Optimization of a Real-Time Simulator Based on Recurrent Neural Networks for Compressor Transient Behavior Prediction

M. Venturini
Engineering Department in Ferrara (ENDIF),
University of Ferrara,
Via Saragat, 1,
44100 Ferrara, Italy

In this paper, feed-forward recurrent neural networks (RNNs) with a single hidden layer and trained by using a back-propagation learning algorithm are studied and developed for the simulation of compressor behavior under unsteady conditions. The data used for training and testing the RNNs are both obtained by means of a nonlinear physics-based model for compressor dynamic simulation (simulated data) and measured on a multistage axial-centrifugal small-size compressor (field data). The analysis on simulated data deals with the evaluation of the influence of the number of training patterns and of each RNN input on model response, both for data not corrupted and corrupted with measurement errors, for different RNN configurations, and different values of the total delay time. For RNN models trained directly on experimental data, the analysis of the influence of RNN input combination on model response is repeated, as carried out for models trained on simulated data, in order to evaluate real system dynamic behavior. Then, predictor RNNs (i.e., those that do not include among the inputs the exogenous inputs evaluated at the same time step as the output vector) are developed and a discussion about their capabilities is carried out. The analysis on simulated data led to the conclusion that, to improve RNN performance, the adoption of a one-time delayed RNN is beneficial, with an as-low-as-possible total delay time (in this paper, 0.1 s*) and trained with an as-high-as possible number of training patterns (at least 500). The analysis of the influence of each input on RNN response, conducted for RNN models trained on field data, showed that the single-step-ahead predictor RNN allowed very good performance, comparable to that of RNN models with all inputs (overall error for each single calculation equal to 1.3% and 0.9% for the two test cases considered). Moreover, the analysis of multi-step-ahead predictor capabilities showed that the reduction of the number of RNN calculations is the key factor for improving its performance over a significant time horizon. In fact, when a high test data sampling time is chosen (in this paper,* 0.24 s*), prediction errors were acceptable (lower than* 1.9%*).* [DOI: 10.1115/1.2437232]

Introduction

The interest shown in recent years with respect to the potential contributions that artificial intelligence techniques can provide to the modeling of energy systems and turbomachines mainly lies in the fact that the relative improvement that such techniques offer usually appears to be more pronounced as the complexity of a problem increases [1]. This effort is demonstrated by the great amount of work reported in recent literature dealing with the development of a particular type of tools based on artificial intelligence techniques, such as neural network (NN) models, for different fields of applications of gas turbines.

A first example is that of gas turbine diagnostics in steady-state conditions, for which neural network models have been employed alone [2–6], in combination with other techniques [7], or coupled with alternative artificial intelligence-based methods, such as expert systems [8], genetic algorithms [9], and fuzzy logic systems [10].

Another field of interest is that of sensor fault detection since a system model is usually required and implies a system identification process: this traditionally proves to be one of the preferred areas of expertise of neural networks [11]. Moreover, a demonstration of NN capabilities in detecting sensor faults, through the prediction of the complete engine operating parameters with few variables and the estimation of some nonmeasurable parameters, is shown in [12]. Furthermore, neural networks are still demonstrating their rich potential in the simulation of transient data.

The traditional approach to develop a dynamic simulation code is usually based on the use of the laws of conservation and of the performance maps of the considered machine [13–22]. Though the physics-based approach offers some strong points, such as the possibility to correctly explain the results by analyzing the physics of the considered processes, the development of physics-based dynamic simulation models may present some problems [23]. In fact, the capability of the model to reproduce the real machine is affected by the quality of the calibration process, which may be difficult to perform with a high level of accuracy, since some model parameters have to be estimated and/or the machine performance maps may not be available [20,21]. Moreover, computational times for complex systems may be so high as to limit their use in on-line applications, though some solutions were proposed in [24,25].

Thus, neural networks offer an alternative solution for developing dynamic models because of the capability of easily modeling an even very complex system due to their self-adapting characteristics. Moreover, they usually present good generalization capabilities even with a reduced set of identification data, robustness in

Contributed by the International Gas Turbine Institute of ASME for publication in the JOURNAL OF TURBOMACHINERY. Manuscript received May 26, 2006; final manuscript received May 31, 2006. Review conducted by David Wisler. Paper presented at the ASME Turbo Expo 2006: Land, Sea and Air (GT2006), May 8, 2006–May 11, 2006, Barcelona, Spain. Paper No. GT2006-90117.

 Copyright © 2007 by ASME

Downloaded From: http://turbomachinery.asmedigitalcollection.asme.org/ on 08/28/2018 Terms of Use: http://www.asme.org/about-asme/terms-of-use

the presence of poor and/or noisy input data [26], and high computational speed, which proves crucial mainly when they are used for real-time simulations.

Neural networks also have the ability to incorporate system aging effects by performing their own on-line training [27]. Finally, NNs have also demonstrated their effectiveness for control purposes in cases in which the traditional control techniques were not accurate enough to ensure precise or even safe control [28].

The well-known limit of these models is high prediction error when they operate outside the field in which they were trained, i.e., they are not able to extrapolate. Therefore, great attention has to be paid to selecting NN training data correctly, in order to (i) cover all the possible range of variation of the considered process variables and (ii) set up a system model that is accurate enough in the operating region of interest, since inaccuracies can negatively influence the model reliability.

Hence, NN models for energy system simulation in unsteady conditions have been developed and some examples of their application are reported below in chronological order:

- prediction of the most significant quantities on a heat exchanger used as steam generator [1,29],
- simulation of the response of a combined cycle power plant during a slow transient [30],
- development of a nonlinear gas turbine model [31],
- setup of an observer to predict the air-fuel ratio for a spark ignition engine [32],
- development of a fault detection and diagnosis system for a turbofan engine through "nested NNs" [33] and for a field-operated propulsion engine through "echo state networks" [34], and
- simulation of transient behavior of a small-size axial-centrifugal compressor [23].

In this paper, NN models (feed-forward NNs with a single hidden layer, trained by using a back-propagation learning algorithm) for the simulation of compressor behavior under unsteady conditions are studied and developed. In order to develop a NN model capable of reproducing time-dependent data, neural networks in a so-called recursive computational structure, which makes use of one feedback loop (recurrent neural networks (RNNs), are usually adopted in practice [1,23,29,32,35]. Thus, they are also investigated in this paper.

The data used for training and testing the NNs were obtained by means of a nonlinear modular model for compressor dynamic simulation [21] (simulated data) and also measured on a multistage axial-centrifugal small-size compressor running in the test facility of the University of Ferrara [36] (field data). In particular, note the following:

- The nonlinear model, developed through a physics-based approach, was calibrated on the axial-centrifugal compressor for which the field data were available and was validated through the same experimental data [21]. The generated data represent different variations in the compressor operating condition and are the same as already used in [23].
- The field data are the same as already used for the physics-based dynamic model validation [21] and also for testing the capabilities of the RNN models developed in [23].

The first step of the analysis is the completion of the sensitivity analyses conducted in [23] on simulated data. In particular, starting from the best model setup in that same paper, the influence of (i) the number of training patterns and (ii) each RNN input on model response is evaluated both on data that are uncorrupted and corrupted with measurement errors. Different RNN configurations and total delay time values are analyzed in the current paper as before [23].

Then, RNN capabilities are tested against measured values by training the RNN models directly on field operating data. First, the analysis of the influence of each RNN input on model response is repeated for models trained on field data as previously done for models trained on simulated data in order to evaluate real system dynamic behavior.

Second, predictor RNNs (i.e., those that do not include among the inputs the vector of exogenous inputs evaluated at the same time step as the output vector) are developed and a discussion about their capabilities is carried out by taking into account the main factors that affect their overall performance.

Thus, with respect to [23], which represented the first step developed by the author in applying blackbox models for the simulation of transient data, the current paper investigates two novel aspects in the same field of dynamic simulation by means of neural networks, i.e., the development of RNNs (i) trained on field data and (ii) that act as predictors, in the same way as in [29]. In fact, the use of experimental dynamic data for RNN training and testing is reported in few papers (three years of operating data obtained from a three-spool RB211 driven compressor station in [12]; real data collected from a field-operated propulsion engine during 15 separate takeoffs in [34]; measured variables on a Rolls Royce turbofan engine in [31]). This is because NN dynamic models are usually applied to artificial problems by means of data obtained through system simulation models [23,28–30,34].

The last remark deals with the considered compressor and the dynamic model calibrated on it. It was highlighted in [21] that the deviation of system dynamics from its stationary behavior is small since the physical system is characterized by small size volumes and, thus, small time constants. This feature does not affect the validity of the results presented in the paper, which instead can be useful for applying, for instance, to turbomachines characterized by small size volumes (e.g., microturbines). In fact, such turbines, which are becoming very widespread, usually run under unsteady-state conditions and operate in energy systems that can be quite complex, due to the presence of several additional system components (such as external burners, fuel cells, etc.) [37,38]. Since a dynamic simulation model is always required in practice (for purposes of modeling, control, performance evaluation, etc.), system dynamic models based on RNNs can represent an interesting solution for modeling this kind of energy system.

