for all training subsets. This means that the network topology did not depend on the training subset. Considering the 11 training subsets, the training process converged well producing small final training errors with an average root mean square error of 0.0035.

Figs. 3 and 4 show the training relative errors, E, obtained from all eleven training subsets, using the RBF and MLP networks. The average and maximum relative errors were 0.14% and 0.67% for the RBF networks, and 0.12% and 0.70% for the MLP networks, respectively. In both cases, over 95% of the data were inside the relative error band of ±0.50%.

The performance of the trained networks was evaluated with the validation data, e.g., those data which were not presented to the networks during the training procedures. The positions of control rods were presented to the trained neural networks as input, and the networks outputs, called PF^{NN}, were compared with the respective PFs from the validation subsets. Table 4 shows the results for the root mean square error, S, and the relative error, E, obtained for both types of neural networks. For the RBF and MLP networks, the average root mean square errors were 0.0053 and 0.0068, respectively, and the average relative errors were 0.19% and 0.26%, respectively. The comparison between the validation PF values and the PF^{NN} presented a very good agreement for all 11 validation subsets.

Figs. 5 and 6 show the comparison between the power peak factor results estimated by the neural networks and those from the validation subsets. The dashed lines in the figures represent the ±0.5% error band. For the RBF neural networks, the results show that 95% of the power peak factors were predicted within the ±0.5% relative error band, and for the MLP networks, 92% of them were

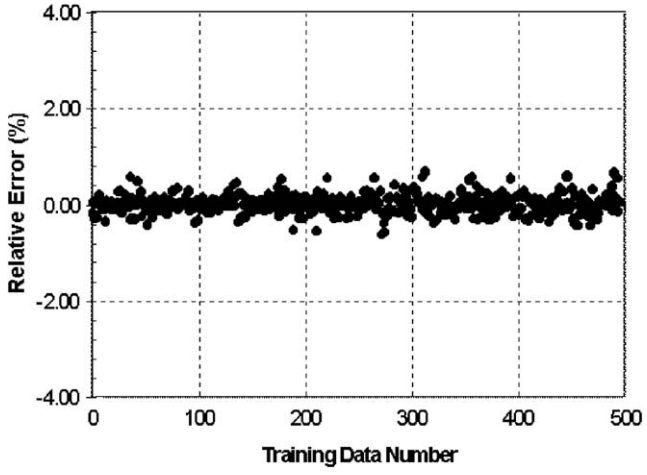

Fig. 3. Training relative error, E, between PF and PF^{NN} obtained with the RBF neural networks using 4 positions of control rods as the input (Eq. (9)).

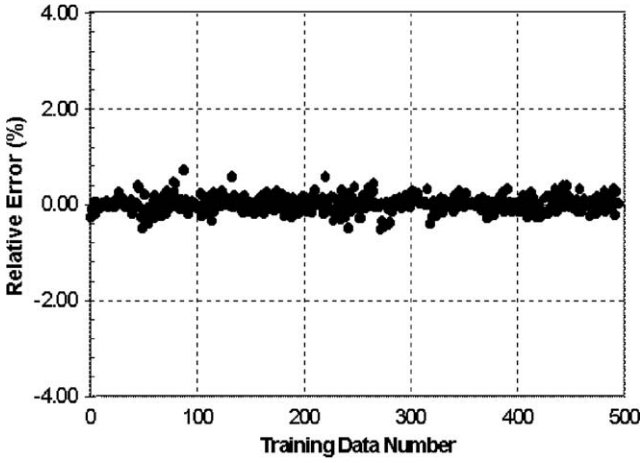

Fig. 4. Training relative error, E, between PF and PF^{NN} obtained with the MLP neural networks using 4 positions of control rods as the input (Eq. (9)).

