14th International Conference on
AEROSPACE SCIENCES & AVIATION TECHNOLOGY,
ASAT - 14 – May 24 - 26, 2011, Email: asat@mtc.edu.eg
Military Technical College, Kobry Elkobbah, Cairo, Egypt
Tel: +(202) 24025292 –24036138, Fax: +(202) 22621908

Testing and Model Identification of a Turbojet Engine Using Neural Networks

I. M. Atia* and A. M. Bayoumy †

Abstract: Artificial Neural Networks (NN) are a well-known tool among artificial intelligence techniques that are able to reproduce arbitrary relationships existing between input and output variables of even highly non-linear systems.

In this paper, a small turbojet engine SR-30 is tested on a minilab test-rig. Then linear ARX (AutoRegressive with eXternal input) structure and nonlinear neural network representations are used for modeling the dynamics of this small turbojet engine. This modeling is based on real engine data obtained from testing of the SR-30 engine.

In order to build a feed forward NN model, one could identify the nature and characteristics of its dynamics and the order of the system to be modeled by using conventional linear system identification. This step is used to obtain a linear ARX model. Using the input/output relationship of this model, a neural model is trained for the SR-30 turbojet engine that represents the nonlinearity of the engine throughout its full operating range. Validation of this neural model is performed using another set of the experimental data.

The work shows that neural model could capture system nonlinearity and represent the real engine dynamics better than the linear ARX model.

Keywords: Small turbojet engines, artificial intelligence, neural networks, system identification, modeling and simulation, engine testing.

Nomenclature

A/D	Analog to digital
ARX	AutoRegressive with eXternal input
b	A bias of a neuron unit
G_f	Fuel flow rate (kg/s)
GSP	Gas turbine simulation program
I/O	Input and output
logsig	The sigmoid function
n	Engine revolution speed,(rpm)
NN	Neural networks

* Egyptian Armed Forces, Egypt, Hema1080@yahoo.com
† Egyptian Armed Forces, Egypt, ambayoumy@gmail.com

P	Pressure
p	External input
PID	Proportional, integral and derivative controller
PR	Pressure ratio
R	Number of inputs
tansig	A hyperbolic tangent function
TIT	Turbine inlet temperature
w	Weight of neural networks connection

Introduction

Gas turbines have various application areas as prime movers in planes, in power plants for electricity generation and in naval vessels for propulsion [1, 2, and 3]. Gas turbine engine design and manufacturing process has its origins back to mid 1940's. Because of their broad application fields, engineers tried to design more efficient, powerful, economic and reliable gas turbines. Nowadays, almost all aircrafts are powered by gas turbines, which have different configurations such as turbo-fan, turbo-prop, turbo-shaft, and turbo-jet.

Increasing demands for gas turbines in many fields have caused different customer needs. Especially more efficient and reliable engines are the most important desirable features in industry. Although gas turbines have some disadvantages such as high fuel consumption rate, high technology production techniques, high technology materials usage, engineers are still working to increase their efficiency. Gas turbines are very important power generator alternatives for today's airplanes. For example, jet fighters utilize small gas turbines as a main engine starter and auxiliary power sources. Small gas turbines are generally composed of a single stage radial compressor, a diffuser, a combustion chamber, a single stage axial turbine and a nozzle. These different structures are combined by complicated aero-thermodynamic rules.

Mathematical modeling, transient behavior analysis of a small gas turbine engine is the focus of this study. Transient intervals are the operating regions where the most critical conditions arise in gas turbine operations [4, 6, and 7].

For this purpose, dynamic modeling of gas turbine engine should be constructed and its responses to different operating conditions with different fuel inputs should be analyzed. Various aero-thermal and differential equations are used for transient analyses [2, 3, 4, 5, and 6]. The resulting differential equations are non-linear in nature and can be solved simultaneously with aero-thermal equations.

In [8] an extended least-squares algorithm with optimal smoothing was used by Norton, to identify time-varying transfer function models to represent large transient and non equilibrium effects and provide a more detail insight into the slow thermal dynamics of the engine. In order to identify a model capable of representing the engine at all operating points, Rodriguez [9] used a multi objective genetic programming approach on the same data and allocated weights to various objectives, to assess their significance in the structure selection of Nonlinear AutoRegressive Moving Average with eXogenous inputs (NARMAX) models of the engine. Chiras et al. [10, 11] used nonparametric data analysis in both time- and frequency-domains and an orthogonal estimation algorithm to estimate NARMAX models of the engine. A simple NARX model was identified which was able to represent both the small and large signal dynamics of the engine.

Neophytos Chiras et al. [12] developed a feed forward neural network model of a Spey gas turbine engine. This model was used to model the fuel flow to shaft speed relationship. The performance of the estimated model is validated against a range of small and large signal engine tests. It is shown that the performance of the estimated models is superior to that of the estimated linear models.

In fact, the number of operations required in neural networks model is minimal with respect to the analytical model, whose non-linear set of thermodynamic equations usually requires an iterative algorithm to be solved. Another advantage of a NN is its intrinsic ability of adaptation to a given plant. While the analytical model has to be “tuned” to have its output represent accurately the behavior of the plant, a NN already adjusts its output implicitly during the training phase. However, the NN approach to the simulation of a real plant can only be considered for a stand-alone model if a large number of data related to the desired input and output variables is available.

A. Watanabe et al. [13] worked on PID and fuzzy logic algorithm in order to control SR-30 turbojet engine. They obtained transfer function of the SR-30 by using frequency response method. They tested and simulated both closed loop controller PID and fuzzy logic controller. They developed their model with MATLAB environment and tested it by NI LabVIEW.

In their study, R. Andoga et al. [14] discussed digital electronic control of a small turbojet engine. They stated that main purpose of control of gas turbine was increasing its safety and efficiency. Their engine was controlled by PIC 16F84A microcontroller, which was manipulating the fuel flow valve.

D. May et al. [15] make a simulation of the SR-30 engine using GSP software provided a model. The model was used to determine the healthy operation of the engine. The integration of a developed algorithm has enabled the system to be used for aerodynamic component monitoring, as well as, mechanical systems monitoring.