Recurrent Neural Networks

Artificial neural networks (NNs) are mathematical structures that are able to link, in a nonlinear way through several interconnected simple units (the artificial neurons), a multidimensional input space with a multidimensional output space, allowing very high computational speed. A detailed description of NN features and capabilities can be found in specific literature, such as, among others, [35], and in papers illustrating NN applications to different fields [1,6,10–12,23,26,28–34]. For this reason, only the assumptions adopted in the paper for the NN model setup will be outlined, with particular reference to the development of recurrent neural networks.

The problem of developing a NN model capable of reproducing time-dependent data consists on the fact that the NN model has to take into account the time variable by means of a memory process. This can be done through NN architectures characterized by feedback connections among the neurons. An exhaustive review of different NN dynamic models structures can be found in [31,32,35].

In the paper, feed-forward neural networks, which approximate a nonlinear dynamic system by means of a static mapping, are developed, as also adopted in [1,23,28]. However, they can perform an inherently nonlinear dynamic mapping by using local (i.e., in the recurrent hidden layer nodes) and/or global (use of time-delayed variables as inputs) feedbacks, which ensure model dynamics.

According to [23,29–32], recurrent neural networks (RNNs) with one global feedback loop in a recursive computational structure are considered. Recurrent networks are also referred to as

Downloaded From: http://turbomachinery.asmedigitalcollection.asme.org/ on 08/28/2018 Terms of Use: http://www.asme.org/about-asme/terms-of-use

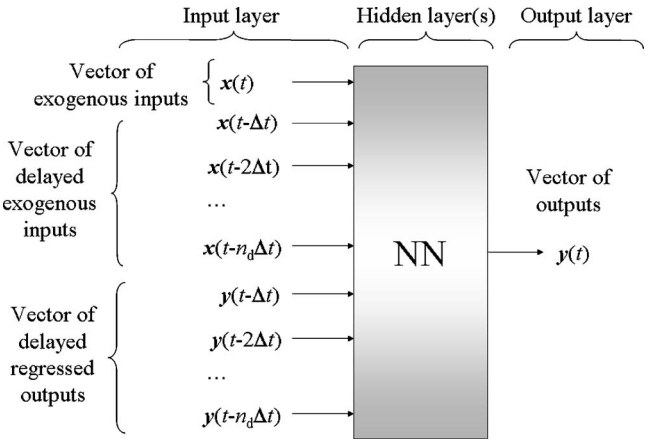

Fig. 1 Scheme of a recurrent neural network (RNN) model

nonlinear autoregressive with exogenous inputs models (NARX). In fact, it can be demonstrated that the role of the delay number in a delay feedback model is similar to that of the system order in a NARX model.

The basic topology of an RNN is shown in Fig. 1, where Δt is the time delay unit. From the scheme reported in Fig. 1, it can be observed that:

- the kernel of the model is the "classical" neural network, the only conceptual difference being (i) the presence of delayed values of the inputs and (ii) the regression of the outputs.
- the vector of exogenous inputs is evaluated at time t (the same as output vector $\boldsymbol{y}$). This condition is not necessary, since RNNs in which the output vector is ahead of the input vector by at least one time unit are presented in [29,35] and are also investigated in the paper. It should be remembered that all the models developed in [23] include the exogenous inputs evaluated at the current time step among the inputs.
- the total delay time of the vector of delayed exogenous inputs and, similarly, of the vector of delayed regressed outputs is $n_d\Delta t$. The condition that RNN regressed outputs (i) start from the time step just before the current time step (t-Δt) and (ii) are delayed up to the same time step as the delayed exogenous inputs (t-$n_d\Delta t$), is not a general requirement. For a discussion about this aspect, the reader is referred to [23,35]. In any case, this represents the most common solution [23,29].

Thus, as seen in Fig. 1, the input layer of the RNN usually consists of two main classes of inputs: (i) exogenous inputs, i.e., originating from outside the network and (ii) delayed values of outputs. Exogenous inputs can be evaluated both at the same time step as the outputs and at antecedent time steps with respect to the outputs. On the other hand, the delayed values of outputs can be either taken from outside the RNN (for example, measured observations or data estimated through a different model) or estimated through the RNN itself at antecedent time steps. In such a way, model outputs are *regressed*. In particular, neural networks, for which the vector of inputs is composed of (i) the exogenous inputs evaluated at antecedent time steps with respect to the outputs and (ii) delayed values of outputs estimated through the RNN itself at antecedent time steps, are identified as predictors and are of great interest since, due to the high computational speed of RNNs, they can be employed as real-time simulators.

Because of the above-mentioned considerations, the dynamic behavior of the RNN model is described by the following equation, where F is a nonlinear function of its arguments:

$$\boldsymbol{y}(t) = F[\boldsymbol{x}(t), \boldsymbol{x}(t-\Delta t), \ldots, \boldsymbol{x}(t-n_d\Delta t), \boldsymbol{y}(t-\Delta t), \ldots, \boldsymbol{y}(t-n_d\Delta t)] \tag{1}$$

The transfer function F depends on the chosen activation function (in this paper, sigmoidal) and on the network parameters (i.e., number of inputs, number of neurons in the hidden layers, and connection weights among the neurons). In particular, the proper connection weights can be identified through the optimization of the training process. Thus, according to Eq. (1), the RNN model implicitly takes into consideration the time variable by means of the dependence-with-time of all variables [1,23,29,32,34,35].

The architecture chosen in the paper is the typical *feed-forward multilayer perceptron* (also adopted in [21,28,29]). In particular, NNs with one hidden layer and a continuous sigmoid activation function are used, since it has been shown that this type of NN architecture is able to represent any type of multidimensional nonlinear function, if a suitable number of neurons in the hidden layer is chosen [35,39]. The use of multiple hidden layers usually requires a great computational effort, while it only allows a small improvement of performance [35], and thus architectures with only one hidden layer are usually adopted in practice [23,26,29,31].

With regard to the choice of the suitable training algorithm for RNN models, a good overview is reported in [40]. In the paper, a traditional *back-propagation* algorithm was used and the TRAINSCG algorithm was adopted. in fact, this algorithm appeared to be effective with a low computational effort [23,26,30] and, above all, it was available in the MATLAB® Neural Network toolbox.

The adopted stopping criterion for the NN training phase is the minimization of a performance function that was chosen to be the mean-square error (MSE) on the whole training set between the target outputs and the corresponding NN computed outputs:

$$\mathrm{MSE} = \frac{1}{n_o n_{\mathrm{patt}}} \sum_{j=1}^{n_o} \sum_{i=1}^{n_{\mathrm{patt}}} [t_{ij} - y_{ij}]^2 \tag{2}$$

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.

In this paper, only off-line NN training is performed. This means that a set of training data is collected and the NN model is constructed from this set of data by computing the best-fitting weight parameter combination that minimizes a cost function (in the paper, the MSE). However, a system that is subject to time-varying disturbance or drifting can never generate the same trajectory twice. In this case, it would be advisable to perform an on-line training: this results in a neural network that can be tuned as the system runs, i.e., whose hidden weights are adjustable. This solution is studied, also theoretically, in [27]. In any case, the commonly adopted solution for RNN training is the off-line training [23,28,29].

Data for NN Training and Testing

The test facility in which the compressor under consideration is installed has been widely addressed in [21,23,36] and, thus, is only briefly described in this paper. The experimental apparatus is composed of an inlet duct (in which an orifice plate is mounted to perform the inlet mass flow measurement), a compressor (which is part of the Allison 250-C18 turboshaft engine and is composed of six axial stages and one centrifugal stage), and an outlet duct, which plays the role of a small-volume plenum in which a butterfly valve is inserted for compressor mass flow rate control. Thus, the compressor operates in an open circuit and can be driven by an electric motor up to 30,000 rpm, by means of a step-up gear box.

Compressor Physics-Based Dynamic Model for Data Generation. A nonlinear modular model for dynamic simulation of compressors was set up and implemented in MATLAB environment by means of the Simulink tool [21]. The model was devel-

Downloaded From: http://turbomachinery.asmedigitalcollection.asme.org/ on 08/28/2018 Terms of Use: http://www.asme.org/about-asme/terms-of-use

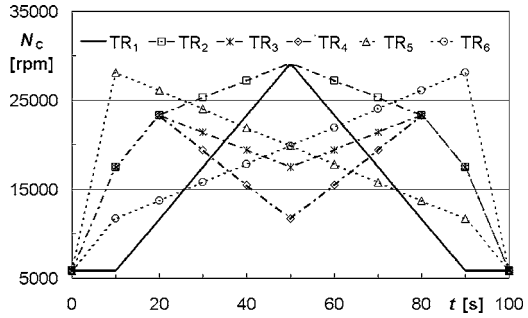

Fig. 2 Compressor rotational speed trend versus time for simulated data used for RNN training (TR_1) and testing (TR_2-TR_6)

oped through a physics-based approach by using the laws of conservation and heat balances, and by using the performance maps of the considered compressor. The model reproduces the test facility by means of three modules: the intake duct (sections 0-1), the compressor (sections 1-2), and the exhaust duct (sections 2-3). The model has three inputs (ambient pressure p_0 and temperature T_0 and compressor rotational speed N_C), and estimates the variables for all model sections, among which the ones considered in the paper are compressor inlet mass flow rate M_1 and pressure p_1 and compressor outlet pressure p_2 and temperature T_2.