Table 4
Prediction errors from the RBF and MLP networks using 4 positions of control rods as input (Eq. (9))

Validation subset	Validation error			
	RBF		MLP	
	Root mean square error	Average relative error (%)	Root mean square error	Average relative error (%)
1	0.0044	0.16	0.0054	0.21
2	0.0053	0.19	0.0068	0.26
3	0.0052	0.19	0.0055	0.21
4	0.0039	0.14	0.0084	0.32
5	0.0051	0.15	0.0071	0.29
6	0.0062	0.19	0.0075	0.24
7	0.0040	0.13	0.0050	0.19
8	0.0041	0.16	0.0051	0.19
9	0.0079	0.29	0.0084	0.31
10	0.0068	0.27	0.0079	0.31
11	0.0058	0.23	0.0074	0.30
Averages considering all validation subsets	0.0053	0.19	0.0068	0.26

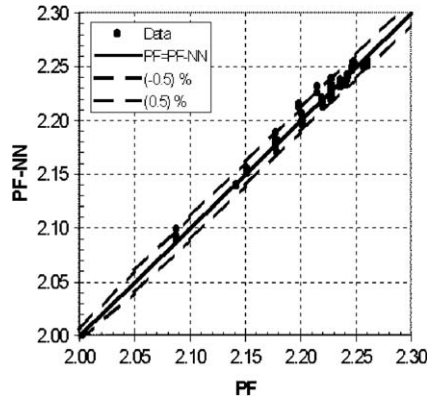

Fig. 5. Prediction of PF^{NN} by the RBF neural networks, considering all validation sets, with 4 positions of control rods as input (Eq. (9)).

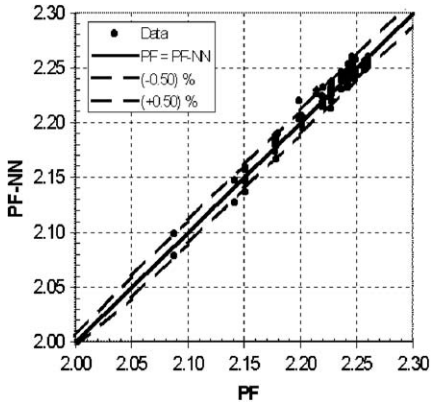

Fig. 6. Prediction of PF^{NN} by the MLP neural networks, considering all validation sets, with 4 positions of control rods as input (Eq. (9)).

predicted within the ±0.5% error band. The maximum relative errors were 0.76% and 0.90% for the RBF and MLP networks, respectively.

Table 5 presents the results obtained for the case which the input vector is given by Eq. (10). The input vector was (BS2, BC1, BC2). It was used only the RBF neural network because it can be trained more easily, and the 11 subsets of training data, defined in Table 2, were used to train the network. The number of neurons in the hidden layer varied from 16 to 35, and the spread, from 1.9 to 11.1, according to the set of training data which was used. The average root mean square error and relative error obtained during the training procedure were 0.0052 and 0.18%, respectively, and during the validation procedure were 0.0083% and 0.30%, respectively. The errors in the PF estimation were small but larger than the case having 4 positions of control rods as input.

5.2. Power peak factor from axial and quadrant power differences

In this section the input parameters to the artificial neural networks were given by Eq. (11), e.g., the axial power difference from the north and west sides, and the quadrant power difference for the bottom and top parts of the reactor core. The output was the corresponding PF value. For the RBF neural networks, the input and output data needed to be normalised in order to have a successful training process. As can be seen in Table 1, the QPDs values, which are smaller than those from the APDs, could be interpreted during the training process as less important. To circumvent this problem, a normalisation was imposed to the data so that the input vector (APDs and QPDs) presented the same standard deviation and mean equals zero. The MLP network did not require any input data normalisation for the training process.

Table 6 shows the topologies and the root mean square errors obtained during the training process. For the RBF neural networks, the average root mean square error for the training was 0.011. The minimum number of neurons in the hidden layer was 10 and the maximum, 37, and the spread varied from 0.42 to 2.10. All MLP networks had 9 neurons in the hidden layer, and the average root mean square error was 0.013. As before, the number of neurons in the hidden layer for the RBF networks varied according to the training subset which was used.

Table 5
Training and prediction errors from the RBF networks using 3 positions of control rods as input (Eq. (10))

Set	Topology		Training		Validation	
	Number of neurons	σ	Root mean square error	Average relative error (%)	Root mean square error	Average relative error (%)
1	18	9.8	0.0055	0.18	0.0090	0.31
2	20	8.8	0.0057	0.19	0.0069	0.22
3	18	8.2	0.0059	0.22	0.0076	0.25
4	23	5.1	0.0055	0.18	0.0043	0.17
5	16	7.8	0.0077	0.28	0.0097	0.35
6	35	6.3	0.0026	0.07	0.0091	0.31
7	30	4.5	0.0037	0.11	0.0080	0.32
8	21	11.1	0.0059	0.19	0.0061	0.23
9	22	4.8	0.0053	0.18	0.0081	0.30
10	21	1.9	0.0062	0.23	0.0110	0.41
11	28	2.5	0.0035	0.12	0.0115	0.38
Average for the 11 sets			0.0052	0.18	0.0083	0.30

