A NONLINEAR NEURAL NETWORK BASED MODEL PREDICTIVE CONTROL FOR INDUSTRIAL GAS TURBINE

Ibrahem M.A Ibrahem
Military Technical College, Aircraft Mechanics Department
EGYPT

Ouassima Akhrif
Ecole de technologie supérieure, Department of Electrical Engineering
CANADA

Hany Moustapha
Ecole de technologie supérieure, Department of Mechanical Engineering
CANADA

Martin.Staniszewski
Siemens Energy Canada Limited, Digitalization Project Manager
CANADA

ABSTRACT

Gas turbines are now extensively used in aviation, oil and gas applications and power generation. With this increasing use in a diverse range of applications, new design and operation of modern gas turbine engines (GTEs) are becoming more and more complex where several limitations and control modes should be fulfilled at the same time to accomplish a safe and ideal performance for the engine. For this purpose, a constrained multi-input multi-output (MIMO) non-linear model predictive controller (NMPC) based on neural network model is designed to fulfill the control requirements of a Siemens SGT-A65 three-spool aero-derivative gas turbine engine (ADGTE) used for power generation. However, the implementation of NMPC in real time has two challenges: Firstly, the design of an accurate non-linear model, which can run many times faster than real time. Secondly, the usage of a rapid and reliable optimization algorithm to solve the optimization problem in real time. In this paper, a novel approaches for gas turbine engine modelling and multivariable advanced controller design are investigated. The constrained MIMO NMPC is created based on the generalized predictive control (GPC) algorithm as a result of its clarity, ease of use, and capacity to deal with problems in one algorithm. In addition, seven ensembles of eight multi-input single-output (MISO) non-linear autoregressive network with exogenous inputs (NARX) models are used as a base model for the GPC controller to predict the future process outputs. Estimation of free and forced responses of the GPC based on the neural network (NN) model of the plant each sampling time without performing instantaneous linearization is proposed in this study, which reduces the NMPC optimization problem to a linear optimization problem at each sampling step. In addition, the Hildreth's quadratic programming algorithm is used to solve the quadratic optimization problem within the NMPC controller, which offers ease of use and reliability in real time applications. To demonstrate the performance of the NNGPC controller developed in this study, we have compared the performance of the neural network generalized predictive control (NNGPC) controller to the existing controller of the SGT-A65 engine. The simulation results show that the NNGPC has demonstrated output responses with less oscillatory behavior and smoother control actions to the sudden variation in the electric load disturbance than those observed in

the existing min-max controller. In addition, computation time required to solve an optimization problem was sufficiently faster than the sampling rate that allow a real time implementation of the NNGPC controller.

1 INTRODUCTION

The importance of aero-derivative gas turbine engines (ADGTEs) in the energy industry has sparked a great interest among manufacturers to improve the performance and increase the reliability of the engine, which in turn requires an accurate and real time model to simulate the engine dynamics during the full operating range. This model can be used in controller design, engine health monitoring and sensor validation. Physics based modelling (White box model) approach has been widely used over many years in order to model gas turbine engines Saleh (2017). However, this approach can only be used when there is enough information about the physics of the system. In addition, a white box model has traditionally a high number of non linear equations requiring iterative solutions which occur at the expense of computation time. The computation time challenge is a big problem for real time modelling and model based applications. An alternative to white box models is given by data-driven based models.

Data-driven based model (or black box model) is one of the modelling approaches which can be used when no or little information is available about the physics of the system. In this case, a data driven model can disclose the relations between system variables using the obtained operational input and output data from the system. Artificial neural network (ANN) is one of the most significant methods in data-driven based modelling Asgari et al. (2013). It presents high computation speed, which allows for real time applications. It is a fast-growing method, which has been used in different fields of industry in recent years. It is also being heavily used in machine learning and artificial intelligence applications. The main idea behind ANN is to create a model based on a human brain in order to solve complex scientific and industrial problems in a variety of areas.

Neural networks can be classified into two main categories: static and dynamic neural networks Ibrahem et al. (2019). Dynamic neural networks attracted many researchers due to their ability to represent the dynamics of gas turbine engines. A significant number of studies across different applications have stated the advantages of dynamic NNs by introducing different methodologiesTayarani-Bathaie et al. (2014), Yu & Shu (2017). Among the existing dynamic NN modelling methods, the non-linear autoregressive network with exogenous inputs (NARX) modelling approach is considered one of the most popular. It has been used in the modelling of gas turbine engines by several researchersBahlawan et al. (2017), Mehrpanahi et al. (2018).

NARX is a recurrent dynamic network with feedback connections enclosing several layers of the network. NARX model is based on the linear ARX model, which is commonly used in time-series modelling, and is used in many applications such as multi step ahead prediction and modelling of non-linear dynamic systems. Eqn. (1) defines a NARX model and represents the relation between the model output and its input parametersBeale et al. (2015),

$$\begin{aligned} y(t) = f(y(t-1), \cdots, y(t-n_y) \\ , u(t-n_k), \cdots, u(t-n_k-n_u+1)) \end{aligned} \tag{1}$$

where, n_y and n_u are the lags of the output and input of the system respectively. n_k is the system input-output delay and f is a non-linear function.

As can be seen, the major challenge to the application of ANNs is to find the best structure of the network which can represent the system. In this regard, usage of single neural model may not be able to provide accurate prediction when it operates outside the field in which it was trained. Besides, gas turbine engines operate in non-stationary operation conditions which may cause unseen scenarios in the observed data this increases the complexity of the modelling operation since the NN needs to be trained with a data as large enough to cover the entire operation conditions. This in turn increases the training time and the possibility of network over-fitting. To address this problem, we consider to train multiple ANNs in parallel to fit the data instead of a single model. This leads to an ensemble of neural network models. Ensemble learning approach refers to a set of models working in parallel on tasks such as classification or regression and they are combined together in some way to obtain the final output de Sousa et al. (2012). In this paper, we will focus on the ensemble generation for regression. The ensemble development process can be divided into three main steps. The first step is ensemble generation, which refers to generation of ensemble base models. The second step is ensemble pruning, which consists of selecting a subset of the best models from the original set of models based on generalization error. Finally, ensemble integration, a strategy to combine the base models is defined. For regression problems, ensemble integration is done using a linear combination of the base models outputs de Sousa et al. (2012),

$$f_{en}(x) = \sum_{i=1}^{K} [w_i(x) * f_i(x)] \tag{2}$$

where, $w_i(x)$ denotes the weight for the ith model, K is the number of models in the ensemble, $f_i(x)$ denotes the output of the ith model corresponding to input x and $f_{en}(x)$ represents the ensemble output. Diversity is a very important key in the ensemble generation. If all ensemble members provide the same output, there is nothing to be gained from their combination Zhang & Ma (2012). Therefore, the ensemble members should be different from each other while each must maintain acceptable accuracy level Amozegar & Khorasani (2016). Two different methodologies can be considered for creating diversity among ensemble members. The first method is heterogeneous ensemble in which ensemble members have different architectures(such as number of neurons, training algorithm). The second method is homogeneous ensemble in which ensemble members have the same architectures but trained with different data sets Brown et al. (2005). The integration of a set of learned models to improve accuracy and generalization is another important step in the ensemble generation. Ensemble integration approaches can be divided into two categories,constant and non-constant weighting functions Merz & Pazzani (1999). Examples of the constant weight approach are basic ensemble method (BEM), generalized ensemble method (GEM), linear regression (LR) and median method. For the second category, the weights vary according to performance of each ensemble member such as dynamic weighting (DW), dynamic weight with selection (DWS)Rooney & Patterson (2007) and dynamic and on-line ensemble regression (DOER)Soares & Araújo (2015).

Control technology has an important role in the evolution and progress of the performance, dependability, lifetime, and safety of modern GTEs. The power generated from a GTE is controlled by fuel flow to ensure that the desired power output is fulfilled. However, the control system must also protect the engine from exceeding engine limits. These limits incorporate shaft speeds, temperatures, and compressor surge. There are many control techniques proposed to manage these requirements going back to 1952. A comprehensive survey and investigation on the history of GTEs control techniques could be found in Jaw & Mattingly (2009).

The Min–Max control strategy is commonly utilized as the control architecture for GTEs, which is known as a practical algorithm. A min-max controller composed of several control loops (steady state loop, max ac-

celeration loop, max shaft loop speed, max temperature loop, and min deceleration loop), every one of which controls the engine during certain controlling mode. These loops operate in parallel and at any moment, as indicated by a predefined fuel control technique, one of them is selected and starts the observation Salehi & Montazeri-GH (2020). Many studies have done for performance improvement of the min-max strategy Jafari & Nikolaidis (2018). However, the final selection technique between the transient control loops is kept fixed in all these researches. In addition, limit violation may happen for some variables during transient operation. Recent investigations have demonstrated that there is no assurance for a min-max strategy with a linear compensator to protect the engine from exceeding constraints during the transient state Imani & Montazeri-Gh (2017), Montazeri-Gh & Rasti (2019).

With the desire to have more robustness and flexibility of the next generation of the GTE control systems to accomplish ambitious objectives and extreme constraints set by governments and organizations, for example, a decrease of nitrogen oxide (NOx) emission, a decrease of fuel consumption, and increase of engine life time, the industry is keen on building up another advanced control strategy that will fulfill the mentioned requirements. Model predictive control (MPC) is an advanced model-based technique, which has attracted the attention of researchers in recent years. The application of MPC to control GTE is introduced by Vroemen and Essen Van Essen & De Lange (2001).