The physics-based dynamic model was calibrated on the considered compressor and validated through experimental data [21]. It was observed that, though the approach adopted for developing the physics-based model is general, the deviation of system dynamics from its stationary behavior is small since (i) the model only captures dynamic effects deriving from mass storage and thermal exchange phenomena (in fact, the shaft dynamics is not taken into account in the model, though it can be easily implemented as shown in [22]) and, above all, (ii) the physical system is characterized by small size volumes.

Simulated Data Generation. The physics-based dynamic model has been used to generate different transients that have already been presented in [23]. The simulated maneuvers reproduce accelerations and decelerations, all taken at ISO conditions ($T_0=15°C$; $p_0=101.3$ kPa): some of them were used for RNN training, while the rest were used to verify the generalization capabilities of the trained RNNs. One of the results obtained through [23] was the identification of the optimal maneuver for RNN training; thus, this curve (triangular shaped, symmetrical with respect to the total time interval, and with a stationary phase at the beginning and at the end) is adopted here for training all the developed RNNs trained on simulated data. For the sake of readability, the simulated curves are also reported in this paper: TR_1 is the curve used for RNN training, while TR_2 through TR_6 are used to verify the generalization capabilities of the trained RNNs.

In Fig. 2, the trend over time of the compressor rotational speed (which is the only input for the compressor dynamic model, once ambient conditions T_0 and p_0 are fixed) is plotted for a time interval of 100 s for each transient TR. All the considered curves cover a range of variation for compressor rotational speed from ~5,000 rpm up to ~30,000 rpm.

The curves used for RNN testing (TR_2-TR_6) differ from each other, since compressor rotational speed varies:

- quite slowly and symmetrically with respect to the total time interval (TR_2-TR_4) and
- quite rapidly, as a fast ramp-shaped curve, and in an asymmetrical way (TR_5 and TR_6).

Finally, it should be specified that for all the generated data, the sampling time Δt_s was equal to 0.1 s; this means that (i) the time delay unit Δt reported in Eq. (1) has also been considered equal to 0.1 s and (ii) each curve contains 1000 data patterns. Furthermore, it should be noted that the developed RNNs have all been tested on patterns (derived from TR_2-TR_6) that do not include the patterns used for training (derived from TR_1).

Table 1 Measurement uncertainty values

Quantity	Absolute uncertainty	Value at $N_C=28{,}948.0$ rpm	Percentage uncertainty (%)
N_C	30.0 rpm	28 948.0 rpm	0.10
p_1	0.636 kPa	98.36 kPa	0.65
p_2	1.817 kPa	165.83 kPa	1.10
M_0	0.012 kg/s	0.526 kg/s	2.28
T_2	1.0 K	374.3 K	0.27

Simulated Data Corruption Through Measurement Errors. To take into account the presence of measurement errors, the generated data were corrupted with random errors included in the measurement uncertainty intervals reported in Table 1. The estimation of the absolute uncertainties for the measured quantities, reported in the second column of Table 1, was performed by using the results of the uncertainty analysis conducted in [36], which was also adopted in [23]. The percentage uncertainty values (fourth column of Table 1) are calculated with respect to the values of the quantities in correspondence to the rotational speed for which the uncertainty analysis was conducted (i.e., 28,948 rpm).

Available Measured Data. The available measured data consist of two test cases (TC1 and TC2), both taken at quasi-ISO conditions ($T_0\approx17°C$; $p_0\approx102$ kPa), which represent acceleration and deceleration maneuvers for the compressor on which the physics-based model was calibrated [21].

The same two curves were also adopted in [23] for testing the developed RNNs trained on simulated data. The two curves are reported in Fig. 3 and differ from each other since TC1 curve covers a wider range of variation for compressor rotational speed (N_C ranges from ~13,000 up to 23,000 rpm), while TC2 curve covers a more restricted region of N_C values (from ~20,000 up to 26,000 rpm). In any case, the rotational speed gradient is comparable for both curves (i.e., ~200 rpm/s).

With respect to simulated data, it can be observed that (i) the range of variation for measured compressor rotational speed N_C lies within the range of variation of compressor rotational speed values in the simulated data (assumed to be in the range of 5000–30,000 rpm) and (ii) the rotational speed gradient for field data is lower than that of the TR_1 curve (~600 rpm/s).

The available field data are characterized by a sampling time equal to 0.0067 s and, thus, by a number of patterns equal to 27,150 (TC1) and to 9450 (TC2). The patterns that will be used in

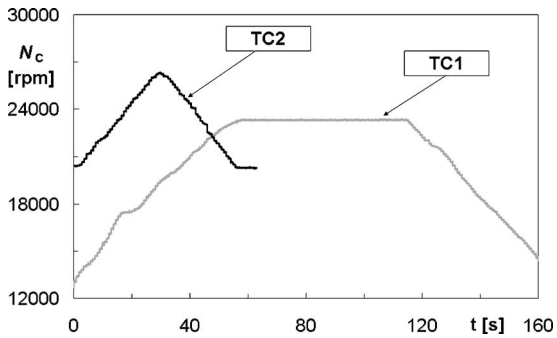

Fig. 3 Rotational speed trend versus time (measured data) for the two test cases TC1 and TC2

Downloaded From: http://turbomachinery.asmedigitalcollection.asme.org/ on 08/28/2018 Terms of Use: http://www.asme.org/about-asme/terms-of-use

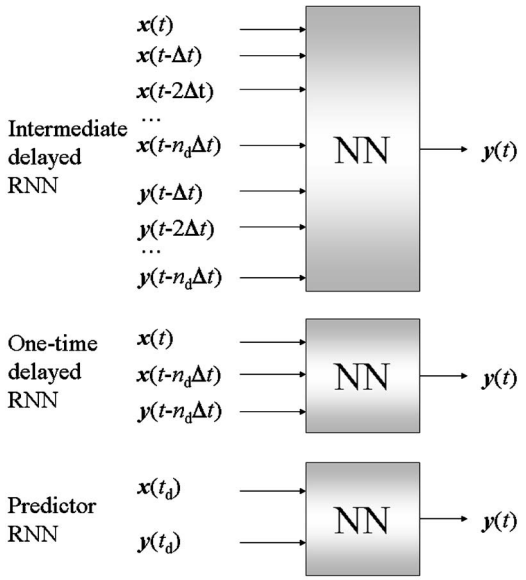

Fig. 4 Scheme of the recurrent neural network models adopted in the paper

the next section for training the RNNs on field data were derived from the available measured data, by uniformly sampling all the originally available 36,600 patterns.

RNN Model Identification

RNN Inputs and Outputs. The RNN models developed in the paper are characterized by a single exogenous input (compressor rotational speed N_C) and four outputs (compressor inlet and outlet pressure p_1 and p_2, compressor inlet mass flow rate M_1 and compressor outlet temperature T_2). Thus, the exogenous input vector $\boldsymbol{x}$ is composed of only one element (i.e., $\boldsymbol{x}(t)=[(N_C(t)]$, and thus, the dimension of the vector is $d_x=1$), while the output vector $\boldsymbol{y}$ is made up of four elements, i.e., $\boldsymbol{y}(t)=[p_1(t),p_2(t),M_1(t),T_2(t)]$ and, thus, $d_y=4$. The choice of adopting a multiinput-multioutput architecture was made in accordance with [23].

RNN Architecture. Two different types of RNNs are developed from the general scheme presented in Fig. 1, as done in [23], and are sketched in Fig. 4:

- "Intermediate delayed" RNN: Inputs and outputs are regressed to the RNN input layer for all intermediate time steps up to the time $n_d\Delta t$. Thus, the total number of inputs is $n_i=d_x+n_d^*(d_x+d_y)$.
- "One-time delayed" RNN: the values of exogenous inputs and of the regressed outputs are both supplied to the RNN input layer only delayed of time $n_d\Delta t$. Thus, the total number of inputs is $n_i=(2d_x+d_y)$, i.e., six.

In this paper, different values of the total delay time for both intermediate delayed RNNs and one-time delayed RNNs are tested.

A third typology of RNNs can be obtained from both intermediate delayed and one-time delayed RNNs, i.e., the predictor RNN, for which an extensive survey can be found in [29]. A predictor RNN is characterized by the fact that it does not include among the inputs the vector of exogenous inputs evaluated at the same time step as the output vector, and thus, its output vector is ahead of the input vector by at least one time unit. For the predictor sketched in Fig. 4, a concise notation is adopted: $\boldsymbol{x}(t_d)$ identifies the vector of exogenous inputs evaluated at antecedent time steps, and $\boldsymbol{y}(t_d)$ represents the vector of delayed regressed outputs. The two vectors obviously differ if an intermediate delayed architecture or a one-time delayed architecture is considered.