Table 6
Topologies and root mean square errors obtained in the training of RBF and MLP neural networks using APDs and QPDs as the input (Eq. (11))

Training subset	RBF			MLP	
	Topology		Training	Topology	Training
	Number of neurons	σ	Root mean square error	Number of neurons	Root mean square error
1	22	0.42	0.010	9	0.014
2	26	0.97	0.010	9	0.021
3	28	1.09	0.010	9	0.013
4	20	1.49	0.013	9	0.011
5	10	2.10	0.022	9	0.013
6	22	1.99	0.011	9	0.011
7	37	0.92	0.008	9	0.014
8	28	1.35	0.009	9	0.010
9	27	1.90	0.012	9	0.013
10	24	1.28	0.012	9	0.017
11	36	1.07	0.004	9	0.010
Average root mean square error, $\overline{S}$			0.011		0.013

Figs. 7 and 8 show the relative errors obtained during the training process. For the RBF networks, 99% of the estimated PF were inside the ±1.70% range. The maximum and average relative errors were 3.05% and 0.35%, respectively. For the MLP neural networks, the maximum and average relative errors were 3.84% and 0.44%, respectively, and 98% of the estimated PF were inside the same error band.

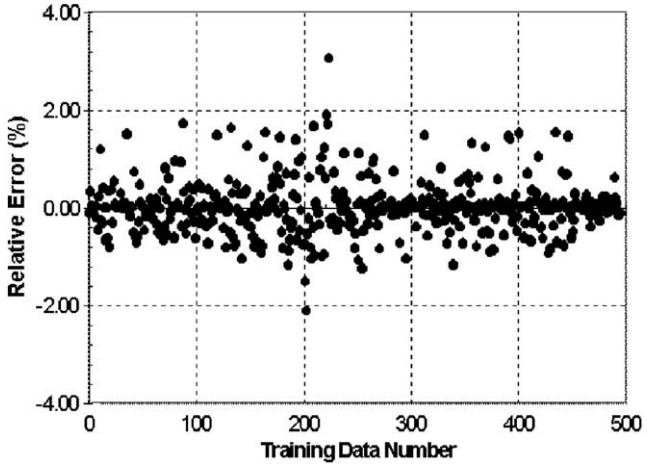

Fig. 7. Training relative error between PF and PF^{NN} obtained with the RBF neural networks using APDs and QPDs as the input (Eq. (11)).

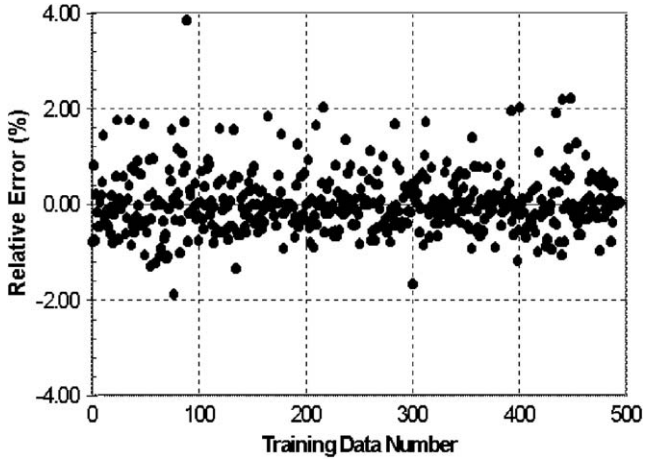

Fig. 8. Training relative error between PF and PF^{NN} obtained with the MLP neural network using APDs and QPDs as the input (Eq. (11)).

Table 7 presents the results obtained during the validation process for both types of networks. The errors were, in general, small. The average root mean square errors were 0.018 and 0.017 for the RBF and the MLP networks, respectively. The average relative errors were 0.67% and 0.60% for the RBF and the MLP networks, respectively. The average errors were taken over the results from the 11 validation subsets.