The philosophy behind the construction of MPC laws is paradoxical when compared with the conventional error-feedback control methods. Rather than creating a control action in response to the current and past errors, MPC makes a forecast of future system behavior dependent on its open loop model, the current system state, the input trajectory, or a disturbance entering the system. Then, it chooses the best possible input action according to a cost function. To do this, it needs to solve an optimization problem subjected to possible constraints. Finally, it applies the first element of the optimal selected input sequence to the plant. At each time step, this procedure is repeated, which introduces the so-called receding horizon principle Richter (2011). Fig. 1 shows the strategy of all MPC controllers.

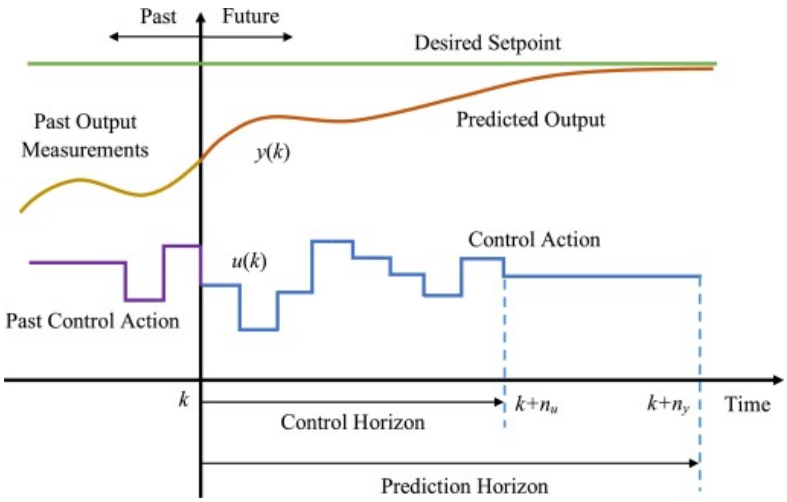

Figure 1: The MPC Strategy Montazeri-Gh & Rasti (2019)

There are different MPC algorithms just differ among themselves in the onboard model used to simulate the controlled process. The MPC algorithms which utilize for prediction linear models are usually named linear

MPC algorithm (LMPC) Pandey et al. (2018). However, The MPC algorithms which utilize for prediction non-linear models are usually named non-linear MPC algorithm (NMPC) Pires et al. (2018). The LMPC algorithms are easy to use and computationally uncomplicated. However, the acquired control quality might be unsuitable for nonlinear systems, specifically when the operating point is changed significantly and fast Kim et al. (2013). On the other hand, the utilization of NMPC controllers that consider the non-linearities of the process infers an improvement in the performance of the process by decreasing the effect of the disturbances and by improving the tracking abilities of the control system. However, the usage of a non-linear model inside the NMPC algorithm converts the control problem from a convex quadratic program to a nonconvex non-linear problem Poursafar et al. (2010). In addition, in this circumstance, there is no assurance that the global optimum can be found particularly in real time control when the optimum solution must be determined in a prescribed time. The computation time challenge is a major issue for the real time implementation of the NMPC. In addition, in the case of a model-based controller, the onboard model, which represents the plant inside the controller, must run many times faster than real-time. Besides, it should be accurate and can represent the plant dynamics during the full operating range.

The implementation of NMPC based on NN model is able to eliminate the most significant problems experienced in non-linear predictive control applications, because NN gives a helpful way for modelling complex non-linear systems with good accuracy and less computational complexity. The MPC algorithm based on NN model has recently attracted many researchers (Rusnak et al. 1996). Two principle approaches were utilized to use a non-linear neural model within a predictive control algorithm (Aly & Atia 2012). The first approach is to utilize the non-linear optimization methods to calculate the optimum control actions (Lazar & Pastravanu 2002). The second approach is to linearize the non-linear neural model each time step to get the discrete linearized model (Mu et al. 2005). The advantage of utilizing Lineraization method over conventional non-linear design is the avoidance of the problem of local minimums. However, this could result in a large computational load for MIMO systems. Furthermore, the lineraized model is only an approximation of the original non-linear one. Therefore, the obtained control quality may be unsatisfactory as mentioned before.

Based on the literature survey, we can note the following issues:

1) In the area of MIMO ANN model of gas turbine engines, the research activities used mostly one of the following two methods to generate a nonlinear model for the MIMO engine: Either, by building a neural network model for each output parameter (MISO) with the same structure for each one of them and trained with the same training algorithm Bahlawan et al. (2017), or by building one block neural model to represent the MIMO system Mehrpanahi et al. (2018), Salehi & Montazeri-Gh (2018). However, it is more powerful, as will be shown in this paper, to use a different neural network's structure for each output (MISO). The idea of using different NN structures for different outputs comes from analysis of the structure of the human brain based on its function. Human brain consists of many complex biological neural networks; each has to perform a certain function [talk, walk, breath and so on] as shown in Fig. 2. The main component of these neural networks are neurons. The shape and number of the neurons in each network depend upon the function of the network. For example, a single sensory neuron from your fingertip has an axon that extends the length of your arm, while neurons within the brain may extend only a few millimetres. They also have different shapes depending on their functions Khan (2018). Motor neurons that control muscle contractions have a cell body on one end, a long axon in the middle and dendrites on the other end. Sensory neurons have dendrites on both ends, connected by a long axon with a cell body in the middle. Inter-neurons, or associative neurons, carry information between motor and sensory neurons as shown in Fig. 3. Based

on these facts, to generate a neural network model for an ADGTE, we propose to build MISO neural network model for each output. Moreover, the structure of each neural model is different according to the function of the model (output type).

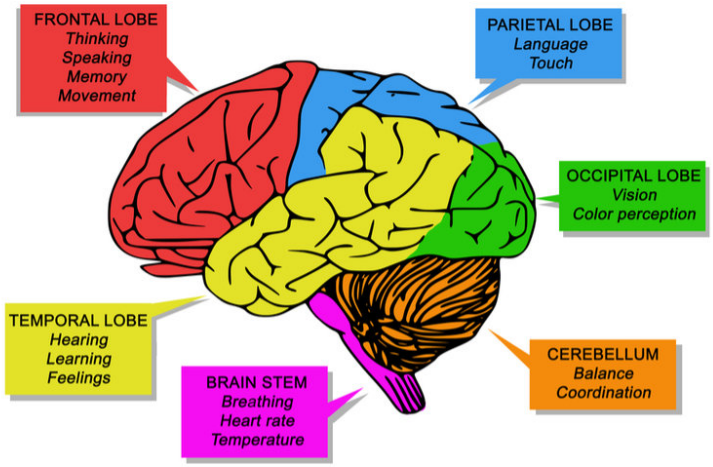

Figure 2: Functions of the human brainCASTRO (2018).

2) Most works on ensemble approach focus on classification applications for tackling the FDI problem in gas turbine engines Lu et al. (2019), Wen et al. (2019). However, little research has been done on the use of ensemble approach for gas turbine engine real time performance prediction Amozegar & Khorasani (2016), Xu et al. (2017). Therefore, this paper presents a novel development approach of the data driven model for ADGTE based on recurrent neural networks (NARX). Firstly, closed loop experimental data preprocessing was performed as will be explained in the data acquisition and preprocessing subsection. Secondly, generation of multiple MISO ARX models was used to identify the order of the MISO systems to be modelled based on the experimental data.

3) The Min–Max control strategy is the most widely used control algorithm for industrial GTEs. This strategy uses minimum and maximum mathematical functions to select the winner of different engine control loops at any instantaneous time. However, recent studies (Imani & Montazeri-Gh 2017, Montazeri-Gh & Rasti 2019) indicate that this method with linear compensators suffers from lack of safety guarantee in fast load demands. On the other hand, MPC method, which incorporates input/output constraints in its optimization process, has the potential to fulfill the control requirements of an industrial GTEs.

4) The LMPC algorithms are simple to design and are computationally uncomplicated. Unfortunately, the obtained control quality may be unsatisfactory for nonlinear systems, in particular when the operating point is changed significantly and fast. In such cases non-linear models are straightforward, but their identification is more demanding. Furthermore, complexity of NMPC algorithms is higher than that of the classical linear ones. The implementation of NMPC of ADGTE in real time has two challenges: Firstly, the design of accurate non-linear model which can run many times faster than

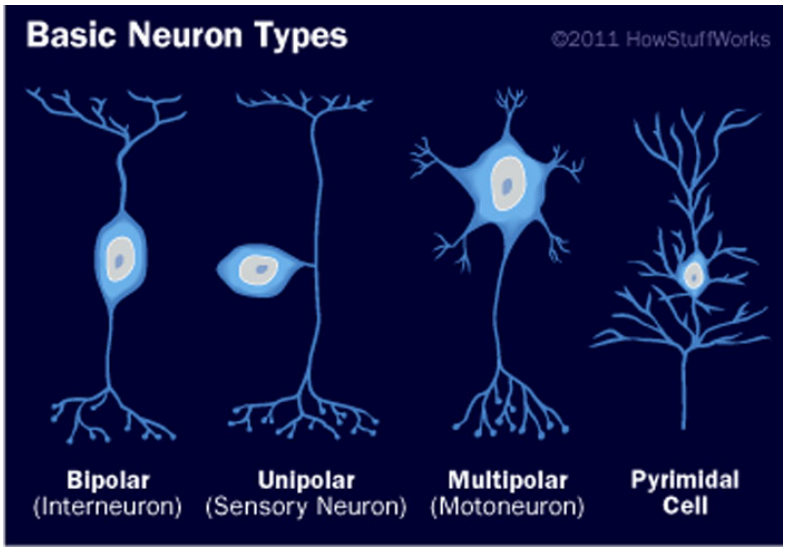

Figure 3: Basic neuron typesKhan (2018).

real time. Secondly, the usage of rapid and reliable optimization algorithm to solve the optimization problem in real time.