O. Léonard et al. [16] make a modification to the SR-30 engine and the test bench. These modifications were led by a triple objective: the improvement and the enrichment of the measurement chain, the widening of the engine’s operational domain. Several performance models of the engine were developed to support data validation and engine condition diagnostic. Ecosimpro [17] was selected as the platform for the development of the engine model.

M.Ghoreyshi et al. [18] presents the design procedure and application of a nested neural network for diagnostics of a small jet engine. Such a diagnostics technique is based on the performance analysis while the performance model was developed with TURBOMATCH, the Cranfield University's gas turbine simulation code. To validate this model, an outdoor test was conducted to run the small engine. Areas examined in this paper are performance validation of the engine, neural network design, training data generation, and networks training procedures. The assumptions, measured parameters selection and the results obtained are presented and discussed. The results obtained show the good prospects for the use of NNs for detection of existing faults, isolation of faults and quantification of fault levels.

In this paper, a small turbojet engine SR-30 is tested on a minilab test-rig. Then linear ARX (AutoRegressive with eXternal input) structure and nonlinear neural network representations are used for modeling the dynamics of this small turbojet engine. This modeling is based on real engine data obtained from testing of the SR-30 engine.

In this paper a feed forward neural network is used to model the fuel flow to shaft speed relationship of a SR-30 turbojet engine. The performance of the estimated model is validated against a range of small and large signal engine tests. In order to build a feed forward NN model, one could identify the nature and characteristics of its dynamics and the order of the system to be modeled by using conventional linear system identification. This step is used to obtain a linear ARX model. Using the input/output relationship of this model, a neural model is trained for the SR-30 turbojet engine that represents the nonlinearity of the engine throughout its full operating range. Validation of this neural model is performed using another set of the experimental data.

Neural Network Model

Neural networks are so fashionable that even old types of models known by other names, have been converted to, or reinvented as neural networks. This makes it difficult to find a universal definition of what a neural network is and even impossible to cover all types of neural networks. A simple definition of neural networks can be formulated as in [28]: A system of simple processing elements, neurons that are connected into a network by a set of (synaptic) weights. The function of the network is determined by the structure of the network, the magnitude of the weights and the mode of operation of the processing elements.

The basic neural network element is a neuron shown in Fig. 1. This is a processing element that takes a number of inputs, applies some weights and sums them up with a bias *b*, and feeds the result (E) to an activation function as shown in equation(1). The inputs to the neuron can be external inputs or outputs of proceeding neurons.

$$E = \sum_{i=1}^{R} w_i * p_i + b \tag{1}$$

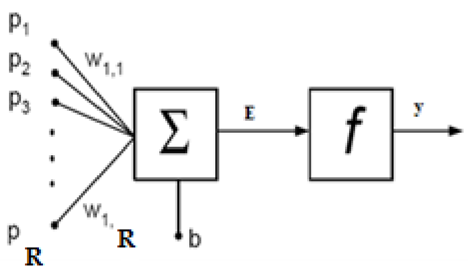


Fig. 1 Neuron: y = f (Wp + b).[29]

The activation function (f) can be any kind of singular valued function, linear (purelin) or nonlinear. The most popular nonlinear activation functions used in system identification are the sigmoid function (logsig) (2) and the hyperbolic tangent function (tansig). These functions are nonlinear thus they define the nature of the particular neural network.

$$\text{logsig}(x) = \frac{1}{1 + e^{(-x)}} \tag{2}$$

The net work consists usually of three layers: input layer, hidden layer and output layer. , each layer consists of number of neurons which are the computational atoms, and which are interconnected with each other using weighted connection [25]. Each neuron sums the incoming signal weight products and a bias and passes the result through a nonlinear activation function as represented by the following equation(3).

$$y(t) = \text{logsig}\left(\sum_{i=1}^{R} w_i p_i(t) + b \right) \quad (3)$$

where y(t) is the output , $p_i(t)$ is multiple inputs, w_i is a weight of connection, b is a bias of a neuron unit, t is the time and R is the number of inputs.
The setup of a NN requires the choice of the number of layers, the number of neurons in each layer, the transfer function of each layer and the training algorithm [19]. Two phases are then required to make the NN become operative. The first one is the training (or learning) phase, in which the NN is taught to match a known set of corresponding input and output values. This allows the NN to "learn" the relationship existing between inputs and outputs.

During the learning process, "learning" is achieved through modification of weights associated with each neural connection made by the training algorithm (also called "learning rule"). The training process aims, in general, at the minimization of the error between predicted and actual values. This phase is the most time consuming and it is critical for the success of the NN as a predictive model. The second phase is called generalization (or testing). Here, the NN is tested on another known set of corresponding input and output values different from the training set and the performance is evaluated [19, 20].

Cybenco [22] proved that a neural network with one hidden layer of sigmoid or hyperbolic tangent units and an output layer of linear units is capable of approximating any continuous function. The network is described by the magnitude of the weights and biases and should be determined by training the network on the estimation data. The estimation of the weights is usually a conventional estimation problem and several algorithms are available for this purpose.

Neural network are built from neurons [21, 23]. Which are the computational atoms, and which are interconnected with each other using weighted connections. The information is propagated from each neuron's output to a number of neuron's inputs. The neurons are arranged in layers, and if there are only connections from neurons to neurons in the next layer, the net work is called a feed forward neural net work. In case where there are feedback connections, the network is called a recurrent neural network

Recurrent neural networks often suffer from instability and long training and recall times. These problems are not present in feed-forward neural network. Considering these advantages of feed-forward neural networks only this type of neural networks is used in this work, the dynamic behavior of the plant is modeled using external feedback lines and delayed values of the neural net work inputs.

Feed forward neural networks are proved to have excellent function approximation capabilities [22-24] thus justifying the enormous amount of research dedicated to the subject in recent decades. Feed forward networks often have one or more hidden layers of sigmoid neurons followed by an output layer of linear neurons. Multiple layers of neurons with nonlinear transfer functions allow the network to learn nonlinear and linear relationships between input and output vectors.