Figs. 9 and 10 show a comparison between the neural network predictions, PF^{NN}, and the validation PF values. Fig. 9 shows for the RBF networks that 96% of the PF^{NN}

Table 7
Prediction errors from the RBF and MLP neural networks using APDs and QPDs as input (Eq. (11))

Validation subset	Validation error			
	RBF		MLP	
	Root mean square error	Average relative error (%)	Root mean square error	Average relative error (%)
1	0.013	0.50	0.013	0.51
2	0.020	0.79	0.017	0.58
3	0.015	0.49	0.008	0.26
4	0.016	0.61	0.016	0.55
5	0.022	0.91	0.019	0.70
6	0.020	0.67	0.016	0.50
7	0.019	0.72	0.021	0.75
8	0.017	0.60	0.016	0.56
9	0.022	0.81	0.018	0.66
10	0.015	0.55	0.017	0.70
11	0.024	0.77	0.022	0.81
Averages considering all validation subsets	0.018	0.67	0.017	0.60

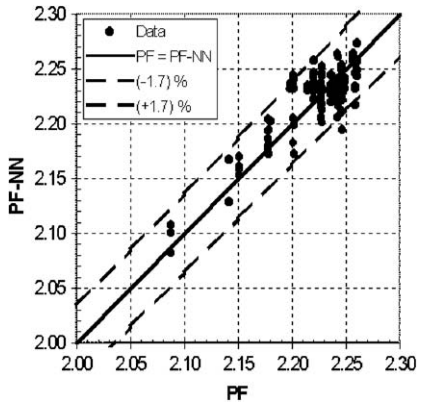

Fig. 9. Prediction of PF^{NN} by the RBF neural networks, considering all validation sets, with APDs and QPDs as input (Eq. (11)).

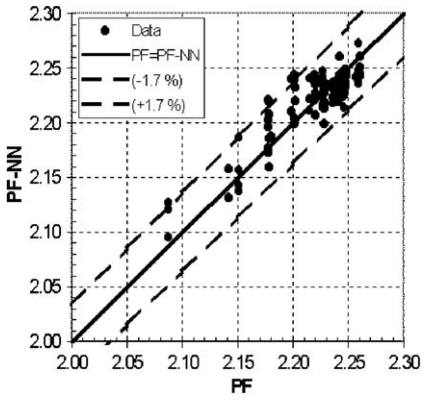

Fig. 10. Prediction of PF^{NN} by the MLP neural networks, considering all validation sets, with APDs and QPDs as input (Eq. (11)).

values were predicted within the ±1.70% error band, with maximum relative error of 2.35%. For the MLP networks, Fig. 10 shows that 93% of the PF^{NN} values were predicted within the same error band, and the maximum relative error was 1.97%.

5.3. Power peak factor from position of control rods and axial and quadrant power differences

In this section we present the results in which the input vector included information from position of control rods and axial and quadrant power differences, e.g., PF given by Eqs. (12)–(14). Table 8 summarises the principal parameters for the RBF neural networks obtained with these input vectors. As before, the RBF produced models with different numbers of neurons in the hidden layer, according to the input set used. The maximum and minimum number of hidden neurons varied from 10 to 41 for the three input vectors. The spread value varied in accordance with the number of neurons in the networks.

The training and validation errors were small for the three input vectors. Comparing their errors, we have an indication about the relative importance of each input variable to the network result. Let us consider the network result for the input vector represented by Eq. (12) which has 8 variables. Its power peak factor relative error was 0.21%. Excluding the QPDs from the input vector, we obtained the network defined by Eq. (13), which improved slightly the PF result decreasing the relative error to 0.19%. On the other hand, excluding one control rod variable from the input vector, we obtained the network defined by Eq. (14), which degraded the PF result increasing the relative error to 0.27%.

5.4. Discussion of the results

The performance of the networks to estimate the power peak factor is summarised in Table 9. The networks are ordered with respect to their relative error for estimating PF, which ranged from 0.19% to 0.67%. For practical purposes, the small errors would indicate that all networks are adequate to estimate the power peak factor but, still, some considerations can be made about the models studied.

For the networks with position of control rods as input (Eq. (9)), the comparison between the two types of neural networks, MLP and RBF, shows that the RBF networks presented little better results. Conversely, for the networks with the APDs and QPDs as input (Eq. (11)), the best results were from the MLP networks. The small differences in the errors indicate that both types of networks can be considered of similar accuracy.