5) The resulting implementation of NMPC based on NN model is able to eliminate the most significant obstacles encountered in non-linear predictive control applications by facilitating the development of non-linear models and providing a rapid, reliable solution to the control algorithm.

6) The use of NN models together with the GPC algorithm is a promising technique. Most applications of GPC algorithm based on NN model have used instantaneous linearization of the NN model at each time step. In this thesis, to design an NMPC of ADGTE, a trade-off approach between usage of non-linear model and successive linearization approach is used in order to reduce the computation effort. In addition, in order to solve this problem, Hildreth's quadratic programming procedure is utilized which offers simplicity and reliability in real-time implementation.

The rest of this paper is organized as follows: Section two illustrates the specifications of the ADGTE used in this study and Section three provides a detailed look at the methodology of building operation of ensemble of MISO NARX models for an ADGTE. Section four presents an overview of the current min-max controller of the SGT-A65 engine. Section five presents and discusses the results obtained from this study and Section six presents the conclusion.

2 GAS TURBINE ENGINE DESCRIPTION

2.1 Overview of the SGT-A65 Engine Configuration

In this paper, Siemens (SGT-A65) three spool ADGTE was modelled. The Siemens SGT-A65 is one of the world's leading ADGTE used in the power generation and oil-and-gas compression industries. It is the industrial

version of the Rolls-Royce Trent 60 high by-pass-ratio aero GTE, which has high efficiency, and in service on the Airbus A330 and Boeing 777. The SGT-A65 is capable of producing 65 MW at thermal efficiency of 42% (H.I.H. Saravanamuttoo 2017). It has a two-stage LPC, eight-stage IPC, and six-stage HPC with dry low emission combustion. Furthermore, both HPT and IPT consist of a single stage each and the LPT has five stages used to drive the LPC and the power generator at fixed speed (3600 rpm used for power generation at 60 Hz, which represents the engine power turbine (PT). Fig.4 shows a sketch of SGT-A65 aero derivative gas turbine engine with its stations numbers. In addition, the engine specifications are illustrated in Table 1.

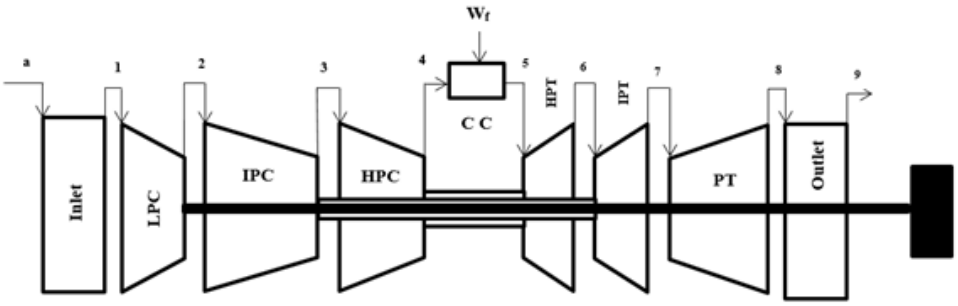

Figure 4: Sketch of three spool aero derivative gas turbine engine (SGT-A65).

Table 1: SGT-A65 engine technical data.

Parameter	Value
Exhaust mass flow rate	171 kg/s
Output power	65 MW
Power turbine speed	3600 rpm
Total compression ratio	38 : 1
Exhaust temperature	437 ^oC

2.2 Overview of the SGT-A65 Engine Current Min-Max Control System Architecture

The Industrial Trent (SGT-A65) engine control system schedules the fuel flow to maintain the engine power (**PW**) or speed to the desired level (3600 rpm). While, maintaining the other parameters of the engine, such as spool speeds (**N**), temperatures (**T**) and pressures (**P**) within its operating limits at all times. The min-max controller with PI compensator in each loop is currently used for the SGT-A65 ADGTE.

The overall control architecture of the SGT-A65 engine is shown in Fig. 5. As can be seen, the fuel control system based on multiple SISO loops all vying for control of the engine fuel flow (WF) through the loop selection logic, which is a series of highest and lowest wins gates (min-max algorithm). The basis of this technique is to control a primary output utilizing a single control input while keeping up the other intended outputs below their limits. The proportional and integral gains are taken from a lookup tables. These gains are tuned in order to provide the required bandwidth and stability margins across the operating envelope.

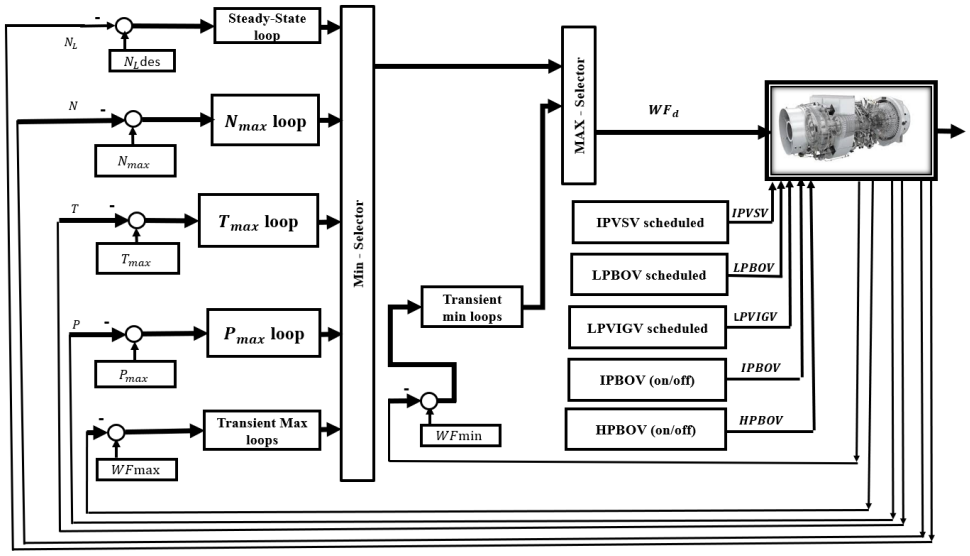

Figure 5: Engine fuel control system using min-max controller

In addition, to protect the engine during the acceleration or deceleration operation five manipulated parameters were used by the ECS to protect engine as shown in Fig. 5. These manipulated parameters are IPVSV, LPVIGV, LPBOV, IPBOV, and HPBOV. This air system control is used to maintain peak performance and adequate surge margin during engine acceleration and deceleration.

3 ARTIFICIAL NEURAL NETWORK MODELLING APPROACH

This paper presents a novel methodology for modelling ADGTE using ensembles of NARX neural networks. This is a general methodology which can be used in generation of accurate, generalized and real time black box models and has the advantage of greatly reducing the network training time. The flow diagram of the modelling approach is illustrated in Fig. 6.

3.1 DATA ACQUISITION AND PRE-PROCESSING

The datasets which were used in this study are time series datasets consisting of six input parameters (WF, VIGV, LPBOV, IPVSV, IPBOV, and HPBOV) and seven output parameters (N_L, N_I, N_H, PW, T_{30}, P_{30}, and TGT) representing the engine operation from the Synch-Idle or Synch speed (3600 rpm) with no load (unloading) regime to Synch speed with full load regime (loading). The starting and shutdown regimes are not represented in this work.

Note that, these datasets were taken at different operation conditions: T_{amb} , P_{amb}, LHV and η_{th}. The η_{th} was used to represent the performance degradation of the engine, which is defined as the ratio between the output shaft energy to the added energy as shown in Eqn. (3)H.I.H. Saravanamuttoo (2017). The only available closed loop experimental data from testing of SGT-A65 engine was collected at a specific operation condition. More

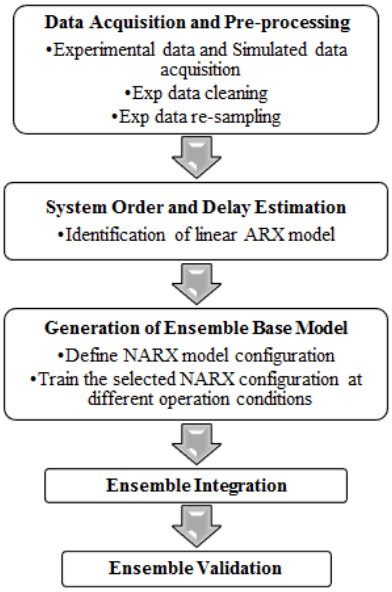

Figure 6: The flow diagram of the modelling approach.

data was needed however to train NN models at different operation conditions and use those models to generate the NN ensemble. To generate data at different operation conditions we used Siemens high fidelity thermodynamic model. The accuracy of this model has been well validated by Siemens, so combining experimental data with high fidelity simulated data was the best solution given the limited availability of experimental data. Table 2 shows more details about these datasets, where TR is the training dataset and TS is the testing dataset.

$$\eta_{th} = \frac{PW}{WF * LHV} \tag{3}$$

The experimental time series datasets $TR1_{exp}$ and $TS1_{exp}$ are used for pre-processing operation for extracting the most valuable information about the system dynamics to use in identification of ANN models. Firstly, the experimental datasets $TR1_{exp}$ and $TS1_{exp}$ are cleaned by filtering any measurement noise. A simple second order low pass Butterworth digital filter with cut-off frequency 0.5 Hz was used to remove noise from experimental datasets. After that, cleaned datasets $TR1_{exp}$ and $TS1_{exp}$ were re-sampled at a lower frequency F_s =10 Hz instead of high sampling frequency 20 Hz to avoid aliasing effects and to avoid having the poles of the discrete models being very close to the +1 point on the unit circle. In addition, the lower sampling rate reduces the number of data points which reduces the computation time during training operation and reduces data collinearity. The selection of the sampling rate was performed based on the Nyquist sampling criterion, which requires setting the sampling rate at least twice the highest frequency of the system.