In this paper a feed forward neural network with two hidden layer having a sigmoid function and output layer having a linear function as shown in Fig. 2 and using past inputs and outputs terms as inputs to model the SR-30 turbojet engine

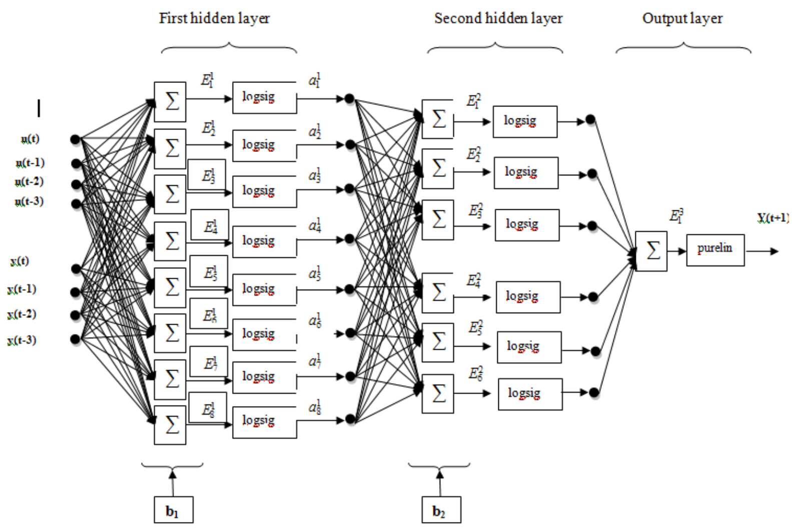


Fig. 2   Two-hidden layer feed forward neural network

In order to build a feed forward NN model, one could identify the nature and characteristics of dynamics and the order of the system to be modeled by using conventional linear system identification. This step needs an experimental data which will be prepared in the next section.

Testing of the SR-30 Turbojet Engine

The NN approach was used to predict the operation of the gas turbine plant SR-30 turbojet engine. The SR-30 (Fig. 3) turbojet has been incorporated into many laboratories worldwide. The SR-30 engine is a turbojet engine with a single-stage radial-flow compressor with a maximum pressure ratio of, PR=3.4, a reverse-flow annular combustion chamber and a single-stage axial-flow turbine, and it operates obeying the Brayton thermodynamic cycle in the same fashion as large turbojet engines. The engine, as produced by Turbine Technologies, includes five pressure transducers, five thermocouples, a load-cell for thrust measurements, a custom motor winding for reading the engine RPM, and a fuel-flow-rate measurement system to monitor/measure the operating parameters of the engine. The engine generates 178 N of thrust at 87,000 rpm while ingesting m = 0.5 kg/s of air. The engine has a length of 27 cm, and the exit exhaust diameter of D_{exit}= 5.715 cm.

Fig. 3 SR-30 Test Rig

Thirteen engine parameters are measured with the stock MiniLab configuration. The basic sensor package includes pressure, temperature, RPM and flow sensors (calibrated) measuring parameters common to Brayton Cycle type analysis. The sensors are routed to a central access panel and interfaced with data acquisition hardware and software from National Instruments. The sensor locations are shown in Fig. 4. The integrated sensor system (Mini-Lab) includes the following probes:

- Compressor inlet static pressure (P_1=P_{01}).
- Compressor stage exit stagnation pressure (P_{02}).
- Combustion chamber pressure (P_3).
- Turbine exit stagnation pressure (P_{04}).
- Thrust nozzle exit stagnation pressure (P_{05}).
- Compressor inlet static temperature (T_1=T_{01}).
- Compressor stage exit stagnation temperature (T_{02})
- Turbine stage inlet stagnation temperature (T_{03})
- Turbine stage exit stagnation temperature (T_{04})
- Thrust nozzle exit stagnation temperature (T_{05}).
- RPM (n) engine rotational speed derived from measuring the output voltage of a generator mounted on the compressor.
- Additionally, the system includes a fuel flow (G_f) sensor and a digital thrust readout measuring real time thrust force based upon a strain gage thrust yoke system.

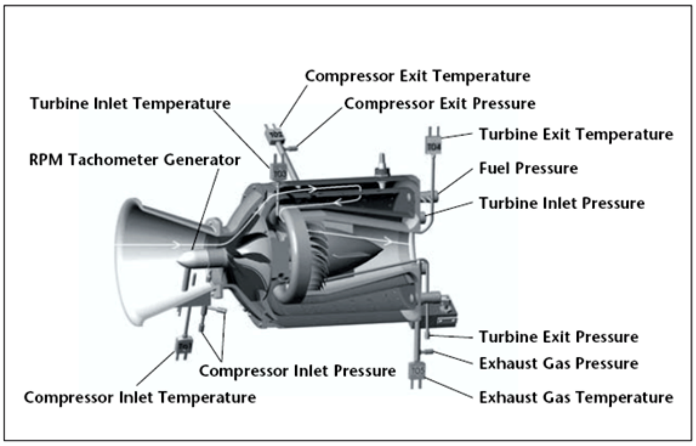


Fig. 4 Sensor location [26]

The current system at the Egyptian Air forces Research Center includes a National Instruments NI PCI-6031E A/D board, which has a 16-bit resolution, 100 k Samples/s sampling rate, 64 analog input, 2 analog output, and 8 digital I/O ports. Signal connections to the A/D board are made using two enclosures, which are attached to the A/D board using cables. While one of these units (NI-SC-2345) houses the thermocouple input modules (NI-SCC-TC02), the millivolt range input modules (NISCC- AI06), a connector block for digital output signals, and the analog output modules, the other unit (NI-CA-1000) houses a connector block (NI-CB- 68LPR) to connect other analog inputs. Each input module includes self-contained signal-conditioning units such as low pass filters and instrumentation amplifiers.

While the thermocouple input modules are used to read the temperatures, the mill volt input range modules are used to read the load-cell output voltage and one of the pressure signals. Digital I/O lines are used to generate signals to turn on and off the relays. Relays (BASCO Company, ELK 924) are used to replace manual switches, which are used to start and to stop the ignition, the fuel flow and to turn on and off the valve for high-pressure air. More data about the engine can be found in [26].

Experiment Procedure

The objective of this experiment is to measure (P_{01}, T_{01}, n, G_f, etc…) at different engine speeds (at steady state).

Follow the “Pre-Start Checklist” and “Starting Procedure” to start the engine [27]. The turbine is inspected to insure proper rotation of the blades and to remove any debris that may get sucked into the SR-30 engine. The fluid levels are checked to insure adequate lubrication and fuel. The lab equipment is set up by connecting an air compressor and a computer to the Mini-Lab. The air compressor is used to initiate rotation of the impeller in order to get the engine up to operating RPM range. The Virtual Bench program is used for data acquisition during a steady RPM of the engine.