The data, with many different control rod motion patterns, were characterised by localised perturbations in the power density distribution. Comparing Figs. 3 and 4 with

Table 8
Prediction errors from the RBF neural networks using positions of control rods, APDs and QPDs as input (Eqs. (12)–(14))

	4 Control rods, APDs and QPDs[a]	4 Control rods and APDs[b]	3 Control rods, APDs and QPDs[c]
Maximum number of nodes	41	36	32
Minimum number of nodes	11	15	10
Maximum spread	28.4	35.7	18.1
Minimum spread	1.6	13.4	2.5
Average training root mean square error	0.0037	0.0042	0.0059
Average validation root mean square error	0.0057	0.0052	0.0076
Average validation relative error (%)	0.21	0.19	0.27

[a] Input vector defined by Eq. (12).
[b] Input vector defined by Eq. (13).
[c] Input vector defined by Eq. (14).

Table 9
Performance of all neural networks for predicting the power peak factor

	Type of input	Type of network	Root mean square error	Average relative error (%)	Maximum relative error (%)
Eq. (13)	BS1, BS2, BC1, BC2, APD^N, APD^W	RBF	0.0052	0.19	0.74
Eq. (9)	BS1, BS2, BC1, BC2	RBF	0.0053	0.19	0.76
Eq. (12)	BS1, BS2, BC1, BC2, APD^N, APD^W, QPD_B, QPD_T	RBF	0.0057	0.21	1.07
Eq. (9)	BS1, BS2, BC1, BC2	MLP	0.0068	0.26	0.95
Eq. (14)	BS2, BC1, BC2, APD^N, APD^W QPD_B, QPD_T	RBF	0.0076	0.27	1.28
Eq. (10)	BS2, BC1, BC2	RBF	0.0083	0.30	1.41
Eq. (11)	APD^N, APD^W QPD_B, QPD_T	MLP	0.017	0.60	1.97
Eq. (11)	APD^N, APD^W QPD_B, QPD_T	RBF	0.018	0.67	2.35

Figs. 7 and 8, we notice that the training errors of the networks with position of control rods as input are about half of those having the APDs and QPDs as input. The majority of the relative errors are less than ±1%, and ±2%, respectively. The neural networks that had in the input vector information about the position of control rods were more able to describe the localised changes in the power density distribution. We could say that the position of control rods bears the detailed and localised information about the power density distribution, and that the APDs and QPDs, obtained from signals of ex-core detectors, describe its global variations in the axial and radial directions. Since in this problem the perturbations in the reactor were caused by the motion of control rods, these networks presented the best performance. Had the reactor been perturbed by xenon poisoning, which causes global axial perturbations in the power density distribution, the networks with APDs and QPDs would probably have presented better performance.

The best network was the one with 4 positions of control rods and the APDs in the input vector, Eq. (13). The inclusion of the QPDs in the input vector, obtaining the network defined by Eq. (12), did not improve the results. A possible explanation is that the global information brought by the QPDs was in excess and complicated the solution of the problem. On the other hand, the inclusion of the APDs in the input vector that had only control rod information, Eq. (13), improved the PF estimation. We can say that the global information brought to the network by the APDs were complementary and helped to estimate PF.

This complementary behaviour was also observed when the BS1 control rod position was taken out of the input vector. When the input vectors had only information about position of control rods (networks defined by Eqs. (9) and (10)), the lack of the BS1 control rod position increased the root mean square error of the PF estimation in 56%. When the input vector had also information about the APDs and QPDs (networks defined by Eqs. (12) and (14)), the lack of the BS1 control rod position increased the root mean square error of the PF estimation in only 33%. This result shows that the APDs and QPDs provided redundant information that substituted the BS1 control rod.

All results showed that 45 examples appear sufficient to train adequately the neural networks, and that the data contained the necessary information to accomplish that objective. Even though 45 examples do not constitute a large training set, it indicates that the training data covered all possible states that were generalised after the training procedure. And this was the case here since the 45 training subsets contained examples from all classes that were identified as important to describe the power peak factor behaviour (Souza and Moreira, 2006).

It seems that the RBF neural networks performed two jobs in order to produce good results. First, they identified the input vector class, and second, they interpolated the PF value according to the input vector and the identified class. In this problem the neural networks identified patterns and interpolate variables.

The maximum relative errors presented in Table 9 are comparable to the typical uncertainties that are obtained in detailed power density distribution measurements in power reactors or in experimental facilities, 5% (Miranda and Moreira, 1997). Since the targets from the training data consisted of accurate power peak factors obtained from a three-dimensional, multi-energy group diffusion theory code, the procedure presented here estimated the core power peak factors with an accuracy better than that we would obtain performing a power density distribution map with in-core detectors.

