Proceedings of ASME Turbo Expo 2004
Power for Land, Sea, and Air
June 14-17, 2004, Vienna, Austria

GT2004-53421

SET UP OF A ROBUST NEURAL NETWORK FOR GAS TURBINE SIMULATION

R. Bettocchi
Dip. di Ingegneria – University of Ferrara
Via Saragat, 1 - 44100 Ferrara, Italy

M. Pinelli
Dip. di Ingegneria – University of Ferrara
Via Saragat, 1 - 44100 Ferrara, Italy

P. R. Spina
DIEM – University of Bologna
Viale Risorgimento, 2 - 40136 Bologna, Italy

M. Venturini
Dip. di Ingegneria – University of Ferrara
Via Saragat, 1 - 44100 Ferrara, Italy

M. Burgio
Dip. di Ingegneria – University of Ferrara
Via Saragat, 1 - 44100 Ferrara, Italy

ABSTRACT

In this paper, Neural Network (NN) models for the real-time simulation of gas turbines are studied and developed. The analyses carried out are aimed at the selection of the most appropriate NN structure for gas turbine simulation, in terms of both computational time of the NN training phase and accuracy and robustness with respect to measurement uncertainty. In particular, feed-forward NNs, with a single hidden layer and different numbers of neurons, trained by using a back-propagation learning algorithm are considered and tested. Finally, a general procedure for the validation of computational codes is adapted and applied to the validation of the developed NN models.

NOMENCLATURE

A_{IGV}	Inlet Guide Vane angular position
e	$= t - y$ error
LHV	fuel lower heating value
MSE	mean square error
M	mass flow rate
m	number of inputs
n_{HLN}	number of neurons in the hidden layer
n_{N}	number of neurons
n_{o}	number of outputs
n_{patt}	number of patterns
N	rotational speed
N_{ep}	number of epochs
p	pressure
P	power
Q	quantity
$rand$	random number
RH	relative humidity
$RMSE$	root mean square error
SR	success rate
t	expected target output
T	temperature
u	individual experimental uncertainty
U	combined uncertainty
$U_{y\mathrm{MA}}$	simulation modeling uncertainty arising from modeling assumptions
$U_{y\mathrm{N}}$	simulation numerical solution uncertainty
$U_{y\mathrm{PED}}$	simulation modeling uncertainty arising from the use of previous experimental data
w	weight associated to the neuron x input
x	neuron input
y	computed output
Δ	variation
η	efficiency health index, learning rate
ϑ_i	$= \partial t/\partial x_i$ sensitivity coefficient
μ	corrected mass flow health index

Subscripts and Superscripts

amb	ambient
cool	cooling flow
C	compressor
ED	exhaust duct
f	fuel
GT	gas turbine
i	inlet section
ID	inlet duct
l	lower limit
max	maximum
min	minimum
norm	normalized value
o	outlet section

PH	post heating
T	turbine
u	upper limit

INTRODUCTION

Thermodynamic-based programs have been widely used over many years in order to perform thermodynamic and thermoeconomic analyses and optimizations of energy conversion systems [1-5]. Due to the non-linearity of equations to be solved, these programs usually present computational times which are excessive for real-time calculations, especially when:

- the system to simulate is complex - i.e. it consists of a high number of sub-systems (units) - and, so, the number of equations to be solved is very high;
- optimizations are performed, which usually require iterative calculations to solve objective functions;
- the time available for the real-time calculations is very short, compared with computational time. This happens, for instance, when an almost instantaneous optimized re-allocation of the loads among the units of a power generation system is required following a variation in the electric and/or thermal energy demand [6,7].

An alternative to *physical* models is given by *black box* models, such as autoregressive models, Neural Networks, etc., which present some advantages: (i) they do not require knowledge of the physics of the problem under investigation, (ii) capability of learning different typologies of information, (iii) high robustness in the presence of poor and incorrect input data and, above all, (iv) high computational speed, which allows real-time calculations [8-11]. On the other hand, the main limit of black box models is high prediction error when they operate outside the field in which they were trained, i.e. they are not able to extrapolate. Among the various black box models, Neural Networks (NNs) have proved to be flexible and robust in simulating and diagnosing the behavior of energy systems, since they are able to simulate non-linear systems both in steady-state [8-15] and in transient [16,17] conditions.

In the paper, NN models for the real-time simulation of gas turbines are studied and developed. In particular, the data used for both training and testing the NNs were obtained by means of a thermodynamic-based program (cycle program), previously calibrated on a FIAT Avio 701F single shaft gas turbine working in the ENEL combined cycle power plant of La Spezia (Italy) [18]. The cycle program, which can reproduce machine behavior for different ambient, load and health state conditions, allows the calculation of the expected value of all the measurable variables which might be taken on the machine and of non-measurable quantities (such as the inlet turbine temperature, air and exhaust mass flow rates, turbine cooling flow rates, compressor and turbine efficiency, etc.). A NN model able to simulate a gas turbine in different boundary, load and health state conditions can be very important. As an example, to follow variations of the electric and/or thermal energy demand, a power generation system manager can perform, by means of this tool, an optimized re-allocation of the loads among the units by taking into account the actual health state of each unit (which varies due to aging, deterioration and faults) [6,7].

In order to develop the NN model, the first step was the selection of the input and output variables and of the range of variation for the inputs. Then, a database containing the patterns for NN training and testing was obtained by running the cycle program with random values of the inputs, each value being within its respective range of variation. Some of the patterns were used for the NN training phase, while the rest were used for NN testing.

The analyses carried out are orientated to the selection of the most appropriate NN structure for gas turbine simulation, in terms of both computational time of the NN training phase and accuracy and robustness towards measurement uncertainty during simulations. In particular, feed-forward NNs with a single hidden layer with different numbers of neurons and trained using a back-propagation learning algorithm were considered and tested.