Steady state data are collected at different engine speeds ranging from 41,000 to 80,000 rpm. These data are used to build the engine model by using the neural networks.

The following table summarizes the experimental data on the SR-30 turbojet engine.

Table 1 Experiments Summary

No.	Sampling time ms	Duration min	Fuel type	Max TIT °C	Max speed rpm
1	247	3.5	Diesel fuel	627	59,700
2	247	3.7	Diesel fuel	643	70,000
3	436	2.7	Jet A-1	746	76,400
4	436	12	Jet A-1	775	80,000
5	436	10.48	Jet A-1	706	71,200

System Identification

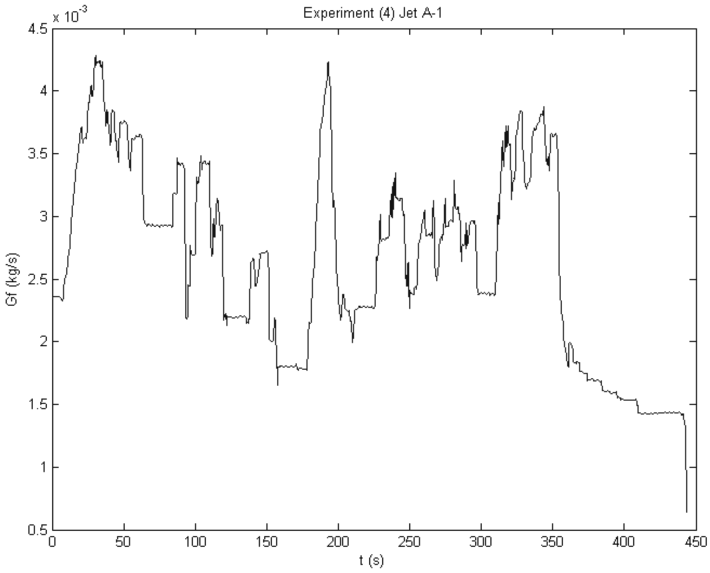
The first and most important step in model building is to identify the nature and characteristics of its dynamics and the order of the system to be modeled. This process is called System Identification. System Identification Toolbox software lets us to estimate linear and nonlinear mathematical models of dynamic systems from measured data.

The resulting model might be used to simulate the output of a system for a given input and analyze the system response, predict future system outputs based on previous inputs and outputs, or for control design.

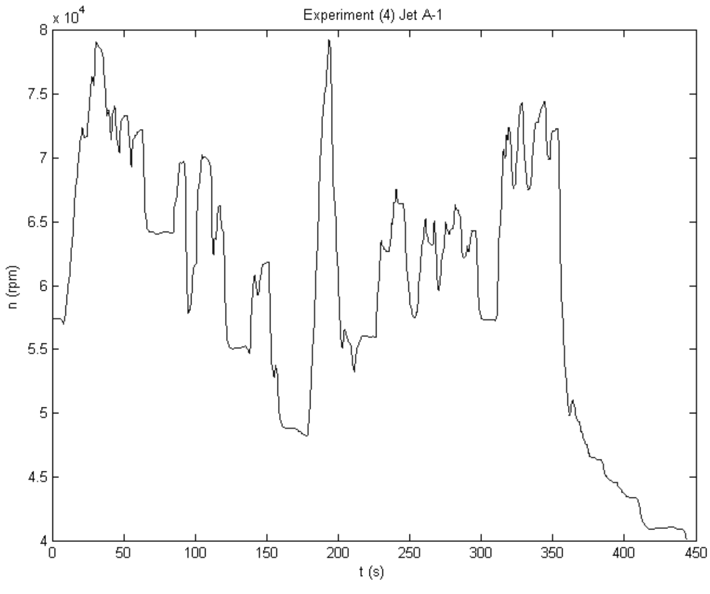
To identify a system, the following steps should be followed:

- Data Collection
- Model Generation
- Minimization of errors

Model generation is a multi-step process, as shown in Fig. 6. It is an iterative technique, and these steps are repeated until we get the final model, a sufficiently accurate representation of the physical system.



(a)



(b)

Fig. 5 Experiment (3) data
(a) fuel flow rate, (b) engine speed

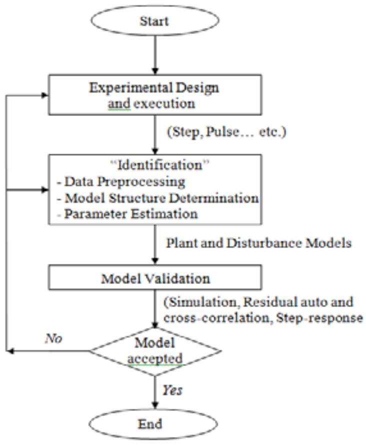


Fig. 6 Flowchart of system identification

The System Identification Toolbox product in Matlab program [29] supports all of these stages except data acquisition. This toolbox provides some support for experimental design by enabling you to generate input signals with different properties. This toolbox lets you estimate different model structures quickly; you should try as many different structures as possible to see which one produces the best results. You can also model data to validate and refine your experimental design.

The identification problem in the time-domain for either linear or nonlinear modeling is to deduce relationships between past input-output data and future outputs. If a finite number of past inputs u (t) and outputs y (t) are collected into the vector ϕ (t)(4).

$$\phi(t) = \left[y(y-1).........y(t-n_y)u(t-1)....u(t-n_{u)} \right]^T \tag{4}$$

Then the problem is to understand the relationship f between the next output y (t) and ϕ (t)
To obtain this understanding a set of observed data is required which consists of the input u(t) and output y (t), from which the vectoϕ (t) can be built. The function (f_{id}) can be linear or nonlinear function. In the case for which (f_{id}) is a linear function several model structures are well documented such as ARX (Autoregressive with eXogenous inputs) model. This model is parametric and has the following structure

$$y(t) + a_1 y(t-1) + ...a_{n_a} y(t-n_a) = b_1 u(t-n_k) + ...b_{n_b} u(t-n_b+1) + e(t) \tag{5}$$

where y (t) represents the output at time t, u (t) represents the input at time t, n_a is the number of poles, n_b is the number of b parameters (equal to the number of zeros plus 1), n_k is the number of samples before the input affects output of the system (called the delay or dead time of the model), and e(t) is the white-noise disturbance.