3.2 SYSTEM ORDER AND DELAY ESTIMATION

$TR1_{exp}$ and $TS1_{exp}$ datasets are used in this step after cleaning and re-sampling. Selecting the model order and delay is a key first step towards the goal of identification of non-linear NNs model. A selection procedure

Table 2: Time series datasets

Dataset	NO of samples	T_{amb}	P_{amb}	LHV	η_{th}
$TR1_{exp}$	21983	26	101.32	48360	0.32
$TS1_{exp}$	8117	26	101.32	48360	0.32
$TR2_{sim}$	15132	30	101.32	47826	0.4231
$TS2_{sim}$	17231	30	101.32	47826	0.4231
$TR3_{sim}$	22480	15	101.32	47826	0.4067
$TS3_{sim}$	12890	15	101.32	47826	0.4067
$TR4_{sim}$	13172	0	101.32	47826	0.4236
$TS4_{sim}$	12067	0	101.32	47826	0.4236
$TR5_{sim}$	16691	-15	101.32	47826	0.4038
$TS5_{sim}$	8816	-15	101.32	47826	0.4038
$TR6_{sim}$	8732	30	101.32	35717	0.3931
$TS6_{sim}$	3700	30	101.32	35717	0.3931
$TR7_{sim}$	8820	15	101.32	35717	0.4061
$TS7_{sim}$	12750	15	101.32	35717	0.4061
$TR8_{sim}$	15044	-15	101.32	35717	0.4092
$TS8_{sim}$	21065	-15	101.32	35717	0.4092

frequently is implemented by developing several NN models with different orders and delays and comparing models to evaluate their performance. However, the influence of NN parameters such as number of neurons, training algorithm and activation function may lead to an inappropriate selection. In-addition this method is time consuming. Another approach was used in this paper, which starts by estimation of the time delays from inputs to outputs n_k by using a non-parametric estimate of the impulse response using MATLAB® routine *impulseest*. This gives good starting points in identification because it requires minimal user specification Alves et al. (2013).

Secondly, estimation of the system order by generation of a set of candidate MISO ARX models and determining the best model order in the set. The ARX model structure is one of the simplest parametric structures and it is considered a good starting point for identification process because of its simplicity. MATLAB® routine *struc* is used to generate a set of model-order combination of MISO ARX model estimation. The routine *struc* changes n_u and n_y for all inputs and outputs from one to five, as higher-order models result in excessive computational effort and pose the risk of losing particular physical meaning of the model, and uses the input-output delay values n_k from delay estimation step. After that, MATLAB® routine *arxstruc* uses the set of model order combination to estimate a set of MISO ARX models based on the $TR1_{exp}$ dataset. The Final Prediction Error (FPE) evaluates model quality, where the model is tested on another dataset $TS1_{exp}$. Such a procedure is known as cross validation. The most accurate model has the smallest FPE. The FPE equation is defined by the following equation:

$$FPE = \frac{1 + \frac{d}{N}}{1 - \frac{d}{N}} V \tag{4}$$

where d is the total number of estimated parameters and N is the length of the data record. V is the loss function (quadratic fit) for the structure in question. Finally, checking Pole-Zero Cancellations for this selected ARX model is performed. MATLAB allows plotting estimated pole-zeroes with their confidence intervals, which allows you to detect any pole-zero cancellations (confidence ellipse overlap). Removing these provides a trimmed model order. Table 3 summarizes the results from delay and order estimation process for all MISO models for all output parameters of the engine and uses these results in the non-linear NN identification process. This will be illustrated in the next subsection.

Table 3: System order and delay estimation results

Output parameter	n_y	n_u	n_k
N_H	4	[4 4 4 4 4 4]	[1 0 2 0 0 0]
N_I	4	[4 4 4 4 4 4]	[1 1 3 0 0 0]
N_L	4	[4 4 4 4 4 4]	[0 2 1 1 0 0]
PW	4	[2 2 4 4 4 4]	[1 0 3 0 0 0]
TGT	4	[1 4 4 1 1 4]	[1 0 0 0 0 0]
T_{30}	4	[3 1 4 2 1 1]	[6 2 4 3 0 0]
P_{30}	4	[1 2 2 1 1 4]	[0 0 2 1 0 0]

3.3 NARX MODEL CONFIGURATION

This subsection describes the development of multiple MISO NARX models with different configurations to represent each of the engine output parameters. Constructing the MISO NARX model requires determination of network parameters. Such as (i) number of neurons, (ii) number of hidden layers, (iii) hidden layer activation function and (iv) training algorithm. To limit the network complexity, the number of hidden layers is limited to one. Besides, Cybenko (1989) proved that NN with one hidden layer of hyperbolic tangent or sigmoid activation function and one output layer of linear activation function could simulate any non-linear system. Another important parameter in the NARX configuration is the training architecture. The NARX network training can be implemented via two architectures: (i) series-Parallel architecture (S&Pr), where the network is trained in open loop mode then transformed to closed loop mode for validation operation, (ii) parallel architecture (Pr), where the network is trained and validated in closed loop mode. In this paper, to get the optimal NARX model structure which can represent the ADGTE dynamics, we performed an extensive comparative performance study using different combinations of NARX neural network architectures, training algorithms and activation functions while using different numbers of neurons. As a result, a comprehensive computer program was developed in the MATLAB environment. This program generates 240 NARX models with different structures by performing the following:

1) Changing of the number of neurons from 1 to 20.

2) Usage of two activation functions logsig and tansig.

3) Usage of three training algorithms trainlm, trainscg and trainbr.

4) Training the network with series-parallel architecture and parallel architecture.

One of the problems that occur during NN training is network over-fitting. The early stopping and cross validation are the default methods for improving network generalization and reduce occurrence of over-fitting during the training operation. When the network begins to over-fit the data, the validation error begins to increase, and after a certain number of iterations, the training is stopped, and the weights and biases at the minimum validation error is fixed.

In this paper, the network training parameters are defined as: (i) the mean square error (mse) performance function which is minimized until it reaches a sufficiently low cut-off value of (0.01), (ii) the maximum number of training epochs (1000) which represents the number of times that all the training patterns are presented to the NN and (iii) the maximum number of validation increase (100) which represents the number of successive epochs in which the performance function fails to decrease. Training operation was repeated three times for the same neural network with the same input data set to increase the accuracy of the network. The $TR1_{exp}$ dataset is partitioned into 80% used for training the network and 20% used for cross validation. After finishing the network training operation, the $TS1_{exp}$ dataset is used for testing the network and evaluating its generalization performance. (*RMSE*) was used for the evaluation of the network performance in the training and testing operation. It was calculated for the whole set of data of each output parameter from the NN, and defined according to

Eqn. (5),

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\frac{y_m - y}{y_{max}})^2} \tag{5}$$

where, y_m is the actual output and y is the predicted output. The results of each computation cycle were recorded in a matrix form which includes the network structure, the root mean square error (*RMSE*) for training process, (*RMSE*) for testing process and training time. Next, the best NN was selected based on the minimum value of (*RMSE*) during testing operation. Table 4 summarizes the best MISO NARX models for each output parameters of the engine.

Table 4: The best MISO NARX models configuration

Output parameter	No of neurons	Training algorithm	S & PR / Pr	Activation function	Training RMSE	Testing RMSE
N_H	2	trainbr	S & PR	logsig	0	0.0022
N_I	3	trainscg	S & PR	logsig	0.0002	0.0018
N_L	2	trainlm	Pr	logsig	0.0006	0.0007
PW	15	trainscg	S & PR	logsig	0.0004	0.0107
TGT	11	trainscg	S & PR	logsig	0.0002	0.0011
T_{30}	15	trainlm	S & PR	logsig	0.0002	0.0076
P_{30}	5	trainscg	S & PR	logsig	0.0041	0.005

3.4 ENSEMBLE GENERATION

In this subsection, a homogeneous ensemble for each output parameter of the engine is generated based upon the best selected structure of the MISO NARX model from the last subsection and diversity among them is ensured by altering the training datasets which represent different operation conditions. Therefore, the ensemble for each output parameter consists of eight MISO NARX models with the same structure. Each model is retrained individually using different training dataset. Which represent certain operation condition. In this work, eight operation condition datasets $[TR1 - TR8]$ were generated to represent the ADGTE operation space. The retraining operation is performed in the same way as mentioned before.

3.5 ENSEMBLE INTEGRATION

Now that we have generated the ensemble for each engine output parameter, we move to the next step. How to combine the identifications that were made for each model in the ensemble and constructing the final output. Three approaches are used to handle the ensemble integration. Firstly, the basic ensemble method (BEM) defined by Eqn. (6) below , is a simple approach to aggregating network outputs by average them together. Secondly, the

median method, which is less affected by outliers and skewed data than the mean one. An outlier is an extreme value that differs greatly from other values.

$$f_{BEM} = \frac{1}{K}\sum_{i=1}^{K} f_i(x) \tag{6}$$

Thirdly, a dynamic weighting method (DWM) is considered. Note that, the previous two methods are considered as a constant weighting methods, while, DWM is considered as a non-constant weighting method. The weights are adjusted dynamically to be proportional to the performance of ensemble members (MISO NARX models), a greater weight will be assigned to the ensemble member with better performance. Finally, the proposed HDWM is performed as follows:

1) Calculation of the performance of each ensemble member as described in Eqn. (7),

$$e_i = (\frac{y_m - y_i}{y_{max}})^2 \tag{7}$$

2) Calculation of the median value of the models' errors

$$MED = median(e_1 e_2 \cdots e_K) \tag{8}$$

3) The weight of each model f_i is calculated according to its error as described in Eqn. (9), which calculate the weights in such a way, the model i with error e_i around the median value MED receives a weight close to 1, while models with e_i lower than MED have their weights exponentially increased, and models with e_i larger than MED have their weights exponentially decreased.

$$w_i = \exp\,(-\frac{e_i - MED}{MED}) \tag{9}$$

4) The ensemble output f_{en} is obtained as,

$$f_{en} = \frac{\sum_{i=1}^{K}[w_i(x) * f_i(x)]}{\sum_{i=1}^{K} w_i(x)} \tag{10}$$

5) Calculation of the error of the ensemble output with respect to the real output and comparison of this with the minimum error from the all ensemble members. If $f_{en} < min(e_1 \cdots e_K)$, then the final output will be the ensemble output , otherwise , the final output will be equal to the output from the ensemble member which has the minimum error value.