Finally, a procedure for the validation of computational codes originally proposed by Coleman and Stern [19], was adapted and applied to the developed NN models.

DATA FOR NN TRAINING AND TESTING

Gas turbine under consideration. The gas turbine considered in the paper is a 255 MW FIAT Avio 701F single shaft gas turbine with variable Inlet Guide Vane, whose main features at ISO conditions are reported in Tab. 1. This gas turbine has a 17-stage axial flow compressor and a 4-stage turbine, cooled by means of air flows extracted from four different bleed ports on the compressor. In particular, the portion of the cooling flow extracted at the compressor outlet used for the cooling of the first turbine nozzle is externally cooled in a heat exchanger used for steam production.

Table 1 – Fiat Avio 701F gas turbine: main thermodynamic and performance data (ISO conditions, natural gas fuel, no intake and discharge pressure drops)

Power output	255 MW
Overall efficiency	36.0 %
Exhaust gas temperature	889 K
Exhaust mass flow rate	574 kg/s
Fuel mass flow rate	13.2 kg/s
Compressor discharge temperature	655 K
Pressure ratio	13.9

For this machine, a program for the thermodynamic cycle calculation (cycle program), previously calibrated on a FIAT Avio 701F gas turbine working in the ENEL combined cycle power plant of La Spezia (Italy) [18], was available. The calibration process permitted a satisfactory reproduction of all the thermodynamic and performance data provided by the gas turbine manufacturer.

In Fig. 1, the sketch of the gas turbine model under consideration is shown.

Data generation. The data used for both training and testing the NNs were generated by means of the above described cycle program in different working points at different boundary, load and health state conditions.

In particular, in order to take into account different gas turbine health states, characteristic parameters of the gas turbine, which are indices of the machine actual operating state (*health indices*), were considered as input variables. These

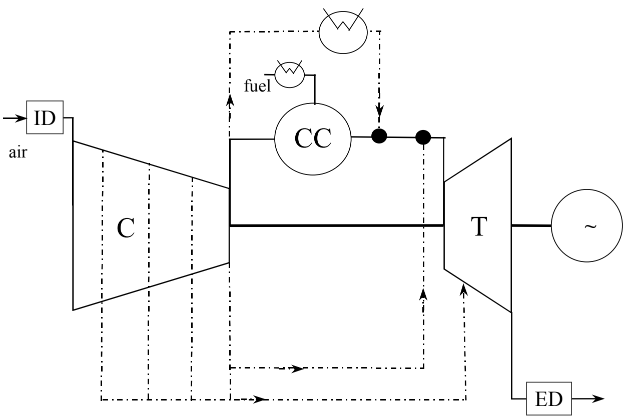


Figure 1 – FIAT Avio 701F gas turbine model

parameters do not depend on the gas turbine operating point but are only sensitive to the gas turbine health state [14,20-22]. The characteristic parameters considered are the health indices related to compressor and turbine efficiencies and corrected mass flow rates (η_C, η_T, μ_C and μ_T).

In Tab. 2 the input variables with their range of variations and the output variables are reported. Regarding the health indices, variations within typical ranges related to the most common deteriorations and faults to which the gas turbine can be subjected were considered.

Table 2 – NN inputs and outputs

		Unit	Variation Range
INPUT			
p_{amb}	Ambient pressure	kPa	[98; 103]
T_{amb}	Ambient temperature	K	[278.15; 308.15]
RH	Relative humidity	%	[20; 100]
Δp_{ID}	Pressure drop at the compressor inlet	kPa	[0; 2.5]
Δp_{ED}	Pressure drop at the turbine outlet	kPa	[0; 4.0]
N	Rotational speed	rpm	[2 990; 3 010]
T_{cool}	Cooling flow temp. after the exchanger	K	[453.15; 493.15]
T_{fPH}	Temperature of the fuel after heating	K	[288.15; 383.15]
LHV	Fuel lower heating value	kJ/kg	[40 000; 50 000]
A_{IGV}	IGV angular position	°	[-4; 25]
T_{iT}	Turbine inlet temperature	K	[1 150; 1 750]
η_C	Compressor efficiency health index	-	[0.93; 1.00]
μ_C	Comp. corrected mass flow health index	-	[0.90; 1.10]
η_T	Turbine efficiency health index	-	[0.93; 1.05]
μ_T	Turb. corrected mass flow health index	-	[0.90; 1.10]
OUTPUT			
p_{oC}	Compressor outlet pressure	kPa	
T_{oC}	Compressor outlet temperature	K	
M_f	Fuel mass flow rate	kg/s	
T_{oT}	Turbine outlet temperature	K	
M_{oT}	Exhaust gas mass flow rate	kg/s	
P_{GT}	Gas turbine power	kW	

Subsequently, a database containing the patterns for NN training and testing was obtained by running the cycle program with random values of inputs, each one lying within its respective range of variation. In particular, the input quantities were generated as follows:

$$Q_i = (Q_i)_l + rand\,\left[(Q_i)_u - (Q_i)_l\right], \quad rand \in [0, 1]$$

where $(Q_i)_u$ and $(Q_i)_l$ are the upper and lower limits of the variation range, respectively. The generated data were then normalized to be comparable with each other. The normalization was performed with respect to the maximum and minimum value of the data set generated for each variable, i.e.:

$$(Q_i)_{norm} = \frac{Q_i - (Q_i)_{min}}{(Q_i)_{max} - (Q_i)_{min}}, \quad (Q_i)_{norm} \in [0, 1]$$

Some of the patterns were used for the NN training phase, while the rest were used for NN testing.

Measurement errors. To take into account the presence of measurement error, the generated data were corrupted with a random value included in the measurement uncertainty intervals reported in Tab. 3. Three different instrumentation categories, characterized by different measurement uncertainties, were considered: laboratory, standard and industrial. This was done in order to consider different situations which can be found in practice, and to perform a sensitivity analysis with respect to the influence of the measurement error magnitude on NN performances.