The System Identification Toolbox product estimates the parameters $a_1.....a_n$ and $b_1.....b_n$ using the input and output data from the estimation data set. In arxqs, n_a=n_b=4, and n_k is estimated from the step response.

In our work, we use this tool to identify the system order which give the best performance. The best system is the ARX model with order [4 4 0] , n_a =4 and n_b=4, is represented

$$A(q^{-1})y(t) = B(q^{-1})u(t) + e(t)$$
$$A(q^{-1}) = 1 - 1.2q^{-1} + 0.1724q^{-2} - 0.307q^{-3} + 0.3371q^{-4}$$
$$B(q^{-1}) = 1746q^{-1} + 946.8q^{-2} - 1360q^{-3} - 1265q^{-4}$$

So the system input will be [y(t),y(t-1),y(t-2),y(t-3),y(t-4),u(t),u(t-1)u(t-2),u(t-3)]

The output from the ARX engine model was illustrated in the following Fig. 7 which shows the comparison between the ARX model with the measured data from experiment (4).The average value of the mean square error between the model output and the experimental data is equal to 0.03907. Linear ARX polynomial model can represent the engine with a fair accuracy.

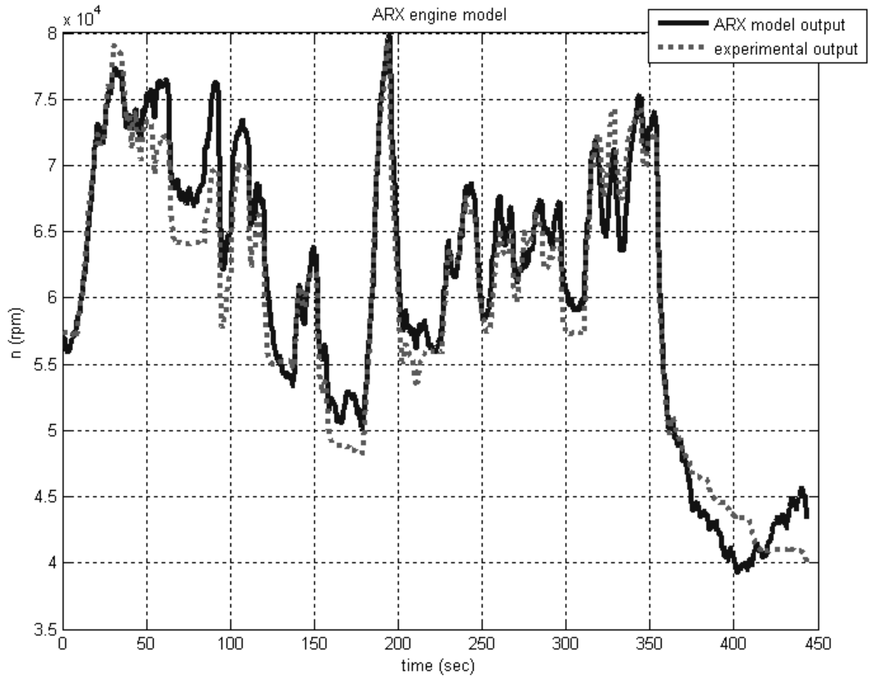


Fig. 7 Comparison between ARX model output and measured data

Design of the Neural Network Topology

Design of the appropriate neural network topology involves several important steps:

1. Choosing the appropriate neurons' transfer functions,
2. Basic decision about the amount of neurons to be used in each layer,
3. Selecting the amount of hidden layers.

In this paper a feed forward neural network was built to model the SR-30 engine. Function approximation has been traditionally one of the most researched uses of neural networks. Typically, a two or three layer networks are sufficient to approximate complex functions with a finite number of discontinuities. In order to gain an insight as to how topology affects the outputs, tangent sigmoid, logarithmic-sigmoid and pure linear neuron transfer functions were selected and tested for further investigation. A network topology study was conducted in order to find the most appropriate architecture neural network to fit SR-30 engine speed parameter. Note that there are several hundred combinations of neurons and layers but for practical purposes we tested four candidate topologies shown in Table 2. The training input data sets for this part of the experiment are 1018 data points selected from the engine testing and then tested (or generalized) with 370 data points selected from the same testing data.

Training the Neural Network

In order to simplify the analysis for development and training of the neural network models the MATLAB

Neural Network Toolbox was employed. MATLAB is a general mathematical package produced by the Math works Company [6,7]. This tool is very efficient to handle matrices and was used throughout this research project to handle data manipulation tasks and neural network computations.

Table. 2 Summary of Neural Network Training and Testing.

No of hidden layer	2	2	2	2
no of neurons in first hidden layer	10	10	10	8
no of neurons in second hidden layer	10	5	1	6
fn of the layer	Logsig	logsig	logsig	logsig
Mean square error	0.05713	0.01881	0.1775	0.007316
computation time (s)	0.18499	0.128	0.185	0.124
simulation time (s)	143.444	143.444	143.444	143.444

From the previous study we conclude the following result. The neural net work consists of two hidden layer with logsig function and the number of neurons in the first hidden layer is 8 and second hidden layer is 6.the function of the output layer is a linear function as shown in Fig. 2.this is structure of the neural networks gives minimum mean square error with value of (0.007316).

Selection of Training Algorithms

Based on the analysis performed with several transfer functions in the neural network the Levenberg-Marquardt algorithm was found to be the most efficient and reliable means to be used for this study. The neural network employed in the SR-30 model is based on non-linear optimization techniques. The objective of the optimization is to train the network parameters weights (w) and biases (b) so they can be adjusted in an effort to optimize the performance of the network. Neural networks are taught to accommodate changes in the weights and biases to appropriately reconfigure the output. During each training operation the error between the output and target becomes smaller until a minimized error goal is achieved. These weighs and biases are somewhat equivalent to the regression constants found in many nonlinear multivariate estimation models and thus can be easily incorporated in any programming environment that supports array manipulation. The experimental data from experiment (4) is used for network training. The results of the neural net work training were shown in Fig. 8 and Fig. 9). The average value of the mean square error Fig. 10 between the model output and the experimental data was equal to 0.007316 the model computation time was 0.124 s while the measured data running time was 443.412 s.