As we can see, the HDWM is a hybrid method which combines two integration approaches, the fusion approach and the selection approach. The former, combines the ensemble members outputs in order to obtain the final output by weighting each model output based on its performance. The latter, selects from the ensemble the most promising model only.

4 NEURAL NETWORK BASED NMPC DESIGN

4.1 The Generalized Predictive Control

Generalized Predictive Control (GPC) first introduced by Clarke et al. (1987*a*) is one of a class of MPC algorithms. This technique is popular not only in industry, but also at universities. A model is the core for any type of model-based control structures. The model used in GPC design is the controlled autoregressive and integrated moving average (CARIMA) given by

$$A(z^{-1})y(t) = B(z^{-1})u(t-1) + C(z^{-1})\frac{e(t)}{\Delta} \tag{11}$$

where u(t) is the input , y(t) is the plant output, and e(t) is the white noise. $\Delta = 1 - z^{-1}$ is the difference operator. $A(z^{-1})$, $B(z^{-1})$ and $C(z^{-1})$ are the polynomials in the backward-shift operator z^{-1} with the orders of n_y, n_u and n_k respectively.

The GPC strategy is based on applying a control sequence that minimizes a quadratic cost function measuring the control effort and the distance between the predicted process output and desired outputs over the prediction horizon, i.e.

$$\begin{aligned} J\left(N_1, N_2, N_u\right) = & \sum_{j=N_1}^{N_2}\left[\hat{y}(t+j|t) - w(t+j)\right]^2 \\ & + \Lambda \sum_{j=1}^{N_u}\left[\Delta u\left(t+j-1\right)\right]^2 \\ subjected\ to\ \Delta u(t+j) = 0\ \ when\ \ & j > N_u \end{aligned} \tag{12}$$

where $\hat{y}$ is the predicted output from the system model, and w is the reference output. $u(t+j-1)$ is the sequence of future control action that is to be determined. N_1, N_2 are the minimum, maximum horizon, and N_u is the control horizon. Λ is a weighting factor penalizing changes in the control inputs. The tuning parameters of the GPC are N_1, N_2, N_u, and Q, which determine the stability and performance of the GPC controller. Notice that, $N_1 \geq 1$, $N_2 \geq N_1$, and $N_2 \geq N_u \geq 1$. In addition, some guidelines for selecting those parameters exit in Clarke & Mohtadi (1987), Clarke et al. (1987*b*).

According to Camacho & Alba (2013) the future output value of the system is given by the equation of the predictor as follows:

$$\hat{y}(t+j|t) = G_j(z^{-1})\Delta u(t+j-1) + f(t+j) \tag{13}$$

where $f(t+j)$ is the **free response** of the system if the input remains to be constant at the last computed value $u(t-1)$. While, $G_j(z^{-1})\Delta u(t+j-1)$ represents the **forced response** of the system. It relies upon future control actions yet to be calculated. The polynomial $G_j(z^{-1})$ contains the system step response coefficients, as shown in Eq. (14).

$$G_j(z^{-1}) = E_j(z^{-1})B(z^{-1}) = g_0 + g_1 z^{-1} + \cdots + g_{j-1}z^{-(j-1)} \tag{14}$$

To simplify the following derivation of the GPC control law, let $N_1 = 1$. Now consider the following set of j step ahead optimal predictions. Then, the predictor in the vector notation can be expressed as:

$$\hat{y} = G\Delta u + f \tag{15}$$

For SISO system, the matrix G is a lower triangular of dimension $[N_2 X N_u]$. In addition, the first column of the G matrix can be evaluated as the step response of the system when a step input is applied to the controlling variable. The cost function Eq. (12) can now be written in matrix form as

$$\begin{aligned} J =& (G\Delta u + f - w)^T (G\Delta u + f - w) + \Lambda\ \Delta u^T \Delta u \\ =& \frac{1}{2}\Delta u^T H \Delta u + b^T \Delta u + f_0 \end{aligned} \tag{16}$$

where the gradient b and Hessian H matrices are defined as

$$\begin{aligned} H &= 2(G^T G + \Lambda\ I) \\ b^T &= 2(f - w)^T G \\ f_0 &= (f - w)^T (f - w) \end{aligned} \tag{17}$$

For unconstrained case, the minimum of J can be obtained by making the gradient of J equal to zero, which yields to

$$\Delta u = -H^{-1} b = (G^T G + \Lambda\ I)^{-1} G^T (w - f) \tag{18}$$

As the GPC is a receding-horizon control strategy, only the first control increment in Δu is applied to the system and the whole algorithm is recomputed at time $t + 1$.

The MIMO version of the GPC is a direct extension of the SISO GPC described above. The matrix and vector elements are not scalars but vectors and matrices. If m-inputs and n-outputs is considered, then matrix $\boldsymbol{G}$ has dimension of $[n * N_2 X m * N_u]$, and it can be obtained as:

$$\boldsymbol{G} = \begin{bmatrix} G_{11} & G_{12} & \cdots & G_{1m} \\ G_{21} & G_{22} & \cdots & G_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ G_{n1} & G_{n2} & \cdots & G_{nm} \end{bmatrix} \tag{19}$$

where each matrix G_{ij} of dimension $[N_2 X N_u]$ contains the coefficients of the ith step response corresponding to the jth input. Notice that, the vectors $\hat{\boldsymbol{y}}$, $\boldsymbol{f}$, $\boldsymbol{w}$ have a dimension of $[n * N_2 X 1]$, and $\boldsymbol{\Delta u}$ vector gas a dimension of $[m * N_u X 1]$. The control weighting matrix $\boldsymbol{\Lambda}$ is with positive elements on its diagonal, i.e.

$$\boldsymbol{\Lambda} = diag(\Lambda_1, \Lambda_2, \cdots, \Lambda_m)$$

As can be seen, one of the advantages of MPC is that multi-variable processes can be handled in a simple way.

The advantages of GPC become obvious only when constraints are considered. The constraints acting on a process can originate from slew rate limits of the actuator, amplitude limits in the control signal, and limits on the output signals. For SISO system, these constraints can be described, respectively, by

$$\begin{aligned} \Delta u_{min} \leq \Delta u(t) \leq \Delta u_{max} \\ u_{min} \leq u(t) \leq u_{max} \\ y_{min} \leq y(t) \leq y_{max} \end{aligned} \tag{20}$$

where Δu_{min} and Δu_{max} are the lower and upper bounds on the future control increment; u_{min} and u_{max} are the lower and upper bounds on the manipulated input amplitude; and y_{min} and y_{max} are the lower and upper bounds on the process output amplitude.

To this end, we have to define the predictive control problem (Eq. (16)) as an optimization problem that considers the constraints present. Therefore, the key here is to formulate the constrained parameters utilizing the Δu parameter. In this way, these constraints (Eq. (20)) can be expressed as

$$\begin{aligned} L_u \Delta u_{min} &\leq I\ \Delta u \leq L_u \Delta u_{max} \\ d_{u_{min}} &\leq T_u\ \Delta u \leq d_{u_{max}} \\ L_y\ y_{min} &\leq G\ \Delta u + f \leq L_y\ y_{max} \end{aligned} \tag{21}$$

where L_u is an N_u vector, whose entries are ones; T_u is an $N_u X N_u$ lower triangular matrix whose entries are ones; I is an $N_u X N_u$ identity matrix; and L_y is an N_2 vector, whose entries are ones. $d_{u_{min}} = (u_{min} - u(t-1))L_u$, and $d_{u_{max}} = (u_{max} - u(t-1))L_u$.

Now, the input and output constraints can be merged in a single inequality on Δu as

$$M_C \Delta u \leq d_C \tag{22}$$

where

$$M_C = \begin{bmatrix} I \\ -I \\ T_u \\ -T_u \\ G \\ -G \end{bmatrix}, d_C = \begin{bmatrix} L_u\ \Delta u_{max} \\ -L_u\ \Delta u_{min} \\ d_{u_{max}} \\ -d_{u_{min}} \\ L_y\ y_{max} - f \\ -L_y\ y_{min} + f \end{bmatrix} \tag{23}$$

M_C is a matrix representing the constraints with its number of rows equivalent to the number of constraints and the number of columns equivalent to the dimension of the vector Δu.

For an m-input n-output system with constraints acting over a prediction horizon N_2 and control horizon N_u, the similar mathematical formula can be derived.

4.2 The Neural Network Generalized Predictive Controller (NNGPC)

An investigation of a novel approach to implement the NMPC based on neural network model is reported in this study. As can be seen, the design of the GPC demands the construction of a predictor (Eq. (15)). Hence, for the calculation of the predicted future output $\hat{y}$, only two characteristics of the system are needed: step (forced) and free responses. To obtain the step and the free process responses, which are needed in the generalized predictive control strategy, we iteratively use an ensemble of MISO-NARX models as a multi-step-ahead predictor.