Table 3 – Measurement uncertainty values

Quantity	Instrumentation categories and measurement uncertainty [a]		
	laboratory	*standard*	*Industrial*
INPUT			
p_{amb} [kPa]	0.525	0.630	1.050
T_{amb} [K]	0.2	0.3	0.5
RH [%]	1.0	2.0	3.0
Δp_{IC} [kPa]	0.08	0.10	0.12
Δp_{ED} [kPa]	0.08	0.10	0.12
N [rpm]	15.0	30.0	40.0
T_{cool} [K]	0.5	1.0	2.0
T_{fPH} [K]	0.5	1.0	2.0
LHV [%] [b]	1.0	3.0	5.0
A_{IGV} [°]	0.5	1.0	2.0
T_{iT} [K]	4.0	10.0	20.0
OUTPUT			
p_{oC} [kPa]	10	20	30
T_{oC} [K]	0.6	1.5	2.5
M_f [%] [b]	0.5	1.5	2.5
T_{oT} [K]	4.0	8.0	12.0
M_{oT} [%] [b]	0.5	1.5	2.5
P_{GT} [%] [b]	1.0	2.0	3.0

[a] All values reported have to be considered as preceded by "±".
[b] Percentage of the reading.

As regards measurement uncertainty values reported in Tab. 3, it should be noted that:

- Relative humidity uncertainty values are absolute values. Thus, since *RH* measurements are expressed in percentages, the uncertainties u_{RH} are also expressed in percentages.
- The *LHV* uncertainty (expressed as a percentage of the reading) has to be considered as an overall uncertainty on *LHV* value, rather than a "strictly-speaking" measurement uncertainty. For instance, in the case of "industrial"-type instrumentation, the 5 % error, which might seem huge, takes into account that the on-line *LHV* measurement is rarely performed in practice but, rather, the *LHV* value is obtained either from previous data or from averaged values furnished by the fuel gas supplier.
- The uncertainty in turbine inlet temperature T_{iT}, which is a non-measurable quantity, was estimated numerically with the cycle program, by imposing variations to the turbine outlet temperature T_{oT} equal to its measurement uncertainty, and by calculating the relative variation of the T_{iT}. In particular, the variations in T_{iT} which correspond to variations of ± 4.0 K, ± 8.0 K and ± 12.0 K on T_{oT} were calculated. The results obtained are in agreement with the ones presented in [23].

ARTIFICIAL NEURAL NETWORKS

Artificial NNs are mathematical structures that distribute input data into several interconnected simple units (the artificial neurons) in which data are processed in parallel. Due to their high connectivity and parallelism, artificial NNs are able to link, in a non-linear way, a multi-dimensional input space with a multi-dimensional output space, allowing very high computational speed [24].

The NN architecture used in this paper for gas turbine simulation is the typical *feed-forward multilayer perceptron*, in which the artificial neurons are arranged in layers, and the neurons of a layer are linked to all the neurons of the following layer, while, there are no links among neurons of the same layer. This type of NN consists of a set of input nodes (also called *input layer*, where no data processing occurs) equal to the number of NN inputs, one or more hidden layers and an output layer with a number of neurons equal to the number of NN outputs. All the calculations are carried out in hidden and output layers. In particular, if x_{ij} is the ith input of the jth neuron and w_{ij} is the weight of x_{ij}, the neuron output y_j is determined by means of an activation function f applied to the weighted sum of the inputs

$$y_j = f\left(\sum_{i=1}^{m_j} w_{ij} x_{ij}\right) \quad , \; j = 1,\ldots,n_{\mathrm{N}} \qquad [1]$$

In particular, NNs with one hidden layer and a continuous sigmoid activation function were used, since it has been shown that this type of NN architecture is able to represent any type of multidimensional non-linear function, if a suitable number of neurons of the hidden layer is chosen [12,15,25]. Moreover, a preliminary analysis carried out to evaluate the influence of the number of hidden layers, showed that the use of multiple hidden layers requires a great computational effort, while it only allows a small improvement of NN performance.

For the NN training, which is the process for weight calculation, the *back-propagation* (BP) algorithm was used [26]. This is a supervised learning process, in which, for each set of inputs, the computed outputs, obtained by propagating the information forward from input to output, are compared with the target ones. The errors between computed and target outputs are then transmitted backward through the network. During this phase, the weights, which were randomly initialized, are progressively updated using a generalized delta rule expressed as

$$\Delta w_{ij} = \eta \cdot e_j \cdot x_{ij} \quad , \; j = 1,\ldots,n_{\mathrm{N}} \qquad [2]$$

where η is the *learning rate* and e_j is the error relative to the output of the jth neuron. In particular, η determines the computational time required for the NN training phase: by increasing η the rate of error reduction during the learning process increases, but, in the final learning phase, high η values may cause error oscillations, so that the convergence of the learning process may not be achieved.

In the MATLAB Neural Network toolbox, a number of different BP algorithms are present. After a preliminary analysis, the *TRAINSCG* algorithm was used since it appears to be very effective and requires a reduced computational effort. This algorithm is based on a Conjugate Gradient scheme in which the learning rate is varied at each iteration. The adopted stopping criterion for the NN training phase is the minimization of a performance function which was chosen to be the Mean Square Error (*MSE*) on the whole training set between the target outputs and the corresponding NN computed outputs:

$$MSE = \frac{1}{n_{\mathrm{o}} \cdot n_{\mathrm{patt}}} \sum_{j=1}^{n_{\mathrm{o}}} \sum_{i=1}^{n_{\mathrm{patt}}} e_{ij}^2 = \frac{1}{n_{\mathrm{o}} \cdot n_{\mathrm{patt}}} \sum_{j=1}^{n_{\mathrm{o}}} \sum_{i=1}^{n_{\mathrm{patt}}} \left[t_{ij} - y_{ij}\right]^2 \qquad [3]$$

where n_{o} is the number of NN outputs, n_{patt} is the number of patterns used for the NN training, t_{ij} are the target outputs and y_{ij} the NN computed outputs. Two additional stopping criteria were used:

- attainment of an imposed maximum number of epochs (i.e. number of times that all the patterns are presented to the NN) for the NN training phase; this number was imposed as equal to 10^6;
- *early stopping method*, i.e. the training process is stopped when the *MSE* on a validation set of data increases or remains the same for an imposed number of epochs; this number was imposed as equal to 100.