Validation of the Neural Network Data

The validation of the neural model is performed with a different input data to check the accuracy of the model. The experimental data from experiment (3) is used for network validation. The result of the validation process is shown in Figure (12). The average value of the mean square error between the model output and the experimental data is equal to 0.01434.

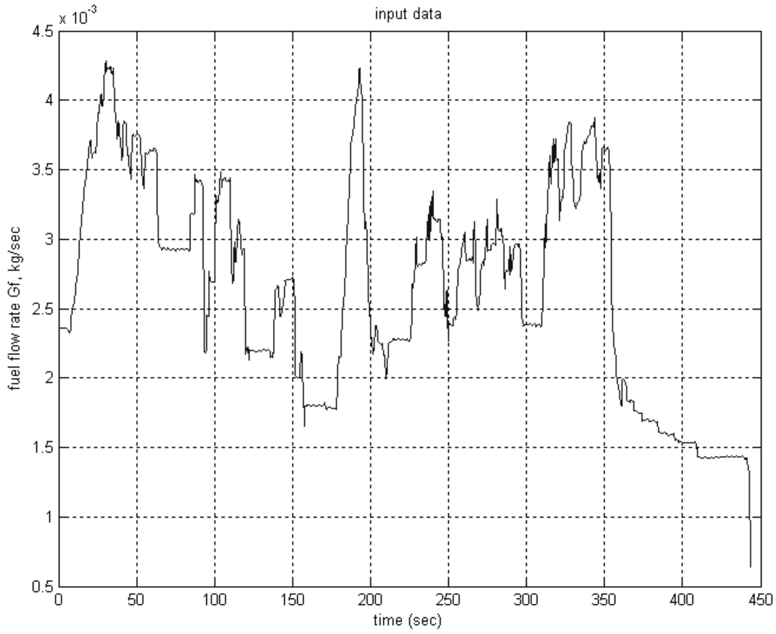


Fig. 8 Input measured data

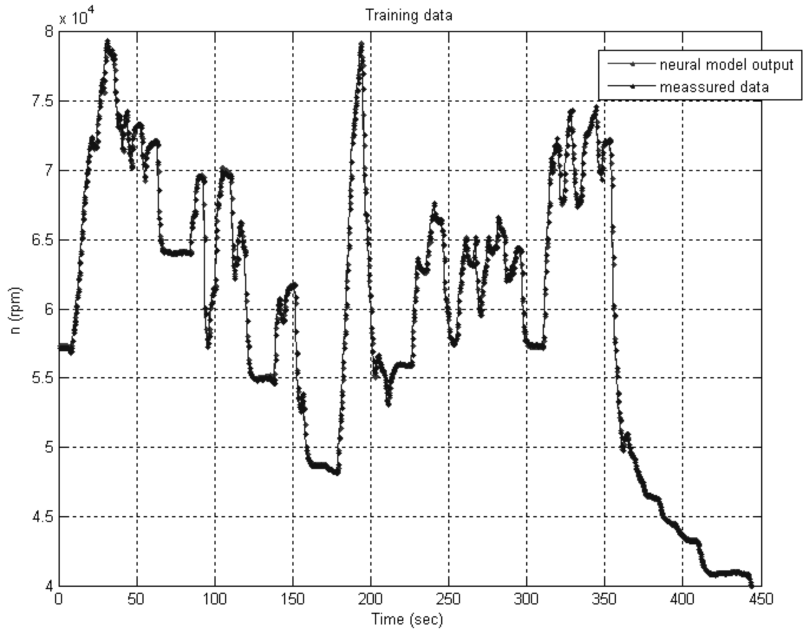


Fig. 9 Comparison between neural output and measured data

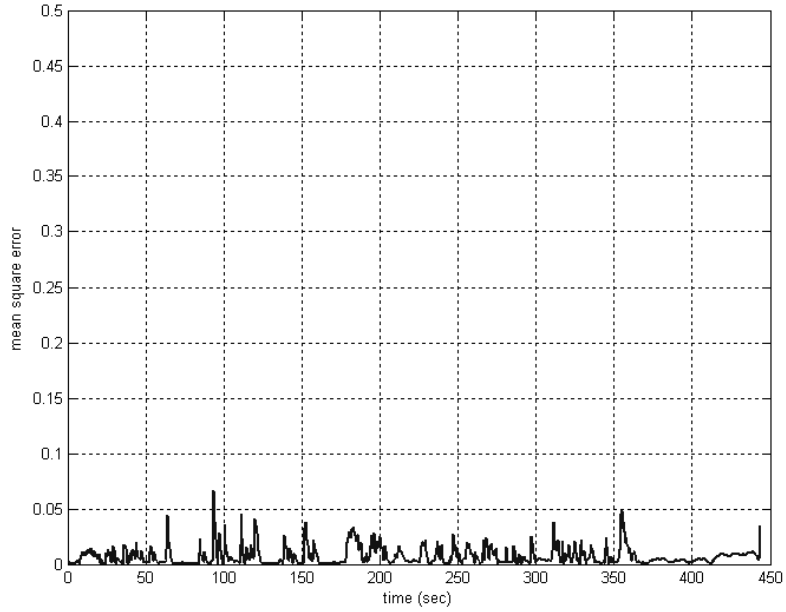


Fig. 10 Mean square error result

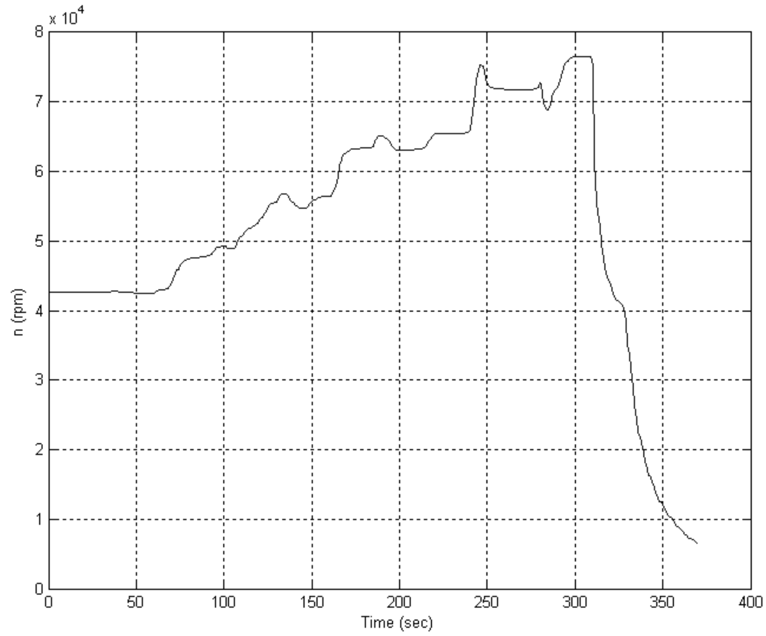


Fig. 11 Measured data used for model validation

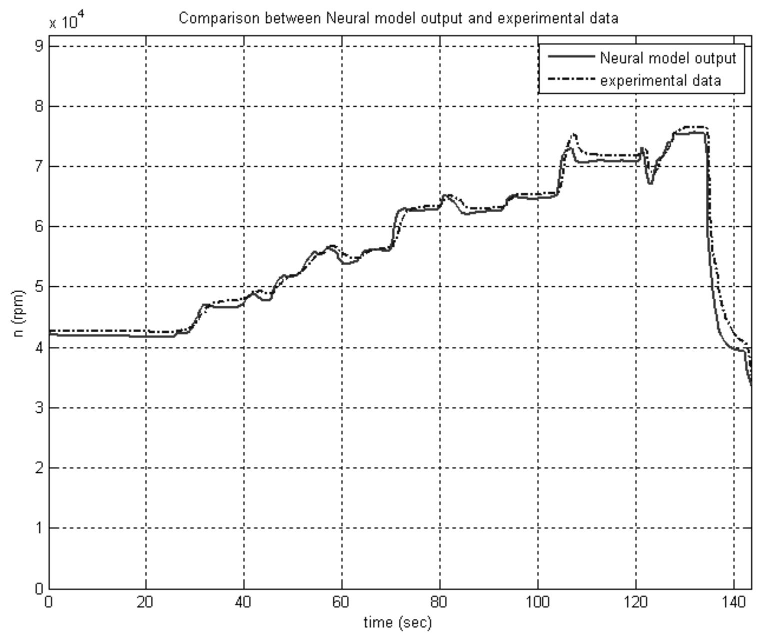


Fig. 12 Validation data results

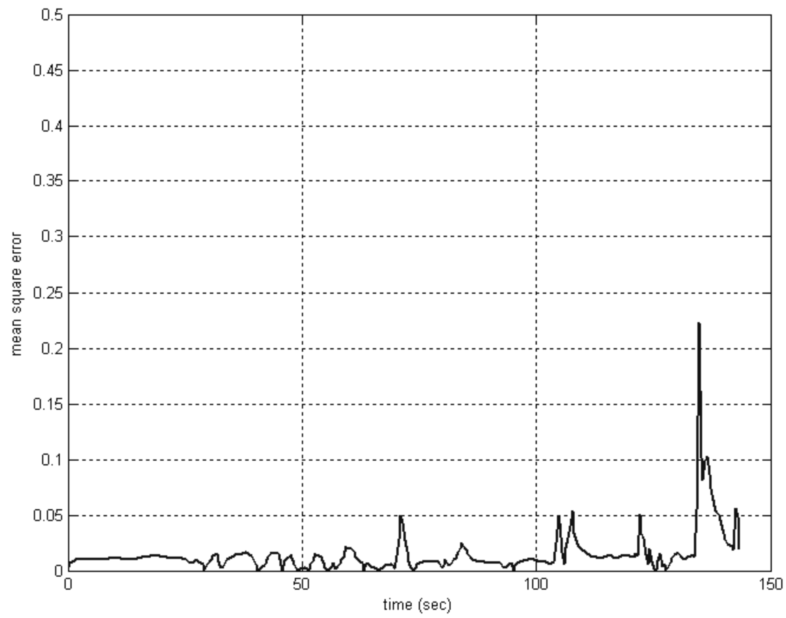


Fig. 13 Mean square error result for validation data