The proposed control scheme in Fig. 7 composes of a non-linear NN model of the process to be controlled in the form of ensembles of multiple MISO-NARX models and the GPC algorithm block. This model works as a predictor which produces the free and forced responses, that are used as an input to the GPC algorithm block. The GPC algorithm block generates an output that is either utilized as an input to the plant or the predictor. The double pole double throw switch, S, is set to the plant when the GPC algorithm block has found the optimal control input, $u(k)$, that minimizes the cost function. Between samples, the switch is set to the predictor where the GPC algorithm block utilizes this predictor to calculate the next control input, $u(k+1)$, by making prediction

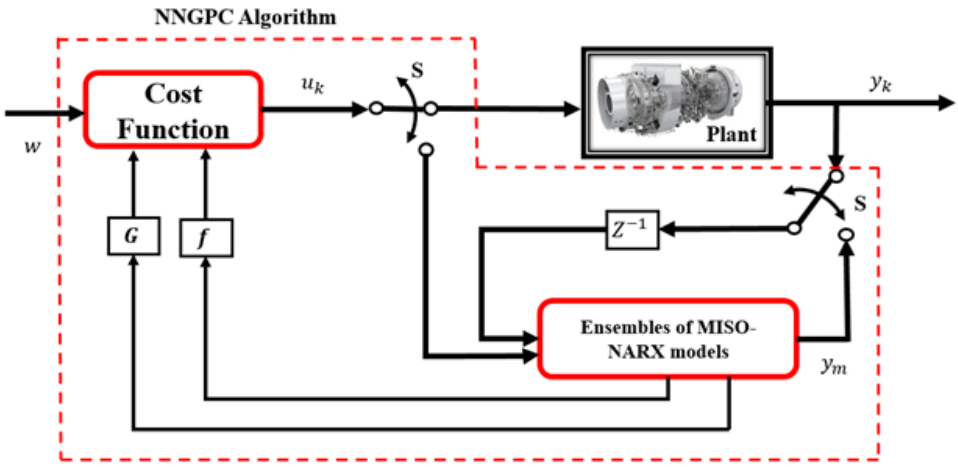

Figure 7: Block diagram of NMPC system.

of the response over the prediction horizon N_2. At the point when the cost function is minimized, the optimal control input is passed to the plant. This algorithm is outlined below.

An approximation of the predicted future output is given by:

$$\hat{y} = G\Delta u + f \tag{24}$$

In this expression at each sampling time the vectors $G(t)$ and $f(t)$ are reconstructed. The free response f depends only on the past inputs and outputs. Therefore, to get the free response the prescribed predictor is given a zero increment vector Δu i.e.

$$u(t) = u(t+1) = \cdots = u(t+N_2-1) = u(t-1) \tag{25}$$

Hence, the predictor output will be the system free response.

$$\begin{aligned} y_{free}(t+i|t) = F(&y(t+i-1), \cdots, y(t+i-n_y), \cdots \\ &, u(t+i-1), \cdots, u(t+i-n_u)) \\ with: u(k) = &\begin{cases} u(t-1), \text{if } k > t-1 \\ u(k), \qquad\qquad \text{otherwise} \end{cases} \end{aligned} \tag{26}$$

where $F(.)$ represents the NN response.

The estimation of the step response coefficients to construct G matrix is obtained as follows:

$$\begin{aligned} g_{k-1} &= \frac{y_{step}(t+k|t) - y_{free}(t+k|t)}{\delta u(t)} \\ For \;\; k &= 1, \cdots, N_2 \end{aligned} \tag{27}$$

where δu represents the step size, and y_{step} represents the predictor step response which can be obtained by:

$$\begin{aligned} y_{step}(t+k|t) = f(&y(t+k-1), \cdots, y(t+k-n_y), \cdots \\ &, u(t+k-1), \cdots, u(t+k-n_u)) \\ with: u(k) = &\begin{cases} u(k), \text{if } k \leq t-1 \\ u(t-1) + \delta u(t), \quad \text{if } k > t-1 \end{cases} \end{aligned} \tag{28}$$

The main problem to get a precise estimate of the step response around operating point is to choose a proper value for the amplitude and the sign of the step $\delta u(t)$, in light of the fact that the step response of a non-linear system is determined by the operating point, the magnitude and the sign of the step signal Oviedo et al. (2006). Therefore, the $\delta u(t)$ should be chosen near to the predicted $\Delta u(t)$. Since this value is just known after the optimization, a good decision is to choose $\delta u(t)$ equal to $\Delta u(t-1)$ acquired in the previous optimization step $(t-1)$. However, if $\Delta u(t) = 0$ as the system reaches the steady state, Eq. (27) will be badly conditioned. Consequently, a δu_{min} vector ought to be characterized with the minimum value of δu, so that the estimation of step response coefficients is reliable.

Once f and G have been obtained from the response of the predictor (ensemble of MISO-NARX models), the gradient **b** and Hessian **H** of the GPC quadratic cost function can be calculated based on Eq.(17). Therefore, by using this novel approach, the optimization problem can be solved as a linear optimization problem instead of a nonconvex and non-linear programming problem, that will improve the computation time and reliability of the solution. In this paper, Hildreth's quadratic programming algorithm is utilized which offers straightforwardness and reliability in real time implementation Wang (2009). Besides, Hildreth's algorithm might be helpful to execute on non-PC platforms like programmable logic controllers or embedded machine which do not support linear algebra libraries.

4.3 NNGPC Design for SGT-A65 Engine

The NNGPC is implemented in MATLAB®/SIMULINK environment. As shown in Fig. 8, the NNGPC controller substitutes the seven control loops in the min-max controller by one NNGPC multi-variable controller. The first objective of the NNGPC is to maintain the low pressure spool speed N_L at a certain set point (3600 rpm) as the generator load changes. The second objective considered is to ensure that N_H, N_I, TGT, and P_{30} stay below its maximum limits so as not damage the gas turbine. These two objectives are achieved by manipulating three controlling variables: WF, $IPVSV$, and $LPBOV$.

In addition to these two objectives, input constraints on amplitude and slew rate of the controlling variables (WF, $IPVSV$, and $LPBOV$) are introduced, to ensure safe acceleration and deceleration of the engine. In practice, input and output constraints are not necessarily fixed values. It is possible that constraints change with a certain parameter variable. Acceleration and deceleration are limited by maximum and minimum allowable fuel flow request WF_{max} and WF_{min} respectively, which are changed based on engine performance. Beside that, the TGT_{max} is scheduled according to ambient temperature and relative humidity. This value is used as the upper limit of the TGT inside NNGPC controller.

In practical situations such as our case, the controlled variables may not be subject to constraints, and the constrained outputs may not need to be controlled. Based on that, three actuators are used to control a single

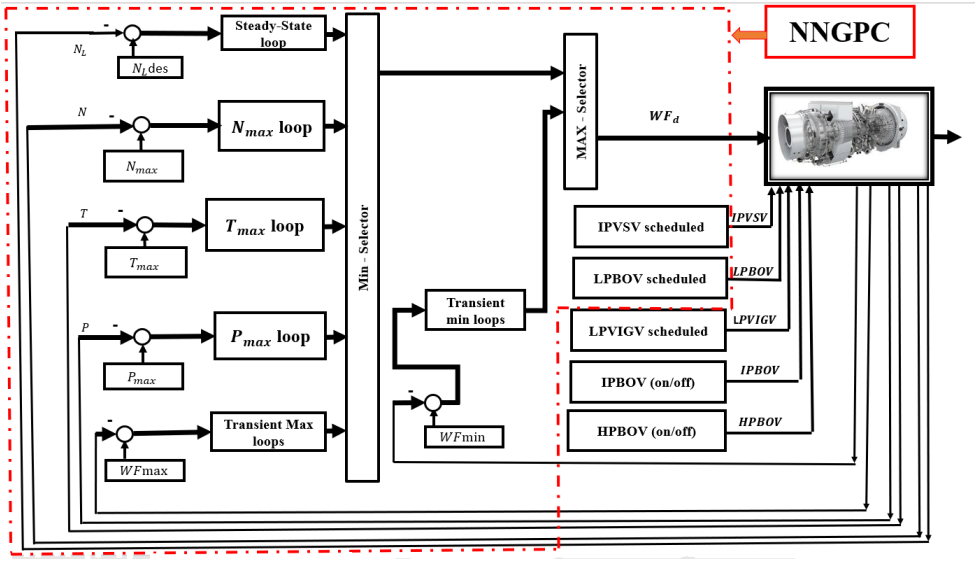

Figure 8: The NNGPC for SGT-A65.

controlled output, namely N_L. Four other output variables, namely N_H, N_I, TGT, and P_{30} are considered as constrained outputs. This requires that different sets of G and f matrices must be calculated, according to whether an output is controlled or constrained.

The proposed NNGPC design comprises of three tuning parameters that should be appropriately tuned to accomplish good controller performance. The tuning parameters that influence the closed loop response of the system include the prediction horizon N_2, the control horizon N_u, and the weighting factor Λ. In this study, the NNGPC tuning parameters are obtained by trial and error.

5 RESULTS

5.1 Neural Model Validation

In this subsection, validation of the generated ensembles for all engine output parameters is performed by using the testing datasets ($TS1 - TS8$) in the MATLAB/SIMULINK environment. Note that for space reasons only figures from N_H, PW and N_L prediction are presented. Firstly, Fig. 9 shows that the outputs from each model in the N_H ensemble are different from each other, which explains the ensemble diversity. In this work, the input space is partitioned into eight subspaces, each one represents certain operation conditions, and each model in the ensemble is then assigned to one of these sub-spaces. In another word, we used a mixture of experts to develop a homogeneous ensemble which can represent the engine at different operation conditions.