INFLUENCE ANALYSIS OF NN CONFIGURATION PARAMETERS

In order to identify the most appropriate NN structure for gas turbine simulation, in terms of both computational time of the NN training phase and accuracy and robustness against measurement uncertainty, an influence analysis was carried out with respect to the main NN configuration parameters.

As explained in the previous paragraph, the NNs considered are *feed-forward multilayer perceptrons* with one hidden layer using a continuous sigmoid activation function. Moreover, the *TRAINSCG* BP algorithm of the MATLAB Neural Network toolbox was selected for the NN training. Therefore, the NN configuration parameters whose influence was analyzed in the paper are:

- number of neurons in the hidden layer (n_{HLN});
- optimal training set dimension, i.e. optimal number of patterns that have to be used for the NN training process.

The influence of these parameters was also analyzed in relation to NN accuracy and robustness against measurement uncertainty.

The parameter used for the comparison of the NNs is the Root Mean Square Error *RMSE* made by the NN on the whole set of test data in the calculation of each *j*-th output (p_{oC}, T_{oC}, M_f, T_{oT}, M_{oT}, P_{GT}):

$$\left(RMSE = \sqrt{\frac{1}{n_{\mathrm{patt}}} \sum_{i=1}^{n_{\mathrm{patt}}} \left(\frac{t_i - y_i}{t_i} \right)^2} \right)_j \quad , \ j = 1,\dots,n_o \qquad [4]$$

where t_i are the target outputs, y_i the computed outputs, n_{patt} is the number of patterns used for the NN test and n_o is the number of NN outputs.

An overall *RMSE* was also calculated as the Root Mean Square Error made by the NN on the whole set of test data on all outputs:

$$RMSE_{Overall} = \sqrt{\frac{1}{n_o} \sum_{j=1}^{n_o} (RMSE)_j^2} \qquad [5]$$

Number of neurons in the hidden layer. The influence of the number of neurons in the hidden layer on the NN simulation error was evaluated by comparing the response of different NNs with different numbers of neurons in the hidden layer. Each NN was trained by using 10 000 patterns, while the *RMSE* was evaluated by running the NN in order to reproduce 20 000 patterns (not including the patterns used for training). Figure 2 shows the *RMSE* on each output and the overall *RMSE* for different numbers of neurons in the NN hidden layer. It can be noticed that the $RMSE_{Overall}$ presents a minimum for a number of neurons in the hidden layer near to 60. In the analyses carried out in the rest of the paper, NNs with 60 neurons in the hidden layer were used. Nevertheless, even a 30-neuron hidden layer could be considered an acceptable compromise between NN accuracy and computational time required for the NN training. In fact, as shown in Fig. 3, in which the number of epochs required for the NN training phase are reported, the computational time required to train the NN dramatically increases with the number of neurons in the hidden layer.

Number of training patterns. The influence of the number of patterns used to train the NN on the simulation error was evaluated by comparing the response of different NNs, all characterized by 60 neurons in the hidden layer, which were trained by using different training sets with a number of patterns from 100 to 10 000. The *RMSE* of each NN was evaluated by running the NN in order to reproduce 20 000 patterns (not including the patterns used for training). It has been observed that, for a number of patterns lower than 500, errors are very high, due to the phenomenon called *overfitting*. In fact, in such cases (i.e. a low number of training patterns) the NN learns the training data so well, that it loses the ability to generalize and tends to reproduce the training data even when different inputs are supplied to the NN [24]. In any case, although the *early stopping method* is used, errors remain unacceptably high when less than 500 training patterns are used. In Fig. 4 *RMSE* on each output for training sets with the number of patterns for 500 to 10 000 is reported. It can be observed that, when at least 1 000 training patterns are used, *RMSE* is quite small and, thus, it is not convenient to use a higher number of training patterns due to the increase in the training phase computational time.

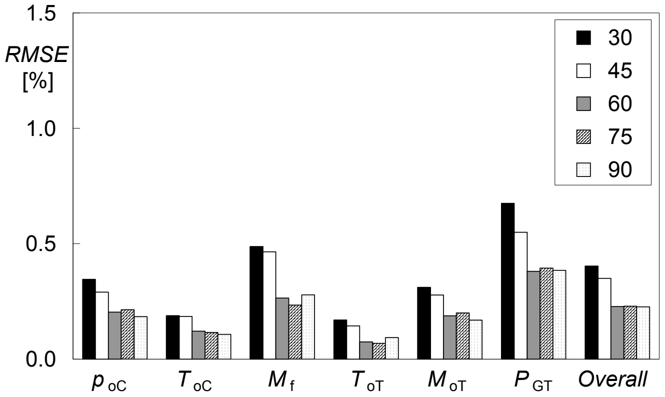


Figure 2 – Influence of number of neurons

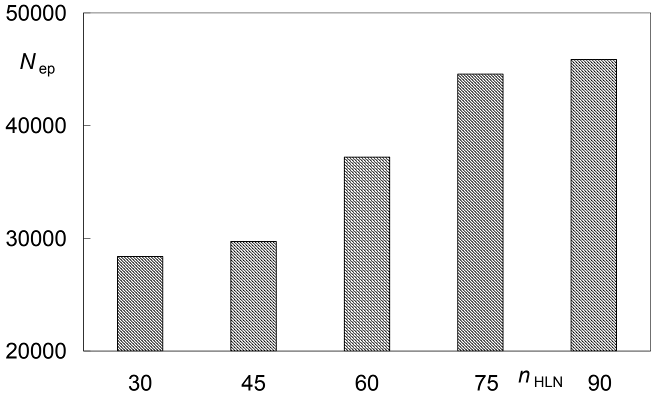


Figure 3 – Number of epochs required for NN training vs. number of neurons in the hidden layer