Then, as a function of the forecasting horizon, two types of predictors can be distinguished:

- single-step-ahead (SS) predictor. The SS predictor RNN estimates the vector of outputs for one time step following the current time step of the inputs. In the paper, the vector of delayed regressed outputs $\boldsymbol{y}(t_d)$ is not calculated by the RNN, but is always supplied from outside (estimated through the physics-based model in the case of simulated data and measured in the case of experimental data).
- multi-step-ahead (MS) predictor. The MS predictor RNN estimates the vector of outputs for many time steps ahead and reaches the desired forecasting time step through a recursive use of an SS predictor. With the exception of the first time step, for which the predictor is obviously fed by a vector of delayed regressed outputs $\boldsymbol{y}(t_d)$ supplied from outside the RNN (estimated through the physics-based model in the case of simulated data and measured in the case of experimental data), the MS predictor is always fed by a vector of delayed regressed outputs $\boldsymbol{y}(t_d)$ estimated through the SS predictor at antecedent time steps. For this reason, the accuracy of the SS predictor is of utmost importance, since even small SS prediction errors at the beginning of the horizon accumulate and propagate, thus resulting in poor prediction accuracy [29].

Comments on Data Sampling Time, RNN Delay Time, and Number of RNN Calculations. The sampling time for training and test data (respectively, $(\Delta t_s)_{tr}$ and $(\Delta t_s)_{test}$) can be different. The only constraint is that the adopted delay time $n_d\Delta t$ must be the same for both training and test data, since the RNN needs to be fed with consistent data. Moreover, for an intermediate delayed RNN, the choice of the appropriate total delay time also has to take into account the availability of data in correspondence with the intermediate time steps. The relation between RNN total delay time and the sampling time of training and test data can be expressed as in Eq. (3), where k_{tr} and k_{test} are multiplicative factors,

$$n_d\Delta t = k_{tr}(\Delta t_s)_{tr} = k_{test}(\Delta t_s)_{test} \tag{3}$$

On the other hand, no relation exists between the time interval $(\Delta t_{tot})_{tr}$ available for extracting the training data and the time interval $(\Delta t_{tot})_{test}$ to be simulated. Available data sampling time may be recommended to have low values, since data with a higher sampling time (i.e., lower sampling frequency) can be obtained by performing a subsequent data postprocessing. In the paper, quite low sampling times are considered ($\Delta t_s=0.1$ s and $\Delta t_s=0.0067$ s for simulated and field data, respectively), while relatively higher sampling times can be found in literature ($\Delta t_s=5.0$ s in [29] and $\Delta t_s=0.25$ s in [34] for simulated and field data, respectively).

In order to maximize RNN accuracy at each time step (i.e., for a single-step-ahead predictor), an improvement of RNN performance can be obtained by a proper setup of the RNN architecture and configuration, as shown later in the paper through the sensitivity analyses conducted on both simulated and field data.

On the other hand, if an MS predictor RNN is considered, another possibility is offered through resetting the propagated error at fixed time steps. This can be done by supplying "good" data in correspondence with the selected reset time step. Thus, in this manner, the vector of delayed regressed outputs is no longer estimated by the RNN itself, but is provided from outside the RNN, as, for example, measured observations or data estimated through a different model. This possibility is investigated below in Fig. 8 for different values of the reset time step.

For an assigned time interval and a given sampling time Δt_s, the number of available patterns can be obtained as the ratio between the total time interval and the sampling time Δt_s. In particular, if

Downloaded From: http://turbomachinery.asmedigitalcollection.asme.org/ on 08/28/2018 Terms of Use: http://www.asme.org/about-asme/terms-of-use

one or more resets are performed during the time interval to be simulated $(\Delta t_{tot})_{test}$, the following relation can be written, by using Eq. (3),

$$(\Delta t_{tot})_{test} = \left[(n_{patt})_{test} \left(\frac{n_d \Delta t}{k_{test}} \right) \right] (n_{reset} + 1) \quad (4)$$

where n_{reset} is the number of resets performed in the given time interval to be simulated ($n_{reset}=0,1,2,\ldots$) and $(n_{patt})_{test}$ represents the number of patterns (and also the number of RNN calculations) between two subsequent resets.

Performance Evaluation of RNN Models. The parameter used for the comparison of the RNNs is the root-mean-square error RMSE, which is frequently adopted for this kind of RNN applications [23,26,29].

Such a parameter represents the error RMSE made by the RNN on the whole set of test data in the calculation of each jth output (p_1, p_2, M_1, and T_2) for any given kth transient used for RNN testing (TR_2–TR_6 for simulated data and TC1 or TC2 for field data),

$$\left(\mathrm{RMSE} = \sqrt{\frac{1}{n_{patt}} \sum_{i=1}^{n_{patt}} \left(\frac{t_i - y_i}{t_i} \right)^2} \right)_{jk} \quad \begin{matrix} j = 1, \ldots, n_o \\ k = 1, \ldots, n_{TR} \end{matrix} \quad (5)$$

where t_i are the target outputs, y_i the computed outputs, n_{patt} is the number of patterns for each transient used for RNN testing, n_o is the number of RNN outputs (i.e., four) and n_{TR} is the number of transients used for RNN testing (i.e., five in the case of simulated data and two in the case of field data).

Two different concise indices are then derived:

- $RMSE_{ov}$, which represents the overall RMSE made by the RNN on all outputs for each transient used for RNN testing and is defined as

$$\left(\mathrm{RMSE}_{ov} = \sqrt{\frac{1}{n_o} \sum_{j=1}^{n_o} (\mathrm{RMSE})_j^2} \right)_k, \quad k = 1, \ldots, n_{TR} \quad (6)$$

- $RMSE_m$, which represents the RMSE made by the RNN on all transients used for testing. In this case, the "mean" is referred to all transients and, thus, the parameter can be calculated for any jth single output, as in Eq. (7), and also as an overall value, as in Eq. (8), as

$$\left(\mathrm{RMSE}_m = \sqrt{\frac{1}{n_{TR}} \sum_{k=1}^{n_{TR}} (\mathrm{RMSE})_k^2} \right)_j, \quad j = 1, \ldots, n_o \quad (7)$$

$$(\mathrm{RMSE}_m)_{ov} = \sqrt{\frac{1}{n_o} \sum_{j=1}^{n_o} (\mathrm{RMSE}_m)_j^2} \quad (8)$$

RNN Sensitivity Analyses on Simulated Data

The starting point of the analyses that are developed in the paper is represented by the results obtained in [23]. In particular, all the RNNs developed for the analyses on simulated data are (i) characterized by 15 neurons in the hidden layer and by a multi-output architecture and (ii) are trained on TR_1 curve data. Two types of analyses are performed on simulated data: (i) determination of the proper number of RNN training patterns n_{patt} and, thus, of the associated data sampling time, and (ii) evaluation of the influence of each input on RNN response.

All the analyses, summarized in Table 2, are carried out for both intermediate delayed and one-time delayed RNNs. The analyses consider different values of the total delay time and different combinations of inputs. In all cases, the presence of measurement errors is taken into account.

Table 2 RNN models developed for the sensitivity analyses on simulated data

RNN architecture	Total delay time (s)	Number of training patterns	RNN inputs
Intermediate delayed	1.0 1.0, 0.5	100, 250, 500, 1000 1000	All Variable
One-time delayed	1.0, 0.2 1.0, 0.5, 0.1	100, 250, 500, 1000 1000	All Variable

Number of Training Patterns. The influence of the number of training patterns $(n_{patt})_{tr}$ on RNN performance is established by comparing the response of three different RNN models: one intermediate delayed RNNs ($n_d=10$, i.e., total delay time equal to 1.0 s) and two one-time delayed RNNs ($n_d=2,10$, i.e., total delay time equal to 0.2 s and to 1.0 s).

All the developed RNNs are compared by means of the parameter $RMSE_m$ for different values of the number of training patterns: 100, 250, 500, and 1000, this latter being the reference case as reported in [23]. When less than 1000 patterns are used, it was decided to uniformly sample the training curve, which was originally composed of 1000 patterns. The results are reported for two different cases: (i) absence of measurement errors, so that both training and test patterns are not corrupted with measurement errors, and (ii) presence of measurement errors both in training and test patterns, i.e., both training and test patterns are corrupted with measurement errors.

From Fig. 5, it can be noted that:

- for all RNN models, errors decrease as the number of training patterns increases, which depends on the chosen sampling time once the total time interval considered for training data is fixed. However, the solution of adopting 500 (or even 250 for one-time delayed networks) training patterns seems not to affect errors noticeably, while 100 training patterns should not be considered as a viable solution.
- independent of the number of training patterns, it is confirmed that the one-time delayed RNN with the lowest total delay time (i.e., 0.2 s) represents the best solution [23]. Moreover, for this latter RNN, the dependence of its performance with $(n_{patt})_{tr}$ becomes less evident.