Another point that deserves to be commented regards the capability shown by the ex-core detectors to estimate the power density peak factor in reactors with small cores. The core of the IPEN/MB-01 reactor, where the training data were obtained, has a height of only 54.6 cm (Souza and Moreira, 2006; Moreira and Souza, 2002). It is known that the neutrons that contribute to the signal of an ex-core detector come from the parts of the core which are close to it. In large cores, because of the large dimensions involved, we can consider that a top ex-core detector views only the top part of the core. In the IPEN/MB-01 reactor, a top ex-core detector certainly views neutrons coming from the bottom half of the core because of the small dimensions involved. The results demonstrate that, even with this difficulty, the ex-core detectors are also capable to produce information about the power density distribution in small reactor cores.

5.5. Application to a power reactor protection system

The reactor protection system requires real time information about the power density distribution and there is an incentive for determining it more accurately because the uncertainty in the current schemes used in PWRs can be as high as 8% (Seon et al., 2002). Considering the maximum relative error of 2.35% presented in Table 9, the neural network scheme presented in this paper would allow decreasing the power peak factor safety margin by as much as 5.65%. Such a reduction could be used for operating the reactor with a higher power level, with more flexibility or even for implementing more economical and simpler core loads.

In this section we make some general comments about how to implement this neural network scheme in power reactors. This could be furnished through a dedicated neural network which has as the input vector variables that are monitored by the reactor protection system. Based on what we have discussed, we can say that neural networks can estimate the power peak factor accurately provided some points are observed: (a) the training data comprise all possible states which will be monitored in the future; (b) the training data shall be clustered into classes of similar power peak factor behaviour so that the neural network can first identify classes, and then interpolate the power peak factor accordingly; (c) the input vector shall consist of actual measured data (signals of ex-core detectors and of position of control rods) to avoid the inclusion of unknown uncertainties or biases which normally appear when simulated data are used; and (d) the power peak factor targets shall be calculated very accurately to guarantee a precise neural network estimation since the interpolation error introduced by the neural network is small.

The training data with the above characteristics can be obtained through the acquisition of the input data (position of control rods and signals from ex-core detectors) from reactor critical states, or quasi-critical states, established during the start-up tests or during operation. Since the reactor operation is carried out with specific patterns of control rod position and movement, choosing adequately the states which will take part of the training data is important.

During load following transients, the power density distribution changes continuously due to xenon poisoning, and thermal-hydraulic feedback. The position of the control rods are not the important variables to detect these perturbations because, in many states during such transients, the power density distribution changes while the control rods are kept in fixed positions. The global perturbation in the power density distribution should be accounted for by the APDs and QPDs. All possible perturbations must be considered through examples in the training set and, surely, they will constitute different classes of particular power peak factor behaviours. In this case, examples for the training data set should be obtained from quasi-static states that the reactor undergoes during transients. The neural networks would require the position of control rods, APDs and QPDs in the input vector (Eq. (12)) to identify these new classes of behaviour, and to interpolate the power peak factor.

5.6. Validation and verification for power reactor protection system

One of the challenges for a neural network application in digital protection systems is its validation and verification (V&V) process. The neural network software is required to undergo a process of V&V which must be carried out during all phases of its development, namely, the definition of the algorithm, codification, integration and test, implementation in the protection system, and its maintenance procedures (ANSI/ANS-10.4, 1998; USNRC, 1999). One advantage of the neural network scheme is that it is simple to be implemented. In this problem, it necessitates only three or four calculation steps of matrix addition, matrix multiplication, and evaluation of exponential functions such as Eqs. (2) and (4) in order to furnish the power peak factor result. Because of this simplicity, the codification may be accomplished using basic machine language, avoiding the difficulty of performing V&V on complex computer operating systems or heavy software packages. The V&V activities related to the integration, test and implementation steps become a more treatable matter.

The maintenance of the system is simple because, once it is deployed, it does not need any change. The network parameters, which take into account new core characteristics and guarantee accurate results, constitute input files that are presented to the system, and such files are easy to maintain.

The V&V process must also demonstrate that the neural network algorithm is accurate and robust, e.g., that it will not produce erroneous results due to unexpected input signals. These are more difficult tasks in the V&V process due to the neural network non-structured modelling of systems. The work presented in this paper is an example of what are these V&V activities.