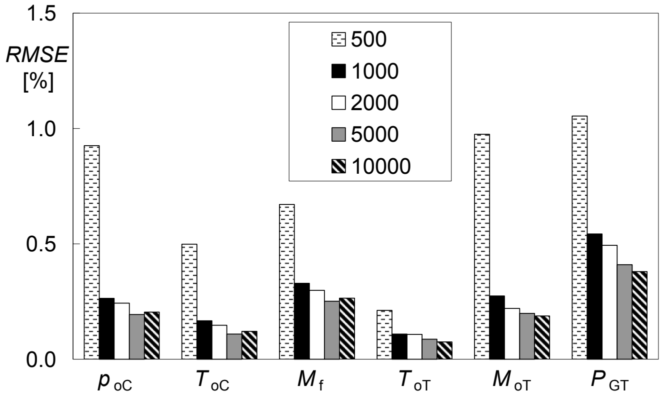


Figure 4 – Influence of Number of training patterns

Measurement uncertainty. In order to assess the influence of measurement uncertainty on NN accuracy and robustness, a NN characterized by a number of neurons in the hidden layer equal to 60 was considered. For training purposes, the NN was trained by using 2 000 training patterns not affected by measurement errors (the reason why 2 000 patterns were adopted for NN training is explained below). In the test phase, the *RMSE* was evaluated by running the NN in order to reproduce 20 000 patterns (as usual, they do not include the patterns used for training) which were instead corrupted by means of random measurement errors included in the uncertainty ranges reported in Table 3.

The results are reported in Fig. 5a, where it can be noticed that, as expected, *RMSE* increases as measurement error increases. In particular, when data are not affected by measurement error, *RMSE* is considerably smaller (less than 0.5 %) as previously observed (Fig. 4), while, considering measurement errors, *RMSE* rises to about 5 % in the worst cases (i.e. for fuel mass flow rate M_f and power output P_{GT} which prove to be the most sensitive quantities).

In order to reduce *RMSE* values, a higher number of neurons in the hidden layer could be adopted, as evidenced in Fig. 2. However, an analysis of the influence of the number of neurons in the hidden layer in the case of data corrupted by measurement errors showed that the benefit of increasing the number of neurons in the hidden layer reduces for high values of measurement error, so that it becomes negligible in case of "industrial"-type measurement uncertainty [27].

A further analysis was performed by training the NN (60 neurons in the hidden layer) by means of 2 000 patterns corrupted by errors within the "standard"-type measurement uncertainty. The results are reported in Fig. 5b and comparison with the results presented in Fig. 5a shows that, for all NN outputs, *RMSE* is lower if a NN trained on corrupted data is used instead of a NN trained with uncorrupted data, only for test data affected by errors equal or higher than those used for training. This result is in accordance with the results presented in [13] and is thoroughly outlined in the next section.

Finally, an analysis was performed in order to establish the influence of the number of training patterns when data affected by "standard"-type measurement errors are used for both training and testing. The results, reported in Fig. 6, if compared to those presented in Fig. 4, show that:

- *RMSE* is higher when the presence of measurement uncertainty is taken into account, as expected;
- an "asymptotic" *RMSE* value for all quantities is reached when at least 2 000 training patterns are used.

VALIDATION OF THE NN MODEL

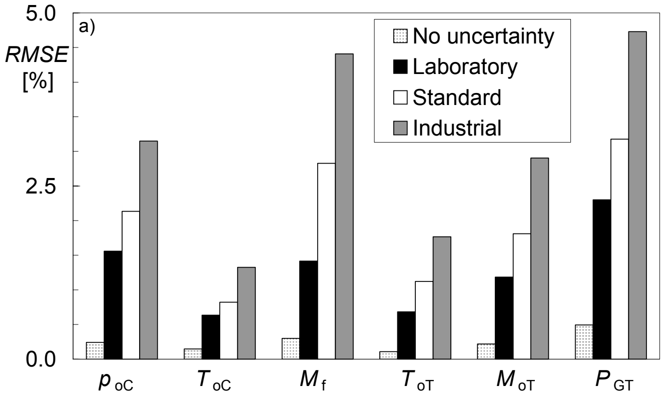
The validation of a prediction model is performed by comparing the values computed by the model with reference values, which can be experimental data, or data obtained by a reference simulation model. A procedure for the validation of prediction codes using experimental data was proposed by Coleman and Stern [19]. This procedure is here adapted and applied for the validation of the developed NN models.