Influence of RNN Inputs. The interest in evaluating the influence of RNN inputs on RNN performance lies in two aspects. First, the analysis allows the identification of which input is more significant so that the model with the best combination of inputs (i.e., the one that allows the lowest errors) can be adopted for further analyses. Second, it allows the user to understand which is the "order" of system dynamics. For instance, if the most signifi-

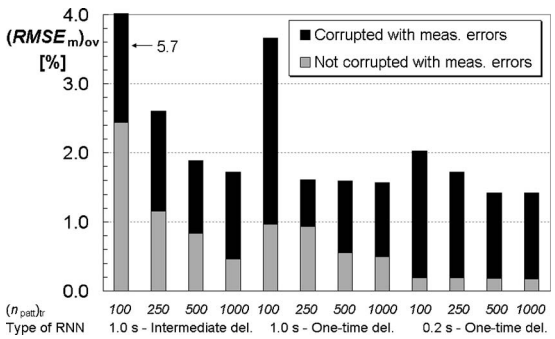

Fig. 5 Influence of the number of training patterns for intermediate delayed and one-time delayed RNNs in the absence of and in the presence of measurement uncertainty

Downloaded From: http://turbomachinery.asmedigitalcollection.asme.org/ on 08/28/2018 Terms of Use: http://www.asme.org/about-asme/terms-of-use

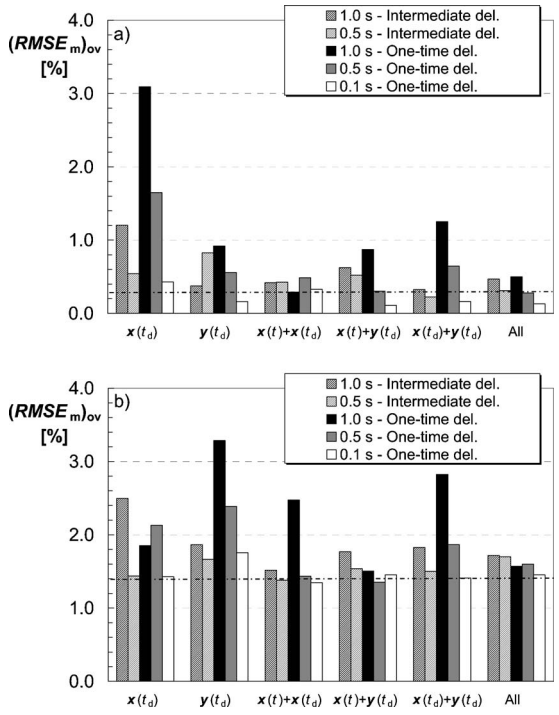

Fig. 6 Influence of RNN inputs for one-time delayed and intermediate delayed RNNs (*a*) in the absence of and (*b*) in the presence of measurement uncertainty; the dashed-dotted line stands for the $(RMSE_m)_{ov}$ value for the RNN with the only input $x(t)$

cant input proves to be the vector of exogenous inputs evaluated at the current time step, then it means that the system is characterized by poor dynamic effects since in this case a pseudostatic model should be adopted.

The influence of RNN inputs on RNN performance was established by comparing the response of five different RNN models: two intermediate delayed RNNs (n_d=5,10, i.e., total delay time equal to 0.5 s and 1.0 s) and three one-time delayed RNNs (n_d =1,5,10, i.e., total delay time equal to 0.1 s, 0.5 s, and 1.0 s). Then, for each of the five RNNs, seven different models with all the possible combinations of inputs have been considered ($\boldsymbol{x}(t)$, $\boldsymbol{x}(t_d)$, $\boldsymbol{y}(t_d)$, $\boldsymbol{x}(t)+\boldsymbol{x}(t_d)$, $\boldsymbol{x}(t)+\boldsymbol{y}(t_d)$, $\boldsymbol{x}(t_d)+\boldsymbol{y}(t_d)$, and $\boldsymbol{x}(t)+\boldsymbol{x}(t_d)+\boldsymbol{y}(t_d)$, this latter solution with all the possible inputs identified as "All"), where $\boldsymbol{x}(t)$ is the vector of exogenous inputs evaluated at the current time step and $\boldsymbol{x}(t_d)$ and $\boldsymbol{y}(t_d)$ are the vectors of exogenous inputs and of delayed outputs, respectively, evaluated at antecedent time steps. It is worth remembering that (i) vectors $\boldsymbol{x}(t_d)$ and $\boldsymbol{y}(t_d)$ are different if a one-time delayed RNN instead of an intermediate delayed RNN is considered and (ii) all the developed RNNs are trained by adopting 1000 training patterns.

The results of the analysis are reported in Fig. 6, in the case both of absence (Fig. 6(*a*)) and of presence (Fig. 6(*b*)), of measurement errors. In the Fig. 6, a dashed-dotted line is inserted to indicate the overall $RMSE_m$ in the case that the vector of exogenous inputs evaluated at the current time step $\boldsymbol{x}(t)$ is the only RNN input. In fact, such error is independent of the considered type of RNN and is equal to 0.3% or 1.4% when measurement errors are absent or present, respectively.

Some general considerations can be highlighted as follows:

- The overall results obtained in the case that both training and test patterns are not corrupted with measurement errors and in the case that both training and test patterns are corrupted with measurement errors substantially agree. In fact, the case in which both training and test patterns are not corrupted with measurement errors can be regarded as a lower limit for RNN simulation error and, thus, the presence of measurement errors can be taken into account through an additive term which only depends on the measurement noise level.
- The choice of the optimal architecture between one-time delayed RNNs and intermediate delayed RNNs depends on the considered combination of inputs. In any case, independent of the considered combination of inputs, the one-time delayed RNN with as low as possible total delay time (i.e., 0.1 s) almost always allows the lowest $RMSE_m$ overall values both for data that are corrupted or uncorrupted with measurement errors, independent of the considered combination of inputs, as already shown in [23,29]. Thus, since the difference between the various RNN models decreases as the number of inputs increases, the adoption of the proper total delay time proves to be the key factor to set up the optimal RNN model.
- In the case that both training and test patterns are corrupted with measurement errors (Fig. 6(*b*)), if a low delay time is adopted (i.e., 0.1 s), the model with the combination of inputs $\boldsymbol{x}(t_d)$ and $\boldsymbol{y}(t_d)$ (i.e., the one-time delayed predictor RNN) allows quite good RNN performance, with an overall prediction error of ~1.4%.

Table 3 One-time delayed RNNs trained and tested on field data

Predictor RNN	Total delay time (s)	Number of training patterns	RNN inputs	Reset time step (s)
SS	0.24 0.0067	1017 1017,2034	Variable	-
MS	0.24 0.0067	509,1017,2034	All and $\boldsymbol{x}(t_d)+\boldsymbol{y}(t_d)$	1,30

A final remark has to be made about the model with $\boldsymbol{x}(t)$ (exogenous inputs evaluated at the current time step) as the only input. In fact, this model often proves to be preferable, both for uncorrupted (Fig. 6(*a*)) and corrupted (Fig. 6(*b*)) data. Such behavior can be attributed to the particular problem considered, since the deviation of system dynamics from its stationary behavior is small: in fact, as partially anticipated, (i) the model only captures dynamic effects deriving from mass storage and thermal exchange phenomena and, above all, (ii) the physical system is characterized by small size volumes.

RNN Application to Field Data

The analyses performed in the previous paragraphs and in [23] on simulated data allow the selection of the parameters to be tuned in order to set up the best RNN model. In this section, RNN capabilities are tested against measured values obtained from the compressor under investigation: the difference between the analyses developed through this paper and through [23] is that in [23] the RNN models were trained on simulated data, while here the RNN models are trained directly on field operating data.

All the RNN models developed in this section, which are summarized in Table 3, are characterized by one-time delayed configuration (with two different values of the total delay time), by 15 neurons in the hidden layer and by a multioutput architecture.

Influence of RNN Inputs. Three different one-time delayed RNNs are considered: (i) total delay time equal to 0.24 s and n_{patt} equal to 1017, (ii) total delay time equal to 0.0067 s and n_{patt} equal to 1017, and (iii) total delay time equal to 0.0067 s and n_{patt} equal to 2034. It should be remembered that the training patterns have been derived by uniformly sampling the available patterns

Downloaded From: http://turbomachinery.asmedigitalcollection.asme.org/ on 08/28/2018 Terms of Use: http://www.asme.org/about-asme/terms-of-use

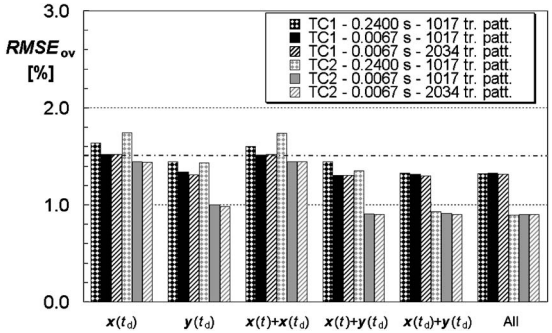

Fig. 7 Influence of RNN inputs for one-time delayed RNNs for TC1 and TC2 curves; the dashed-dotted line stands for the $RMSE_{ov}$ value for the RNN with the only input $x(t)$

from both TC1 and TC2 curves.