To show the advantages of using an ensemble in the prediction of engine performance instead of using an individual neural model, we generated a single MISO NARX for each engine output parameters, and trained them with the same approach as mentioned above. Indeed, concatenated data from different operation conditions is used for training operation. Fig. 10 to Fig. 13 show the comparison between N_H and PW estimated by the

ensemble of MISO NARX models and the single MISO NARX model. This demonstrates that ensembles of diverse models aggregated with HDWM method can provide higher accuracy and higher robustness in real time than the single MISO NARX neural model approach. A summary of comparison results is shown in Table 5. One can observe that the ensemble model demonstrates a significantly better performance in identification of the gas turbine engine dynamics than the individual neural model, as it results in an improvement in accuracy of nearly 90%, compared with the single neural model.

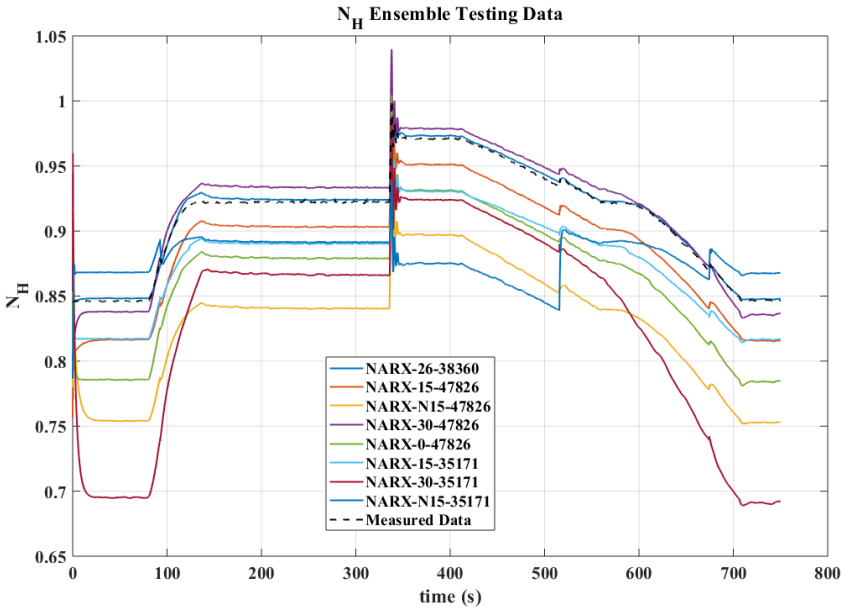

Figure 9: Ensemble models prediction-NH.

On the other hand, in order to verify the performance of the proposed ensemble integration method (HDWM), a comparative study was performed between four integration algorithms to measure their impact on the ensemble performance with respect to $TS1_{exp}$ data set. A summary of results of the four integration algorithms presented in Table 6. Indeed, Fig. 14 to Fig. 16 show estimation of N_H, PW and N_L by ensemble of N_H, PW and N_L respectively with different integration algorithms and tested with $TS1_{exp}$ data set. As we can see, the proposed HDWM has demonstrated superior performance over the other integration methods.

A summary of the ensemble system performance with HDWM integration method for identification of each of the seven gas turbine engine outputs is given in Table 7. As expected, the usage of homogeneous ensemble with HDWM integration method in the prediction of the ADGTE output parameters gave a high prediction accuracy at different operation conditions.

5.2 NNGPC Validation

To assess the performance of the NNGPC developed in this study, we have compared the response of the NNGPC controller to that of the existing min-max controller due to the same load disturbance. For this comparison, the experimental dataset $TS3_{exp}$ has been used. The simulation results of this test are presented in Fig. 17 through

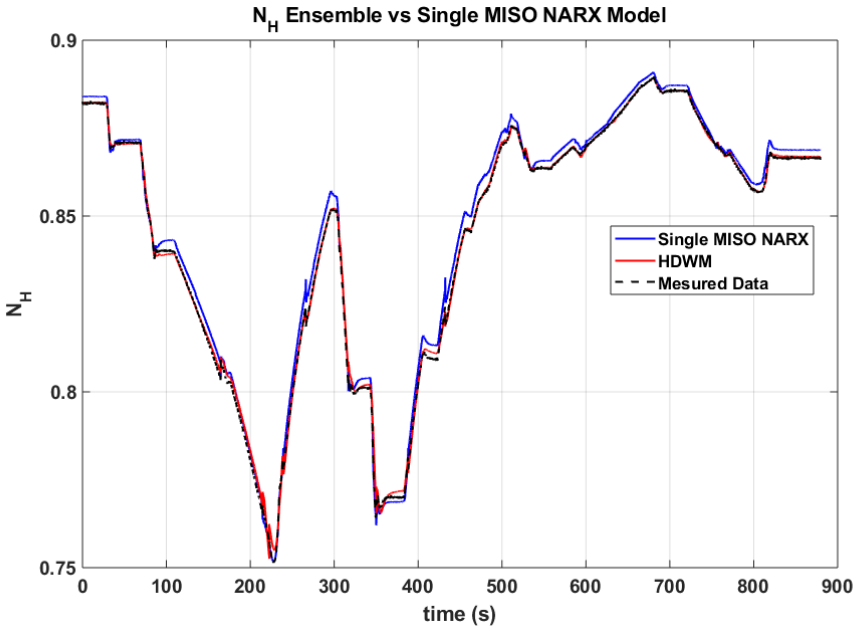

Figure 10: $TS5_{sim}$ testing results.

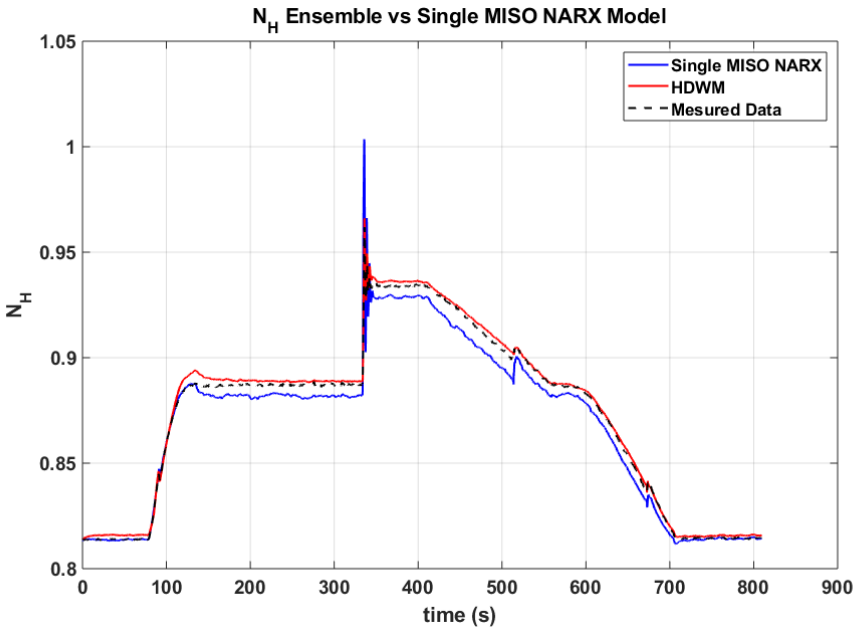

Figure 11: $TS1_{exp}$ testing results-NH.

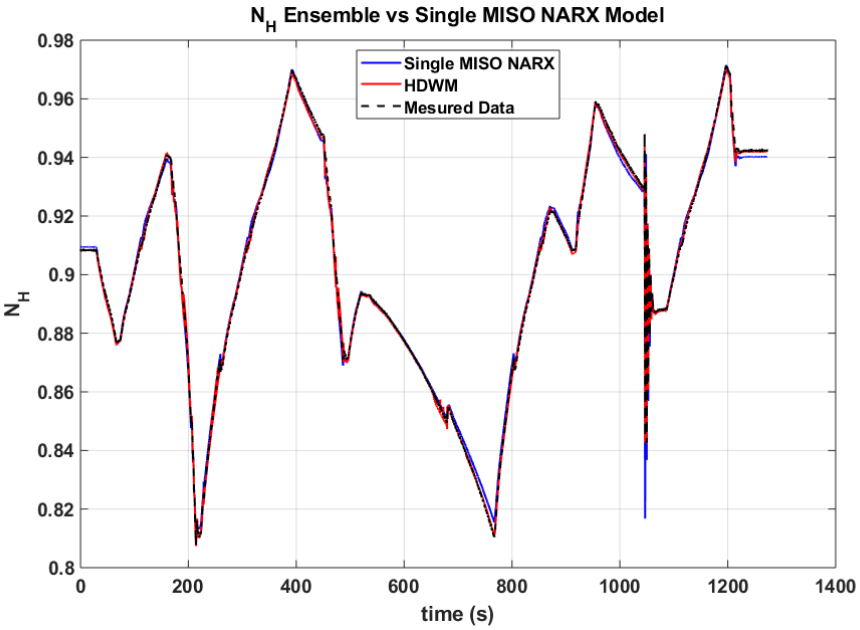

Figure 12: $TS7_{sim}$ testing results-NH.