The comparison error *e*, difference between the expected reference value *t* and the computed value *y*, is the result of all the errors associated with *t* and *y* determination. If it is considered that (i) the expected reference value *t* is, in the most general case, a function of *m* independent variables $\boldsymbol{x} = [x_1, ..., x_m]$ and (ii) *t*, $\boldsymbol{x}$ and *y* share no common error sources, the combined uncertainty U_e can be expressed as:

$$U_e = \sqrt{U_t^2 + U_y^2} \qquad [6]$$

In the case in which the expected reference values *t* are experimental data, Coleman and Stern [19] propose that:

$$U_t^2 = u_t^2 + \sum_1^m \vartheta_i^2 \left(u_x^2\right)_i \qquad [7]$$



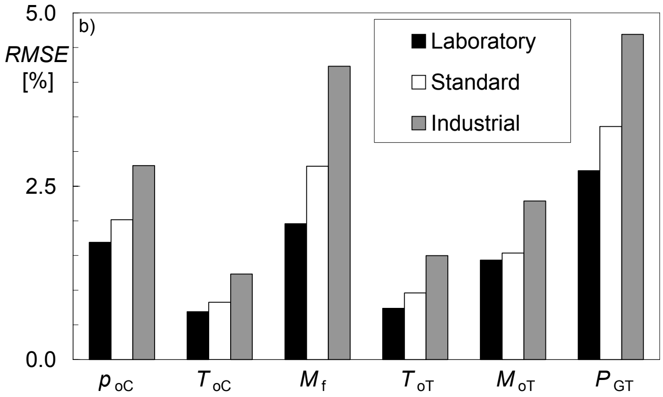


Figure 5 – Influence of measurement uncertainty on RMSE: NN trained using data not affected by measurement uncertainty (a) and corrupted by a "standard"-type measurement uncertainty (b)

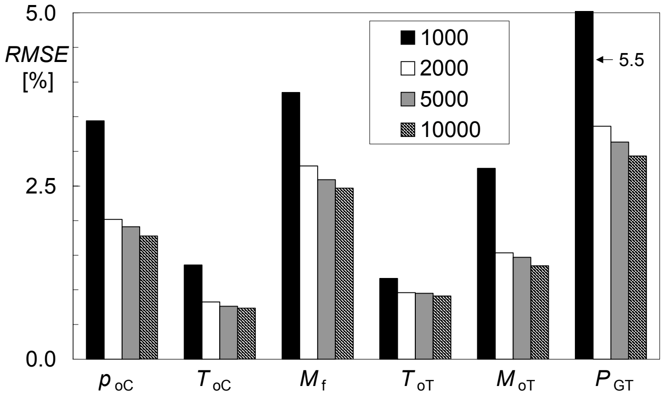


Figure 6 – Influence of number of training patterns when data affected by "standard"-type measurement errors are used for both training and testing

where U_t is the combined uncertainty in the expected reference value *t*, u_t is the experimental uncertainty in *t*, $(u_x)_i$ are the experimental uncertainties associated with x_i, and ϑ_i are the sensitivity coefficients defined as $\vartheta_i = \partial t / \partial x_i$.

For the combined uncertainty U_y in the computed value *y*, Coleman and Stern [19] propose that:

$$U_y^2 = U_{yN}^2 + U_{yPED}^2 + U_{yMA}^2 \qquad [8]$$

where U_{yN} is the simulation numerical solution uncertainty, U_{yPED} the simulation modeling uncertainty arising from the use of previous experimental data and U_{yMA} the simulation modeling uncertainty arising from modeling assumptions.

In the case presented in this paper, the expected reference values t_j (j = 1,…, n_o) are not experimental data but are data obtained by means of the cycle program, corrupted with random errors contained within the ranges of uncertainty typical of each measured variable (Tab. 3). The predicted values y_j are those computed by the NN models. Therefore, in the calculation of the combined uncertainties $(U_t)_j$, the "experimental uncertainties" $(u_t)_j$ and $(u_x)_i$ (i = 1,…, m) were assumed equal to the corresponding uncertainties reported in Tab. 3 (which were used to corrupt the data generated by the cycle program), while $\vartheta_{ij} = \partial t_j / \partial x_i$ were evaluated numerically by using the cycle program.

Moreover, the combined uncertainties $(U_y)_j$ were considered equal to the corresponding *RMSE* made by a NN, trained with data without measurement error, in the simulation of data without measurement error, when the number of patterns tends to infinity. In fact, for a infinite set of patterns, *RMSE* accounts for all the errors made in the calculation of predicted values y_j, which are the simulation numerical solution uncertainty U_{yN}, the simulation modeling uncertainty arising from the use of previous experimental data (which, in this case, are the training data) U_{yPED} and the simulation modeling uncertainty arising from modeling assumptions (such as, NN structure, number of hidden layers and of neurons of each hidden layer, training algorithm, etc.) U_{yMA}.

In particular, for the estimation of the combined uncertainties $(U_y)_j$ associated with each output computed by the NN, the corresponding *RMSE* was calculated, using test sets with a number of patterns ranging from 1 000 to 30 000. In Fig. 7 the *RMSE* value versus the number of test patterns is reported in the case of compressor outlet pressure p_{oC}. It can be observed that the trend is asymptotic: such an asymptotic trend against the number of test patterns can be observed for all the NN outputs. Therefore, the asymptotic values of *RMSE* were assumed as an estimation of $(U_y)_j$.

In Tab. 4 the estimated values of $(U_y)_j$ and the computed values of $(U_e)_j$ for the three considered instrumentation categories considered (laboratory, standard and industrial) are reported. It can be noted that $(U_y)_j$ values are usually about one order of magnitude lower than the corresponding $(U_e)_j$, which means that $(U_e)_j$ values are of the same order of magnitude as the corresponding $(U_t)_j$.