A comparison between the neural model and ARX linear model with the same data was shown in Fig. 14 and it is clear that the neural model can simulate the engine dynamics through its operating rang with good accuracy.

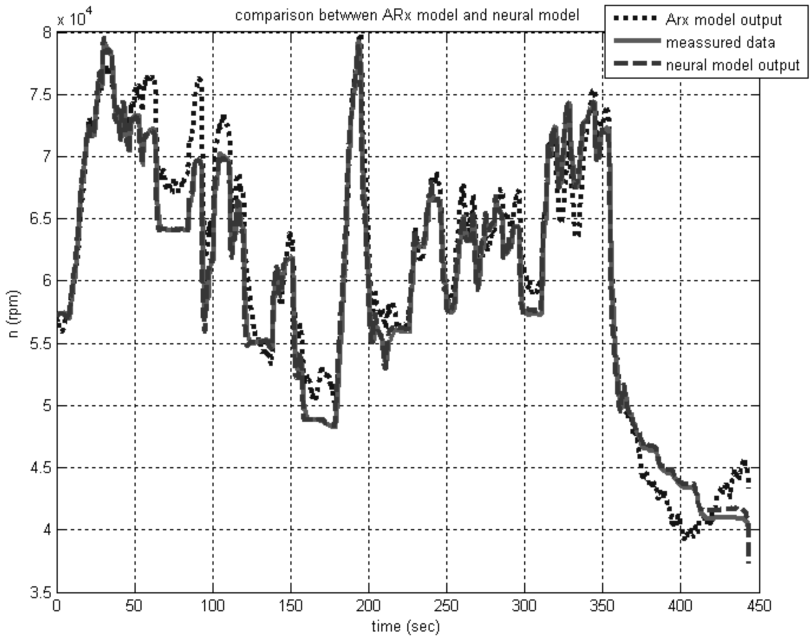


Fig. 14 Comparison between ARX and neural model with experimental data

Conclusion

A representative neural network SR-30 engine model was developed using data given from the testing of the engine. The neural network is trained to estimate the engine revolution speed (n) as a function of the fuel flow rate (G_f). Results from the neural model are compared to the actual performance provided in engine testing data with mean square error of value (0.007316). The results are adequate for the implementation in real-time simulation where the model computation time is 0.124 s while the measured data running time is 443.412 s .