6. Conclusions

This paper proposes a method based on artificial neural networks that predict the power peak factor in a form that can be used in reactor protection systems. The inputs considered to the networks were signals of position of the control rods inside the core, and axial and quadrant power differences, calculated from the ex-core detector signals. The data were collected in experiments performed in the IPEN/MB-01 zero-power reactor.

The results have shown the importance that the training data can be clustered into different classes of power peak factor behaviour. The neural networks performed two jobs in order to produce good results. First, they identified to which class the input vector belonged, and second, they

interpolated the PF value according to the input vector and the identified class. If the data cannot be organised in classes the networks may have difficulty to produce accurate PF estimations.

It was observed that the positions of control rods bear the detailed and localised information about the power density distribution. The axial and the quadrant power differences, obtained from signals of ex-core detectors, describe its global variations in the axial and radial directions.

RBF and MLP networks, having the position of control rods as input, estimated very accurately the power peak factor, with 95% and 92% of the predictions with relative errors inside the $\pm 0.5\%$ interval. Considering the case of having APDs and QPDs as input, the results were also good but with somewhat larger errors: 96% and 93% of the predictions had relative errors inside the $\pm 1.7\%$ interval.

The results showed that the RBF and MLP networks have both good generalisation capabilities to estimate the power peak factor from position of control rods and signals of ex-core detectors. The RBF networks produced slightly better results than the MLP networks but, for practical purposes, both can be considered of similar accuracy and good candidates for implementation in reactor protection systems.

The networks presented here, using position of control rods and ex-core detector signals, estimated the core power peak factors with an accuracy similar to that we would obtain performing a power density distribution map with in-core detectors. The results indicate that they allow decreasing the power peak factor safety margin by as much as 5%, which could be used for operating the reactor with a higher power level or with more flexibility.

In Section 5.5 the application to power reactors was discussed. For this case, the neural networks would require in the input vector the following variables: position of control rods, and the APDs and QPDs. The V&V requirements were discussed in Section 5.6. With this respect, one advantage of neural networks is that it is simple to implement them in protection systems but, on the other hand, it is not a simple task to demonstrate, for all reactor states of interest, that they are accurate and robust.

As a continuation of this research we intend to work in two areas: obtaining data from a nuclear power plant to evaluate the accuracy of this method during xenon transients, and investigating a V&V scheme that would allow its deployment in a power reactor.

Acknowledgements

The authors would like to acknowledge the support of the Centro Tecnológico da Marinha em São Paulo, CTMSP, the Centro de Desenvolvimento da Tecnologia Nuclear, CDTN, and the Instituto de Pesquisas Energéticas e Nucleares, IPEN. Thanks are also due to MSc. Hermelindo P. Manoel, Dr. Leonam dos Santos Guimarães, Dr. Antônio P. Braga, Prof. Dr. Benedito D. Batista Filho, MSc. Daniel Henrique Dominguete Carvalho, and the reviewers for the useful discussions and suggestions.

References

ANSI/ANS-10.4, 1998. Guidelines for the verification and validation of scientific and engineering computer programs for the nuclear industry. Rev. 1998.

Barhen, J. et al., 1978. The HAMMER Code System. Technion, EPRI-NP-565.

Boroushaki, M., Ghofrani, M.B., Lucas, C., Yazdanpanah, M.J., Sadati, N., 2004. Axial offset control of PWR nuclear reactor core using intelligent techniques. Nuclear Engineering and Design 227, 285–300.

Braga, A.P., Ludermir, T.B., Carvalho, A.C.P.L.F., 2000. Redes Neurais Artificiais: Teoria e Aplicações. Rio de Janeiro, LTC (in Portuguese).

Dermuth, H., Beale, M., 2001. Neural Network Toolbox User's Guide – for use with MATLAB. The MathWorks Inc., Massachusetts.

Faria, E.F., Pereira, C., 2003. Nuclear fuel loading pattern optimization using a neural network. Annals of Nuclear Energy 30 (5), 603–613.

Fowler, T.B., Vondy, D.R., Cunninghan, G.W., 1971. Nuclear Reactor Core Analysis Code: CITATION. Rev. 2. Oak Ridge National Laboratory, Oak Ridge, TN (ORNL-2496).