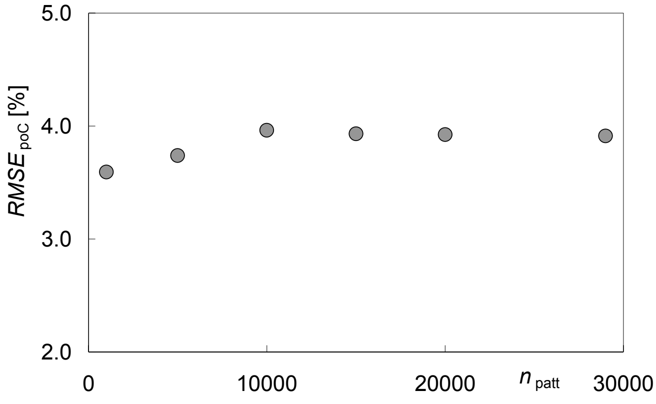


Figure 7 – *RMSE* for compressor outlet pressure p_{oC} versus the number of training patterns

Table 4 - Estimated $(U_y)_j$ values and $(U_e)_j$ computed values for the three considered instrumentation categories

Quantity	U_y	$U_e = \sqrt{U_t^2 + U_y^2}$		
		Laboratory	*Standard*	*Industrial*
p_{oC} [kPa]	3.9	29.2	42.3	64.0
T_{oC} [K]	1.2	5.8	7.9	12.1
M_f [%]	0.3	1.7	4.1	6.8
T_{oT} [K]	1.0	11.7	17.5	27.2
M_{oT} [%]	0.3	1.3	2.3	3.7
P_{GT} [%]	0.4	3.5	5.1	7.8

For the NN model validation it is considered that a success in the simulation of a computed output y_j is achieved when it results that:

$$\left|e_y\right|_j = \left|t - y\right|_j \le \left(U_e\right)_j = \sqrt{\left(U_t^2 + U_y^2\right)_j} \quad , \quad j = 1,\dots,n_o \qquad [9]$$

The success rate (*SR*) for each NN output, defined as the percentage ratio between the number of successes and the number of test patterns n_{patt}, is reported in Fig. 8. In particular, Fig. 8a reports the *SR* for a NN trained with data not affected by measurement uncertainty, while Fig. 8b reports the *SR* for a NN trained with data corrupted by a "standard"-type measurement uncertainty.

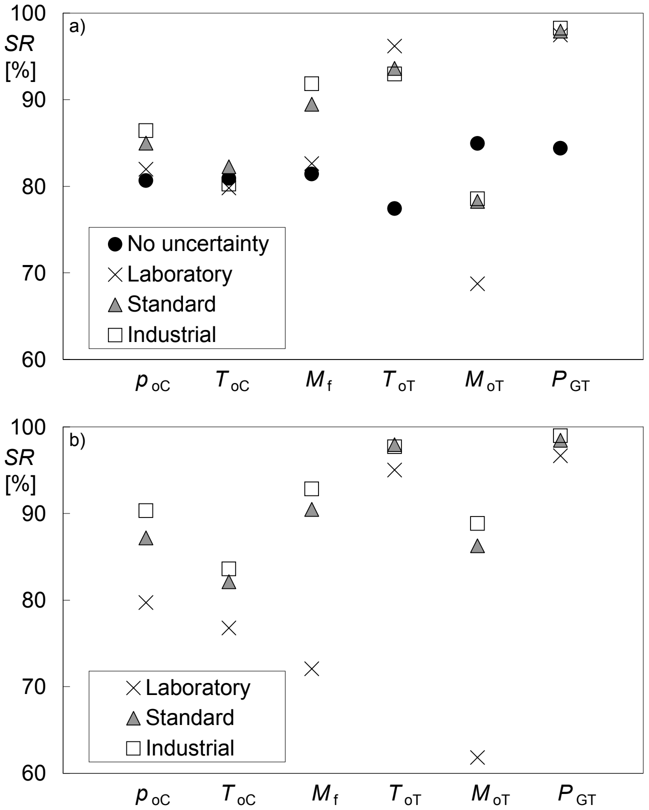


Figure 8 – *SR* for each NN output in the case of NN trained on data not affected by uncertainty (a) and corrupted by a "standard"-type measurement uncertainty (b)

From Fig. 8a, it can be observed that the *SR* for all quantities is usually higher than 80%, except for M_{oT} in the case of data corrupted with a "laboratory"-type measurement uncertainty. Moreover, it can be noted how *SR* increases when the measurement uncertainty increases, since in Eq. (9) $(U_e)_j$ increases with measurement uncertainty. In particular, in the case of data without measurement uncertainty, the *SR* is usually lower than in the other case since, here, in Eq. (9) $(U_t)_j = 0$ and, therefore, $(U_e)_j = (U_y)_j$. From Fig. 8b, it can be observed that the use of corrupted training data allows one to obtain a higher *SR* than in the case in which training data without uncertainty are used. This only applies for test data with an uncertainty equal or higher than that used for training.

An overall success rate (*OSR*) was also calculated. In this case, a simulation is considered a success when all the NN computed outputs verify Eq. (9). The results are reported in Tab. 5 and confirm the result reported in Fig. 8 (a) and (b), i.e. the use of corrupted training data allows one to obtain a higher *SR* than in the case in which training data without uncertainty are used. However, it should be remembered that this is only true for test data with an uncertainty equal or higher than that used for training.

Table 5 – Overall Success Rate for a NN trained with data (a) without uncertainty and (b) corrupted by a "standard"-type measurement uncertainty in the simulation of data corrupted by random errors

		OSR [%]	
	Laboratory	*Standard*	*Industrial*
Training data without uncertainty	46	52	58
Training data with "standard" meas. unc	34	62	71

CONCLUSIONS

In the paper, NN models for the simulation of gas turbines have been studied and developed. The analyses performed have been focused on the determination of the most appropriate NN structure, in terms of both computational time of the NN training phase and accuracy and robustness against measurement uncertainty. The NNs were used for the simulation of a 255 MW single shaft gas turbine at different boundary, load and health state conditions. For the considered 15 inputs/6 outputs system, the most appropriate NN structure proved to be a feed-forward multilayer perceptron with a single 60-neuron hidden layer and trained with at least 2 000 training patterns. In particular, the developed NN can be an effective tool for the real-time simulation of gas turbines when information on the behavior of the machine at different boundary, load and health state conditions is required.

ACKNOWLEDGMENTS

The work was carried out with the support of the M.U.R.S.T. (Italian Ministry of University and Scientific & Technological Research).