The following conclusions are derived from our analysis:

1. The model developed in this research project purely addressed the SR-30 engine
2. The information provided from the testing of the engine is a reliable source to study the performance of the SR-30 turbojet engine.
3. The system identification results obtained in this paper indicate that linear ARX polynomial model can represent the engine with a fair accuracy
4. Along with neural network technology, a neural network model has been developed. Results obtained from the neural network engine model show that a neural network with proper training is an accurate and efficient mean to calculate the engine speed. Neural networks can approximate with good accuracy the dynamics of the engine through its operating range.
5. The comparison made with the results obtained from the linear ARX model and nonlinear neural model shows clearly the superiority of neural networks as a tool in nonlinear system modeling.
6. The neural networks model cannot be build during the early phase of the engine design because the neural model needs a measured data from the engine to be modeled.

References

[1] Hung, W.W. "Dynamic Simulation of Gas-Turbine Generating Unit" IEE PROCEEDINGS-C 138: 342-350, JULY 1991

[2] Ailer, Piroska, Imre Santa, Gabor Szederkenyi, and M. Katalin Hangos. "Nonlinear Model-Building of a Low-Power Gas Turbine" Periodica Polytechnica Ser. Transp. Eng. 29: 117-135, OCTOBER 2001.

[3] Ailer, P., G. Szederkenyi, and K.M. Hangos. "Model-Based Nonlinear Control of a Low-Power Gas Turbine" 15. IFAC Triennial World Congress. Barcelona, 2002.

[4] Lichtsinder, Michael, and Yeshayahou Levy. "Jet Engine Model for Control and Real-Time Simulations" Journal of Engineering for Gas Turbines and Power 128: 745-753, OCTOBER 2006.

[5] Camporeale, S.M., B. Fortunato, and M. Mastrovito. "A Modular Code for Real Time Dynamic Simulation of Gas Turbines in Simulink" Journal of Engineering for Gas Turbines and Power 128: 506-517, JULY 2006.

[6] Koçer, Gülru, and Oğuz Uzol. "Real-Time Simulation of a Small Turbojet Engine" 4. Ankara International Aerospace Conference. Ankara, 2007.

[7] Kim, J.H., T.W. Song, T.S. Kim, and S.T. Ro. "Dynamic Simulation of Full Startup Procedure of Heavy-Duty Gas Turbines" Journal of Engineering for Gas Turbines and Power 124: 510-516, JULY 2002.

[8] Evans, C., Fleming P.J., Hill D.C., Norton J.P., Pratt I., Rees D. and Rodriguez-Vazquez K., 2001, “Application of System Identification Techniques to Aircraft Gas Turbine Engines” Control Engineering Practice, 9, pp. 135-148.

[9] Rodriquez, K., 1999, “Multi Objective Evolutionary Algorithms in Non-Linear System Identification”, Ph.D. Thesis, University of Sheffield, Department of Automatic Control & Systems Engineering, UK.

[10] Chiras, N., Evans, C. and Rees, D., 20001, “Nonlinear Gas Turbine Computer Modeling Using NARMAX Structures” IEEE Trans. Instrumentation and Measurement, August issue.

[11] Chiras, N., Evans, C. and Rees, D., 2001, “Estimating Global Nonlinear Models of Gas Turbine Dynamics Using NARMAX Structures” GT-0019, New Orleans.

[12] Neophytos Chiras, Ceri Evans and David Rees ” Nonlinear Gas Turbine Modeling Using Feed Forward Neural Networks” ASME TURBO EXPO 2002, GT-2002-30035

[13] Watanabe, A., lmen, S. M., Leland, R., Whitaker, K. W., and Trevino, L. C., 2004, “Soft Computing Applications on SR-30 Turbojet Engine” AIAA Paper No. 2004–6444.

[14] Andoga, Rudolf, Ladislav Madarasz, and Ladislav Fozo. "Digital Electronic Control of a Small Turbojet Engine - MPM 2." 12. International Conference on Intelligent Engineering Systems. Miami, Florida. 37-40, 2008.

[15] D. May, S. Farber, Vincient Poirier, Chien Yu Chen, Dr. Ion Stiharu “Design of a Comprehensive Condition Monitoring System for Gas Turbine Engines”.

[16] O. Léonard. J. P. Thomas. S. Borguet “Ten Years of Experience with a Small Jet Engine as a Support for Education’ Journal of Engineering for Gas Turbines and Power”. January, 2009, Vol. 131

[17] EcosimPro by EA Internacional, http://www.ecosimpro.com

[18] M.Ghoreyshi, P.Phillids, K.W.Ramsden,”Diagnostics of a Small Jet Engine-Neural Networks Approach”Volume 1: Turbo Expo 2005, GT2005-68511 pp. 629-637

[19] Fausett L., 1994, “Fundamentals of Neural Networks: Architectures, Algorithms and Applications” Prentice-Hall.

[20] Baughman D. R., Liu Y. A., 1995 “Neural Networks in Bioprocessing and Chemical Engineering”Academic Press.

[21] Smith, L.. “An Introduction to Neural Networks” University of Stirling” (2002).

[22] Cybenco, G., “Approximation by Superpositions of a Sigmoidal Function,” Mathematics of Control, Signals, andSystems, 2, 4, pp. 303-314. 1989,
[23] Cruz,C. “Understanding Neural networks” A Primer. Clutter Information Corps, USA. ISBN 0-943-779-693. (1991).
[24] Park, J. and I. Sandberg, “Universal Approximation Using Radial-Basis Functions Networks,” Neural Computation, 3, pp.246-257. 1993.
[25] Alexander, I., “Neural Computing Architecture” , MIT Press, 1989.
[26] Chicago, “Mini-Lab SR-30 Gas Turbine Engine Document”, Turbine Technologies, Chetek, WI, 2004. September 20-22, 2004.
[27] Gas Turbine Power System Operator's Manual Turbine Technologies, LTD, 2006.
[28] Norgaard M., Ravn O., Poulsen K.N. and Hansen L.K,” Neural Networks for Modelling” and Control, Springer-Verlag, London, 2001.
[29] Neural Network Toolbox User’s Guide, the Mathworks, Feb 2010.