Guanghui, S., Morita, K., Fukuda, K., Pidduck, M., Dounan, J., Miettinen, J., 2003. Analysis of the critical heat flux in round vertical tubes under low pressure and flow oscillation conditions. Applications of artificial neural network. Nuclear Engineering and Design 220 (1), 17–35.

Haykin, S., 1999. Neural Networks – a Comprehensive Foundation. Prentice Hall, Upper Saddle River.

Ikonomopoulos, A., Van der Hagen, T.H.J.J., 1997. A novel signal validation method applied to a stochastic process. Annals of Nuclear Energy 24 (13), 1057–1067.

Kim, H.-C., Chang, S.H., 1997. Development of a back propagation network for one-step transient DNBR calculations. Annals of Nuclear Energy 24 (17), 1437–1446.

Kim, H.-K., Lee, S.-H., 1993. Neural network model for estimating departure from nucleate boiling performance of a pressurized water reactor core. Nuclear Technology 101, 111–122.

Kim, H.G., Chang, S.H., Lee, B.H., 1993. Pressurized water reactor core parameter prediction using an artificial neural network. Nuclear Science and Engineering 113, 70–76.

Lee, G.-C., Chang, S.H., 2003. Radial basis function networks applied to DNBR calculation in digital core protection systems. Annals of Nuclear Energy 30, 1561–1572.

Lin, C., Shen, C.-M., 2000. Neurocontrol of pressurized water reactors in load-follow operations. Nuclear Technology 132 (3), 389–402.

Maiorino, J.R. et al., 1989. Projeto nuclear da unidade crítica IPEN/MB-01In Encontro Nacional de Física de Reatores e Termo-hidráulica, VII, Recife, Brazil, April 26–28, vol. 1. Editora Universitária UFPE, Anais....Recife, 311–323, (in Portuguese).

Miranda, A.F., Moreira, J.M.L., 1997. Mapeamento do fluxo de nêutrons no reator IPEN/MB-01 com câmara de fissão miniatura. In: Encontro Nacional de Física de Reatores e Termo-hidráulica, XI, Poços de Caldas, M.G., Brazil. Anais···Poços de Caldas, ABEN, (CD-ROM) (in Portuguese).

Moreira, J.M.L., Souza, R.M.G.P., 2002. Improving the peak power density estimation for the DNBR trip signal utilizing out-of-core detectors. In: International Symposium on Nuclear Power Plant Life Management, Budapest, Hungary, 4–8 November.

Na, M.G., Upadhyaya, B.R., 1998. A neuro-fuzzy controller for axial power distribution in nuclear reactors. IEEE Transactions on Nuclear Science 45 (1), 59–67.

Na, M.G., Shin, S.H., Lee, S.M., Jung, D.W., Lee, K., Lee, Y.J., 2004. Estimation of axial DNBR distribution at the hot pin position of a reactor core using fuzzy neural networks. Journal of Nuclear Science and Technology 41 (8), 817–826.

Nabeshima, K., Suzudo, T., Suzuki, K., Türkan, E., 1998. Real-time nuclear power plant monitoring with neural network. Journal of Nuclear Science and Technology 35 (2), 93–100.

Seon, Y., Cha, C.K.H., Park, M.G., Lee, C.S., 2002. Predictive mathematical modeling for excore neutron detectors using a neural network. In: PHYSOR 2002, Seoul, Korea, 7–10 October.

Seong, S.-H., Park, H.-Y., Kim, D.-H., Suh, Y.-S., Hur, S., Koo, I.-S., 2002. Development of fast-running simulation methodology using neural networks for load follow operation. Nuclear Science and Engineering 141 (1), 66–77.

Souza, R.M.G.P., Moreira, J.M.L., 2006. Power peak factor for protection systems – experimental data for developing a correlation, in press.

Su, G., Fukuda, K., Jia, D., Morita, K., 2002. Application of an artificial neural network in reactor thermohydraulic. Problem: prediction of critical heat flux. Journal of Nuclear Science and Technology 39 (5), 564–571.

Tsoukalas, L.H., Uhrig, R.E., 1997. Fuzzy and Neural Approaches in Engineering. Wiley, New York.

USNRC, 1995. US Nuclear Regulatory Commission, Standard technical specification, Westinghouse Plants, bases. Section 3.2 – power distribution limits. Rev. 1. v. 2, (NUREG-1431).

USNRC, 1999. US Nuclear Regulatory Commission, Regulatory guide 1105 – Setpoints for the safety related instrumentation. Rev. 3.