REFERENCES

[1] Glassman, A. J., 1974, "Computer Program for Thermodynamic Analysis of Open-Cycle Multishaft Power System with Multiple Reheat and Intercool", NASA Technical Note NASA TN D-7589.

[2] Stecco, S. S., Manfrida, G., Galletti, A., 1985, "Gas Turbines in Cogeneration: Overall Analysis and Numerical Predictions", *Proc. IGTI Beijing International Gas Turbine Conference,* Beijing, China.

[3] Thermoflow Inc., 1989, "GTMASTER: Interactive Software for Design & Performance Analysis of Gas Turbine Power & Cogeneration Systems", Wayland, MA, USA.

[4] Perz, E., 1991, "A Computer Method for Thermal Power Cycle Calculation", ASME J. of Eng. for Gas Turbines and Power, **113**, pp. 184-189.

[5] Cerri, G., 1996, "A Simultaneous Solution Method Based on a Modular Approach for Power Plant Analyses and Optimized Designs and Operations", ASME Paper 96-GT-302.

[6] Cerri, G., Monacchia, S., Seyedan, B., 1999, "Optimum Load Allocation in Cogeneration Gas-Steam Combined Plants", ASME Paper 99-GT-6.

[7] Bianchi, M., Gadda, E., Peretto, A., 2000, "Computational Code for the Management of a Deregulated Power Generation System", *Proc. International Mechanical Engineering Congress and Exposition of ASME*, Orlando, FL, USA, November 5 - 10, AES – Vol. 40, pp. 561-568.

[8] Kanelopoulos, K., Stamatis, A., Mathioudakis, K., 1997, "Incorporating Neural Networks into Gas Turbine Performance Diagnostics", ASME Paper 97-GT-35.

[9] Embrechts, M., Schweizerhof, A. L., Bushman, M., Sabetella, M. H., "Neural Network Modeling of Turbofan Parameters", ASME Paper 2000-GT-0036

[10] Boccaletti, C., Cerri., G., Seyedan, B., 2000, "A Neural Network Simulator of a Gas Turbine with Waste Heat Recovery Section", ASME Paper 2000-GT-185.

[11] Lazzaretto, A., Toffolo, A., Boni, A., 2001, "Gas Turbine Design and Off-Design Simulation Model: Analytical and Neural Network Approaches", *Proc. ECOS'01*, Instambul, Turkey, 4 – 6 July.

[12] Torella, G., Lombardo, G., 1996, "Neural Networks for the Diagnostics of Gas Turbine Engines", ASME Paper 96-TA-39.

[13] Romessis, C., Stamatis, A., Mathioudakis, K., 2001, "A Parametric Investigation of the Diagnostic Ability of Probabilistic Neural Networks on Turbofan Engines", ASME Paper 2001-GT-0011.

[14] Bettocchi, R., Spina, P. R., Torella, G., 2002, "Gas Turbine Health Indices Determination by Using Neural Networks", ASME Paper GT-2002-30276.

[15] Arriagada, J., Genrup, M., Loberg, A., Assadi, M., 2003, "Fault Diagnosis System for an Industrial Gas Turbine by Means of Neural Networks", *Proc. International Gas Turbine Congress 2003 (IGTC'03)*, Tokyo, Japan, November 2 -7, Paper IGTC2003Tokyo TS-001.

[16] Desideri, U., Fantozzi, F., Bidini, G., Mathieu, P., 1997, "Use of Artificial Neural Networks for the Simulation of Combined Cycles Transients", ASME Paper 97-GT-442

[17] Chiras, N., Evans, C., Rees, D., 2002, "Nonlinear Gas Turbine Modeling Using Feedforward Neural Networks", ASME Paper GT-2002-30035

[18] Bettocchi, R., Pinelli, M., Venturini, M., Spina, P. R., Bellagamba, S., Tirone, G., 2002, "Procedura di Calibrazione del Programma per la Diagnosi Funzionale dei Turbogas della

Centrale a Ciclo Combinato di La Spezia", *Proc. 57° Congresso Nazionale ATI*, Pisa, Italy (in Italian).

[19] Coleman,H. W., Stern, F., 1997, "Uncertainties and CFD Code Validation", ASME J. of Fluid Engineering, **119**, pp. 795-803.

[20] Stamatis, A., Mathioudakis, K., Papailiou, K.D., 1990, "Adaptive Simulation of Gas Turbine Performance", ASME J. of Eng. for Gas Turbines and Power, **112**, pp. 168-175.

[21] Bettocchi, R., Spina, P. R., 1999, ""Diagnosis of Gas Turbine Operating Conditions by Means of the Inverse Cycle Calculation", ASME Paper 99-GT-185.

[22] Pinelli, M., Spina, P. R., Venturini, M., 2003, "Optimized Operating Point Selection for Gas Turbine Health State Analysis by Using a Multi-Point Technique", ASME Paper GT2003-38191.

[23] Pinelli, M., Venturini, M., 2002, "Application of Methodologies to Evaluate the Health State of Gas Turbines in a Cogenerative Combined Power Plant", ASME Paper GT-2002-30248.

[24] Haykin, S., 1999, *Neural Networks - A Comprehensive Foundation*, 2nd Ed., Prentice Hall International, USA.

[25] Cybenko, G., 1989, "Approximation by Superimposition of a Sigmoidal Function", Mathematics of Control, Signals and Systems, **2**, pp. 303-314.

[26] Rumenlhart, D. E., Hinton, G. E., Williams, R. J., 1986, "Learning Internal Representation by Error Propagation, Parallel Distributed Processing", *Explorations in the Microstructure of Cognition*, **1**, pp. 318-362.

[27] Bettocchi R., Pinelli M., Spina P. R., Venturini M., Burgio, M., 2003, "Modelli Neurali per la simulazione di Turbogas", *Proc. 58° Congresso Nazionale ATI*, Padova, Italy (in Italian).