On these three RNN models, the analysis of the influence of RNN inputs on RNN performance is performed, as previously conducted on simulated data, by testing the developed RNNs on all the patterns included in the two test cases. In any case, it has been verified that, if the patterns used for training were not used for testing, the results are in practice the same; this can be explained by considering that the considered training patterns represent a small fraction of all the 36,600 available patterns (2.8% or 5.6% when 1017 or 2034 patterns are used for RNN training, respectively).

The results of the analysis are reported in Fig. 7, with the above-introduced notation ($\boldsymbol{x}(t_d)$, vector of exogenous inputs evaluated at antecedent time steps; $\boldsymbol{y}(t_d)$, vector of delayed regressed outputs; "All," solution with all the possible inputs). The dashed-dotted line in Fig. 7 indicates the overall RMSE in the case where the vector of exogenous inputs evaluated at the current time step $\boldsymbol{x}(t)$ is the only RNN input; such error is equal to 1.5% for both TC1 and TC2 curves.

The following comments can be made about the analysis of the influence of inputs conducted on field data:

- It is once again confirmed that, independent of the considered combination of inputs, the lowest RMSE overall values can be obtained with an as low as possible total delay time (0.0067 s) and with an as high as possible number of training patterns (2034). In particular, the reduction of the total delay time seems to be much more effective than the increase of the number of training patterns in reducing the overall error.
- Differently from the case of corrupted simulated data in Fig. 6(*b*), the model with the vector of exogenous inputs evaluated at the current time step $\boldsymbol{x}(t)$ as the only input is almost never preferable. The behavior in the case of field data can be attributed to the fact that, though the dynamic content of real system data is probably still poor (due to small size volumes), some "inertial" effects exist, and thus, the delayed values of exogenous inputs and of regressed outputs have to be both adopted as RNN inputs.
- The combination of inputs $\boldsymbol{x}(t_d)$ and $\boldsymbol{y}(t_d)$, which does not take into account the vector of exogenous inputs evaluated at the current time, allows very good RNN performance, comparable to that of RNN models with "All" inputs. In particular, for the best predictor RNN model (delay time of 0.0067 s and n_{patt} equal to 2034), which uses both $\boldsymbol{x}(t_d)$ and $\boldsymbol{y}(t_d)$ as inputs, the overall prediction error on each single calculation is 1.3% and 0.9% for TC1 and TC2 curve, respectively.

This latter consideration is very important because, by comparing directly the results obtained through the physics-based dynamic model developed in [21] and through the optimal RNN model trained on simulated data developed in [23], it is possible to ascertain that the optimized single-step predictor (total delay time equal to 0.0067 s and n_{patt} equal to 2034), trained on field data, allows better performance than both models. In fact, the detailed results reported in Table 4 for all the quantities show that, independent of the considered quantity and of the maneuver, the RMSE values obtainable through the best SS predictor RNN developed in this paper are the lowest (except p_2 for TC1 curve). The main reason for this result can be attributed to RNN training data, which, in the current paper, are taken experimentally on the compressor under investigation, whereas in [23] they were generated by means of the physics-based model. Finally, it can be noted that RMSE values are all lower (with the only exception of M_1 for TC1 curve) than the percentage uncertainty values reported in Table 1; thus, the SS predictor may be considered validated.

Table 4 RMSE values for the physics-based model for an RNN trained on simulated data and for a predictor RNN trained on field data

	RMSE [%]				
	p_1	p_2	M_1	T_2	Overall
TC1 curve					
Physics-based model [21]	0.23	0.79	3.96	0.47	2.04
RNN trained on simulated data [23]	0.74	0.34	5.93	0.22	3.00
SS predictor RNN trained on field data	0.09	0.51	2.54	0.20	1.30
TC2 curve					
Physics-based model [21]	0.21	0.95	1.69	0.60	1.02
RNN trained on simulated data [23]	1.07	1.65	1.98	0.66	1.43
SS predictor RNN trained on field data	0.09	0.60	1.69	0.19	0.90

Multi-Step-Ahead Predictor RNN. The setup of an MS predictor has been performed by developing an on-purpose simulation program in MATLAB environment. Once the chosen RNN, previously trained off line and acting as an SS predictor, is selected, the program allows the choice of (i) the maneuver to be simulated, (ii) the total time interval to be covered, (iii) data sampling time, and (iv) the time step from which the simulation should start. At each time step, the program estimates the whole vector of RNN inputs by arranging together the vector of exogenous inputs evaluated at antecedent time steps (measured) and the vector of delayed regressed outputs (estimated through the RNN model itself). Thus, the program allows estimation of the outputs for the desired time horizon.

Six different multi-step-ahead predictor RNNs were considered, with a different number of training patterns (509, 1017, and 2034) and with different values of the total delay time (0.24 s and 0.0067 s). With regard to the delay time, it has to be stressed that the available data have been sampled so that 0.24 s and 0.0067 s do not only represent RNN total delay time, but also data sampling time (equal to data "discretization"), i.e., $t_d = \Delta t_s$. This means that, for an assigned time horizon to be simulated, the number of calculations performed by the RNN obviously varies.

The results are presented in Fig. 8 only for the most interesting predictor, i.e., the one with the combination of inputs $\boldsymbol{x}(t_d)$ and $\boldsymbol{y}(t_d)$, while the results for the RNN model with "All" inputs are also reported for comparison purposes only. It should be remembered that the delayed regressed values of outputs $\boldsymbol{y}(t_d)$ used as inputs are estimated through the RNN itself at antecedent time steps.

Downloaded From: http://turbomachinery.asmedigitalcollection.asme.org/ on 08/28/2018 Terms of Use: http://www.asme.org/about-asme/terms-of-use

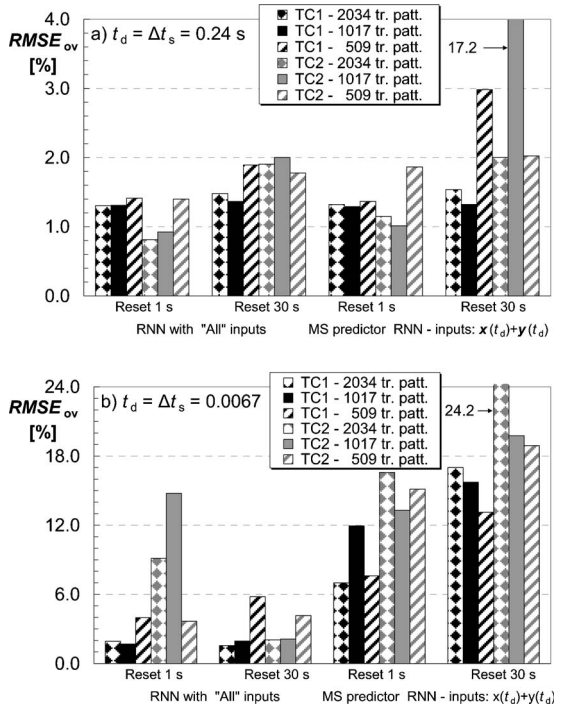

Fig. 8 Overall prediction error for an MS predictor RNN for TC1 and TC2 curve as a function of the reset time step (1 s and 30 s), of the number of training patterns (509, 1017, and 2034) and of data sampling time: (*a*) 0.24 s and (*b*) 0.0067 s

The predictor capabilities are tested over two different time horizons: 1 s and 30 s. This means that the delayed values of outputs $\boldsymbol{y}(t_d)$, which, for an MS predictor RNN are estimated through the RNN itself, are replaced with actual values (measured data) in correspondence with the selected reset time step (1 s or 30 s) to avoid the propagation of the predictor error as the simulation proceeds. As a function of the reset time step, the number of RNN calculations between two subsequent resets varies, as reported in Table 5. However, the calculation of the RMSE_{ov} is performed over the entire curve (TC1 or TC2).

From Fig. 8, it can be observed that the results are approximately independent of the considered test case (TC1 or TC2). As regards the choice of the proper data sampling time and reset time step, it can be observed that:

- for a total delay time of 0.24 s (Fig. 8(*a*)), the performance of the MS predictor is almost comparable to that of the RNN model with "All" inputs, with the exception of two cases (RNN trained on 509 patterns, simulation of TC1 curve and reset after 30 s; RNN trained on 1017 patterns, simulation of TC2 curve and reset after 30 s).
- when a high test data sampling time is chosen (0.24 s, as in Fig. 8(*a*)), prediction errors made by the predictor RNN are acceptable. In fact, if the worst case is excluded (RNN trained on 1017 patterns, simulation of TC2 curve and reset after 30 s), they are in the range of 1.0–1.9% or 1.3–3.0% over a time horizon of 1 s or 30 s, respectively, as a function of the adopted number of training patterns. It is worth noting that this result has been obtained in spite of the fact that the considered RNN is not optimized, since it is characterized by a relatively high total delay time (i.e., 0.24 s), assumed to be equal to the sampling time only for convenience.
- on the other hand, when a low test data sampling time is chosen (0.0067 s), the prediction error is clearly not acceptable, and thus the prediction accuracy can be very poor (Fig. 8(*b*)). In fact, even the comparison of RNN performance for a similar number of calculations performed by the MS predictor (125 for the RNN with total delay time of 0.24 s in Fig. 8(*a*) and 150 for the RNN with total delay time of 0.0067 s in Fig. 8(*b*)) shows that the key factor is represented by the sampling time, which has to be high so that a significant time ahead can be predicted with acceptable errors;
- the comparison of MS predictor performance for the same number of patterns (i.e., 2034) and the same reset time step (i.e., 1 s) also confirms that errors for the MS predictor with total delay time of 0.24 s are acceptable (1.5% or 2.0% for TC1 or TC2 curve, respectively), while errors for the MS predictor with total delay time of 0.0067 s are very high (7.0% or 16.6% for TC1 or TC2 curve, respectively).

Table 5 Number of calculations performed by MS predictor RNNs trained on field data for two values of the reset time step

	Number of RNN calculations	
Reset time step (s)	1	30
$t_d=\Delta t_s=0.0067$ s	150	4500
$t_d=\Delta t_s=0.24$ s	4	125

Discussion on the Capabilities of the Multi-Step-Ahead Predictor RNN. The evaluation of the best single-step-ahead predictor RNN can be done according to the results found in [23] and in the previous section of this paper. In particular, a one-time delayed RNN (i) with as-low-as-possible total delay time $t_d=n_d\Delta t_s=k_{tr}(\Delta t_s)_{tr}$ and (ii) trained with an as-high-as-possible number of patterns $(n_{patt})_{tr}=(\Delta t_{tot})_{tr}/(\Delta t_s)_{tr}$ proved to be the best solution.

Thus, if t_d is kept low in order to achieve a good RNN performance on the single calculation, the number of calculations $(n_{patt})_{test}$ must be high, once the total time interval to be simulated is fixed, in accordance with Eq. (4). This finally leads to ever increasing errors on the delayed values of outputs, since the number of RNN calculations increases.

Another possibility, implicitly suggested by Eq. (4), is the increase of the number of resets. This operation presents two weak points because the use of data from outside the RNN (i) limits the use of the RNN model as a predictor and, mainly, (ii) cannot be applied if the system under investigation is fast varying over time. In fact, in this case, the acquisition system may not have enough time to feed the RNN model. Moreover, the solution of adopting high test data sampling time (as in Fig. 8(*a*)) can be recommended only in the case that the dynamics of the system is slow enough that fast-acting events are not expected to happen.

Thus, the final result of the investigation conducted in this paper is that the best solution in setting up the RNN model can be found by properly weighting the following factors: (i) number of calculations performed by the RNN between two subsequent resets, (ii) data sampling time, (iii) RNN delay time, (iv) number of training patterns, (v) desired RNN model discretization over time, and (vi) system dynamics. Items i–iv (arranged in order of importance) only deal with the set up of the RNN, items v and vi deal with the particular problem under consideration. In fact, if the system is characterized by high "inertia," sampling frequencies can be low since its variations are slow; on the other side, if the system is characterized by low inertia, sampling frequencies must be high if the process (and its fast variations) is to be captured.

In any case, as a general conclusion, the reduction of the number of RNN calculations seems to be the key factor for improving the capabilities of multi-step-ahead predictor RNNs, as also evidenced in [29].

Conclusions

In the paper, recurrent feed-forward neural networks (RNNs) with one feedback loop were set up and tested in order to evaluate their capability in reproducing compressor behavior under un-

Downloaded From: http://turbomachinery.asmedigitalcollection.asme.org/ on 08/28/2018 Terms of Use: http://www.asme.org/about-asme/terms-of-use

steady conditions. The data used for training and testing the RNNs have been obtained both by means of a nonlinear physics-based model for compressor dynamic simulation (simulated data) and measured on a multi-stage axial-centrifugal small-size compressor (experimental data).

The analysis on simulated data led to the conclusion that errors decrease as the number of training patterns (which depends on the chosen sampling time once the total time interval considered for training data is fixed) increases, though good performance can be obtained by using at least 500 (or even 250 for the best networks) training patterns. Moreover, the evaluation of the influence of each input on RNN response showed that, independent of the considered combination of inputs, the lowest errors can be obtained by adopting a one-time delayed RNN with an as low as possible total delay time. Thus, the combined action of an increase in the number of training patterns and of a decrease in the RNN total delay time is beneficial to improve RNN performance, mainly in the case that the exogenous inputs evaluated at the current time step are not supplied, since in this manner the increase in errors can be limited.

The analysis of the influence of each input on RNN response, conducted for RNN models trained directly on field operating data, confirmed that, independent of the considered combination of inputs, the lowest overall errors can be obtained with an as-low-as-possible total delay time (in this paper, 0.0067 s) and by increasing the number of training patterns (in this paper, 2034). In particular, the RNN model, which does not include the vector of exogenous inputs evaluated at the current time step among the inputs, allowed very good RNN performance, comparable to that of RNN models with all inputs (overall error for each single calculation equal to 1.3% and 0.9% for the two test cases considered). Since errors were all lower (with only one exception) than the percentage measurement uncertainty values, the RNN model may be considered validated. Moreover, the optimized single-step predictor allowed better performance than both the physics-based dynamic model and the optimal RNN model trained on simulated data.

The analysis of multi-step-ahead predictor capabilities showed that prediction errors were acceptable when a high test data sampling time was chosen (in this paper, 0.24 s). In fact, with the exception of only one case, they were in the range of 1.0–1.9% or 1.3–3.0% for the two considered values of the reset time step (1 s and 30 s, respectively), as a function of the adopted number of training patterns. This result was obtained in spite of the fact that the considered RNN was not optimized, since it was characterized by a relatively high delay time (0.24 s), assumed equal to the sampling time only for convenience. On the other hand, when a low test data sampling time is chosen (in this paper, 0.0067 s), the prediction error was clearly not acceptable.

Thus, in order to set up the optimal multi-step-ahead predictor RNN model, the best solution can be found by properly setting up the best single-step-ahead predictor (through the choice of the proper number of calculations performed by the RNN between two subsequent resets, data sampling time, RNN delay time and number of training patterns) and by taking into account the particular problem under consideration (i.e., desired RNN model discretization over time and system dynamic content). In any case, as a general conclusion, the reduction of the number of RNN calculations seems to be the key factor for improving the capabilities of multi-step-ahead predictor RNNs over a significant time horizon.

Future developments of the present study will be aimed at evaluating the capabilities of such predictors for dynamic process systems characterized (i) by more considerable inertial effects (e.g., higher-size volumes and effect of spool dynamics) and (ii) by more complex configurations, which, for instance, take into consideration the whole energy system, with the interaction of several machines and pipelines.

Acknowledgment

The work was carried out with the support of the M.i.U.R. (Italian Ministry of Education, University and Research). The author gratefully acknowledges Prof. Roberto Bettocchi, Prof. Pier Ruggero Spina, Michele Pinelli, Ph.D., and Dr. Mirko Morini for the suggestions.

Nomenclature

d = vector dimension
F = RNN transfer function
k = multiplicative factor for data sampling time
M = mass flow rate
MS = multi-step-ahead
MSE = mean square error
n_{d} = number of time delay units
n_{i} = number of inputs
n_{o} = number of outputs
n_{patt} = number of patterns
n_{reset} = number of resets
n_{TR} = number of transients used for RNN testing
N_{C} = compressor rotational speed
p = pressure
RMSE = root-mean-square error
SS = single-step-ahead
t = time, expected target output
t_{d} = total delay time ($=n_{\mathrm{d}}\Delta t$)
T = temperature
TC = test case curve (field data)
TR = transient curve (simulated data)
$\boldsymbol{x}$ = vector of exogenous inputs
$\boldsymbol{y}$ = vector of delayed outputs
y = computed output
Δt = time delay unit
Δt_{s} = sampling time
Δt_{tot} = total time interval

Subscripts and Superscripts

0,1,2,3 = compressor physics-based model sections
m = mean on all transients used for testing
ov = overall
tr = training
test = test
$\boldsymbol{x}$ = input vector $\boldsymbol{x}$
$\boldsymbol{y}$ = output vector $\boldsymbol{y}$

References

[1] Parlos, A. G., Chong, K. T., and Atiya, A. F., 1994, "Application of the Recurrent Multilayer Perceptron in Modeling Complex Process Dynamics," IEEE Trans. Neural Netw., **5**(2), pp. 255–266.
[2] Torella, G., and Lombardo, G., 1996, "Neural Networks for the Diagnostics of Gas Turbine Engines," ASME Paper No. 96-TA-39.
[3] Kanelopoulos, K., Stamatis, A., and Mathioudakis, K., 1997, "Incorporating Neural Networks Into Gas Turbine Performance Diagnostics," ASME Paper No. 97-GT-35.
[4] Bettocchi, R., Spina, P. R., and Torella, G., 2002, "Gas Turbine Health Indices Determination by Using Neural Networks," ASME Paper No. GT-2002-30276.
[5] Arriagada, J., Genrup, M., Loberg, A., and Assadi, M., 2003, "Fault Diagnosis System for an Industrial Gas Turbine by Means of Neural Networks," *Proc. of International Gas Turbine Congress 2003 (IGTC'03)*, Tokyo, Nov. 2–7, GTSJ, Tokyo, Paper No. IGTC2003Tokyo TS-001.
[6] Bettocchi, R., Pinelli, M., Spina, P. R., and Venturini, M., 2005, "Artificial Intelligence for the Diagnostics of Gas Turbines. Part I: Neural Network Approach," ASME Paper No. GT2005-68026.
[7] Volponi, A. J., DePold, H. R., Ganguli, R., and Daguang, C., 2000, "The Use of Kalman Filter and Neural Networks Methodologies in Gas Turbine Performance Diagnostics: A Comparative Study," ASME Paper No. 2000-GT-0547.
[8] DePold, H. R., and Gass, F. D., 1999, "The Application of Expert Systems and Neural Networks to Gas Turbine Prognostics and Diagnostics," ASME J. Eng. Gas Turbines Power, **121**, pp. 607–612.
[9] Sampath, S., and Singh, R., 2004, "An Integrated Fault Diagnostics Model Using Genetic Algorithm and Neural Networks," ASME Paper No. GT-2004-53914.
[10] Bettocchi, R., Pinelli, M., Spina, P. R., and Venturini, M., 2005, "Artificial

Downloaded From: http://turbomachinery.asmedigitalcollection.asme.org/ on 08/28/2018 Terms of Use: http://www.asme.org/about-asme/terms-of-use

Intelligence for the Diagnostics of Gas Turbines. Part II: Neuro-Fuzzy Approach," ASME Paper No. GT2005-68027.
[11] Simani, S., Fantuzzi, C., and Spina, P. R., 1998, "Application of a Neural Network in Gas Turbine Control Sensor Fault Detection," *Proc. of 1998 IEEE International Conference on Control Applications*, Trieste, Italy, IEEE, New York, pp. 182–186.
[12] Botros, K., K., Kibria, G., and Glover, A., 2000, "A Demonstration of Artificial Neural Networks Based Data Mining for Gas Turbine Driven Compressor Stations," ASME Paper No. 2000-GT-0351.
[13] Fink, D. A., Cumpsty, N. A., and Greitzer, E. M., 1992, "Surge Dynamics in a Free-Spool Centrifugal Compressor System," ASME J. Turbomach., **114**, pp. 321–332.
[14] Hale, A. A., and Davis, M. W., 1992, "DYNamic Turbine Engine Compressor Code DYNTECC—Theory and Capabilities," AIAA Paper No. AIAA-92-3190.
[15] Blotemberg, W., 1993, "A Model for the Dynamic Simulation of a Two-Shaft Industrial Gas Turbine With Dry Low No_x Combustor," ASME Paper No. 93-GT-355.
[16] Bettocchi, R., Spina, P. R., and Fabbri, F., 1996, "Dynamic Modeling of Single-Shaft Industrial Gas Turbine," ASME Paper No. 96-GT-332.
[17] Bianchi, M., Peretto, A., and Spina, P. R., 1998, "Modular Dynamic Model of Multi-Shaft Gas Turbine and Validation Test," *Proc. of The Winter Annual Meeting of ASME*, ASME, New York, Vol. AES-38, pp. 73–81.
[18] Camporeale, S. M., Fortunato, B., and Mastrovito, M., 2002, "A High-Fidelity Real-Time Simulation Code of Gas Turbine Dynamics for Control Applications," ASME Paper No. GT-2002-30039.
[19] Theotokatos, G., and Kyrtatos, N. P., 2003, "Investigation of a Large High-Speed Diesel Engine Transient Behaviour Including Compressor Surging and Emergency Shutdown," ASME J. Eng. Gas Turbines Power, **125**, pp. 580–589.
[20] Tveit, G. B., Bjorge, T., and Bakken, L. E., 2005, "Impact of Compressor Protection System on Rundown Characteristics," ASME Paper No. GT2005-68436.
[21] Venturini, M., 2005, "Development and Experimental Validation of a Compressor Dynamic Model," ASME J. Turbomach., **127**(3), pp. 599–608.
[22] Morini, M., Pinelli, M., and Venturini, M., 2006, "Development of a One-Dimensional Modular Dynamic Model For The Simulation of Surge in Compression Systems," ASME Paper No. GT2006-90134.
[23] Venturini, M., 2006, "Simulation of Compressor Transient Behavior Through Recurrent Neural Network Models," ASME J. Turbomach., **128**(4), pp. 1–11.
[24] Bozzi, L., Crosa, G., and Trucco, A., 2003, "Simplified Simulation Block Diagram of Twin-Shaft Gas Turbines," ASME Paper No. GT-2003-38679.
[25] Ohanian, S., and Kurz, R., 2001, "Series or Parallel Arrangement in a Two-Unit Compressor Station," ASME J. Eng. Gas Turbines Power, **124**, pp. 936–941.
[26] Bettocchi, R., Pinelli, M., Spina, P. R., Venturini, M., and Zanetta, G. A., 2006, "Assessment of the Robustness of Gas Turbine Diagnostics Tools Based on Neural Networks," ASME Paper No. GT2006-90118.
[27] Jiang, D., and Wang, J., 2000, "On-Line Learning of Dynamical Systems in the Presence of Model Mismatch and Disturbances," IEEE Trans. Neural Netw., **11**(6), pp. 1272–1283.
[28] Plett, G. L., 2003, "Adaptive Inverse Control of Linear and Nonlinear Systems Using Dynamic Neural Networks," IEEE Trans. Neural Netw., **14**(2), pp. 360–376.
[29] Parlos, A. G., Rais, O. T., and Atiya, A. F., 2000, "Multi-Step-Ahead Prediction in Complex Systems Using Dynamic Recurrent Neural Networks," Neural Networks, **13**(7), pp. 765–786.
[30] Desideri, U., Fantozzi, F., Bidini, G., and Mathieu, P., 1997, "Use of Artificial Neural Networks for the Simulation of Combined Cycles Transients," ASME Paper No. 97-GT-442.
[31] Chiras, N., Evans, C., and Rees, D., 2002, "Nonlinear Gas Turbine Modeling Using Feedforward Neural Networks," ASME Paper No. GT-2002-30035.
[32] Arsie, I., Pianese, C., and Sorrentino, M., 2002, "Recurrent Neural Network Based Air-Fuel Ratio Observer for SI Internal Combustion Engines," *Proc. of ESDA 2002*, Istanbul, ASME, New York, Paper No. ESDA2002/APM038 ACC008.
[33] Ogaji, S. O. T., Li, Y. G., Sampath, S., and Singh, R., 2003, "Gas Path Fault Diagnosis of a Turbofan Engine From Transient Data Using Artificial Neural Networks," ASME Paper No. GT2003-38423.
[34] Menon, S., Uluyol, O., and Gupta, D., 2004, "Turbine Engine Modeling Using Temporal Neural Networks for Incipient Fault Detection and Diagnosis," ASME Paper No. GT2004-53649.
[35] Haykin, S., 1999, *Neural Networks—A Comprehensive Foundation*, 2nd ed., Prentice-Hall, Englewood Cliffs, NJ.
[36] Bettocchi, R., Pinelli, M., and Spina, P. R., 2005, "A MultiStage Compressor Test Facility: Uncertainty Analysis and Preliminary Test Results," ASME J. Eng. Gas Turbines Power, **127**(1), pp. 170–177.
[37] Traverso, A., Scarpellini, R., and Massardo, A., 2005, "Experimental Results and Transient Model of an Externally Fired Micro Gas Turbine Technology," ASME Paper No. GT2005-68100.
[38] Stiller, C., Thorud, B., and Bolland, O., 2005, "Safe Dynamic Operation of a Simple SOFC/GT Hybrid System," ASME Paper No. GT2005-68481.
[39] Cybenko, G., 1989, "Approximation by Superimposition of a Sigmoidal Function," Math. Control, Signals, Syst., **2**, pp. 303–314.
[40] Atiya, A., F., and Parlos, A., G., 2000, "New Results on Recurrent Network Training: Unifying the Algorithm and Accelerating Convergence," IEEE Trans. Neural Netw., **11**(3), pp. 697–703.

Downloaded From: http://turbomachinery.asmedigitalcollection.asme.org/ on 08/28/2018 Terms of Use: http://www.asme.org/about-asme/terms-of-use