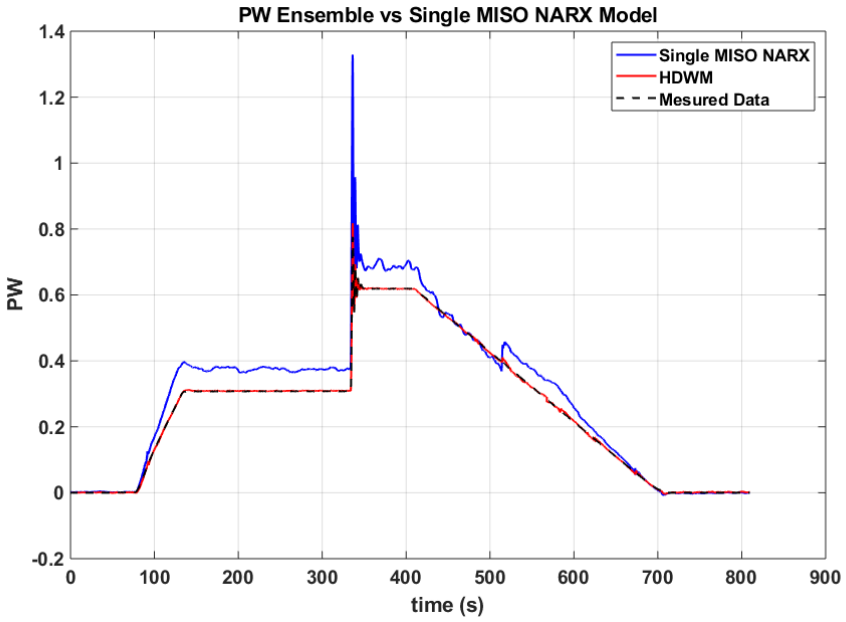

Figure 13: $TS1_{exp}$ testing results-PW.

Table 5: Regression performance [$RMSE$] of single MISO NARX model and ensemble of eight MISO NARX models

Output parameter	Data set	Ensemble	Single model
N_H	$TS1_{exp}$	0.00151	0.00500
	$TS5_{sim}$	0.00005	0.00280
	$TS7_{sim}$	0.00063	0.00200
N_I	$TS1_{exp}$	0.00043	0.00490
	$TS5_{sim}$	0.00086	0.01070
	$TS7_{sim}$	0.00005	0.00380
N_L	$TS1_{exp}$	0.00004	0.02930
	$TS5_{sim}$	0.00003	0.01350
	$TS7_{sim}$	0.00006	0.01030
PW	$TS1_{exp}$	0.00180	0.04930
	$TS5_{sim}$	0.00300	0.01440
	$TS7_{sim}$	0.00903	0.02840
TGT	$TS1_{exp}$	0.00351	0.02020
	$TS5_{sim}$	0.00131	0.02480
	$TS7_{sim}$	0.00054	0.03770
T_{30}	$TS1_{exp}$	0.00389	0.04380
	$TS5_{sim}$	0.00131	0.02480
	$TS7_{sim}$	0.00014	0.07690
P_{30}	$TS1_{exp}$	0.00004	0.07920
	$TS5_{sim}$	0.005191	0.04000
	$TS7_{sim}$	0.00173	0.02200

Table 6: Regression performance [$RMSE$] of ensemble of MISO NARX models with different integration methods.

Output parameter	HDWM	DWM	BEM	Median
N_H	0.00151	0.00005	0.04283	0.02877
N_I	0.00043	0.01319	0.03669	0.03482
N_L	0.00004	0.00033	0.00534	0.00054
PW	0.00180	0.02960	0.44110	0.07860
TGT	0.00351	0.00926	0.02067	0.03183
T_{30}	0.00389	0.02237	0.05820	0.05870
P_{30}	0.00004	0.00262	0.00240	0.00931

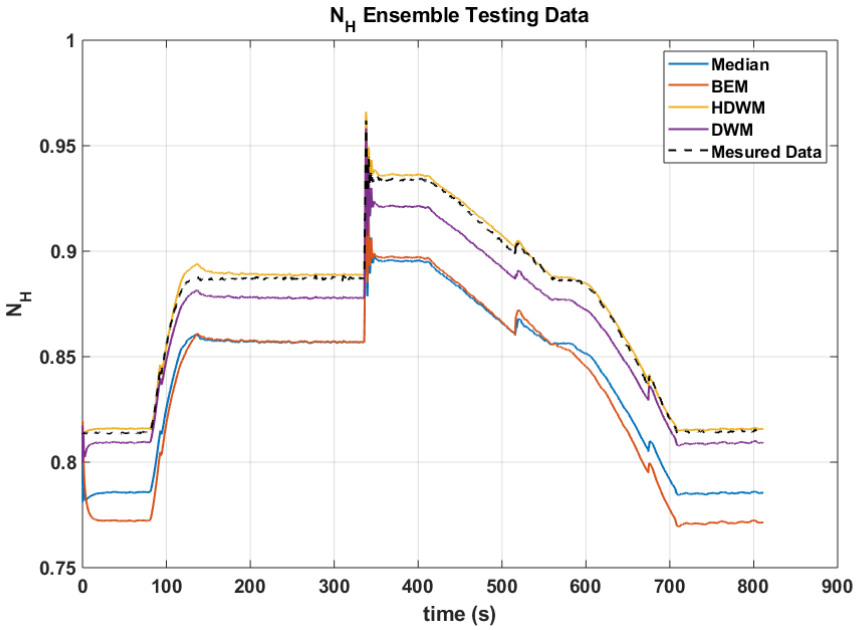

Figure 14: N_H prediction with ensemble by $TS1_{exp}$ data set.

Table 7: Regression performance [$RMSE$] of ADGTE ensembles at different operation conditions.

Data set	N_H	N_I	N_L	PW	TGT	T_{30}	P_{30}
$TS2_{sim}$	0.00015	0.00061	0.00008	0.00045	0.00083	0.00021	0.00228
$TS3_{sim}$	0.00063	0.00011	0.00007	0.00046	0.01713	0.00768	0.00049
$TS4_{sim}$	0.00052	0.00005	0.00026	0.00200	0.00197	0.00012	0.00208
$TS6_{sim}$	0.00081	0.00284	0.00103	0.08739	0.00152	0.00121	0.00649
$TS8_{sim}$	0.00107	0.00082	0.00034	0.00104	0.00390	0.00132	0.00039

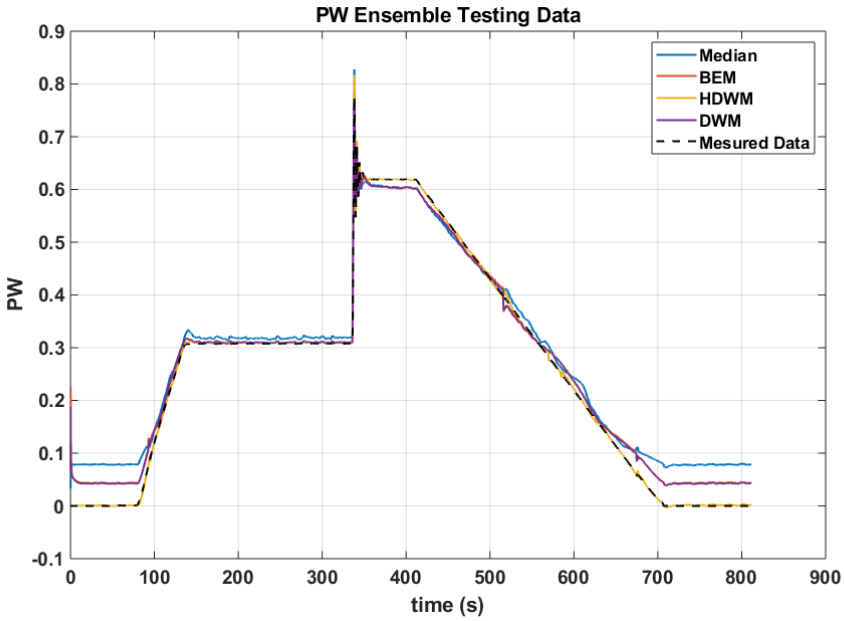

Figure 15: PW prediction with ensemble by $TS1_{exp}$ data set.

Fig. 23 with the summary of comparison results given in Table 8. As can be seen, the NNGPC has demonstrated output responses with less oscillatory behavior and smoother control actions to the sudden variation in the electric load than those observed in the existing min-max controller. In addition, both controllers have the ability to maintain all constrained parameters below the predetermined maximum limits. However, the time constant of min-max controller response is lower than that of the NNGPC controller, which resulted in faster response by using min-max controller.

In Fig. 18, the N_L achieves 1.087 (overshoot is 8.7 %) with the NNGPC controller, which is within % 10 of its nominal value, in accordance with the load rejection test criterion. In addition, the NNGPC controller brought back N_L to its set point value in the course of 4.2 s. However, the min-max controller could bring the response to the set point value in about 12.79 s with overshoot of 8.4 %.

As shown in Fig. 21 through Fig. 23, the control movements resulted from the NNGPC controller show smoother variations in WF, $IPVSV$, and $LPBOV$ parameters than those resulting from the min-max controller. As a consequence of that, the NNGPC controller requires less control effort than the min-max controller to achieve the desired objectives. The minimization of controller effort has the capability to decrease the mechanical wear of the actuators, which could lead in turn to an increase of the functional safety, life time, and economics of the controlled process.

Moreover, Fig. 21 shows that the fuel flow rate WF calculated by the min-max controller exceeds the minimum fuel limit during the load rejection test. In addition, The minimum fuel flow limit violation by the min-max controller may occur as a result of the strong non-linearity of the engine, which can not always be handled adequately by the classical controller especially during the fast load change. In fact, This result coincides with the opinion found in recent studies, which have shown that there is no guarantee for min-max algorithm with linear compensator to protect engine limits during fast transient stateImani & Montazeri-Gh (2017), Montazeri-

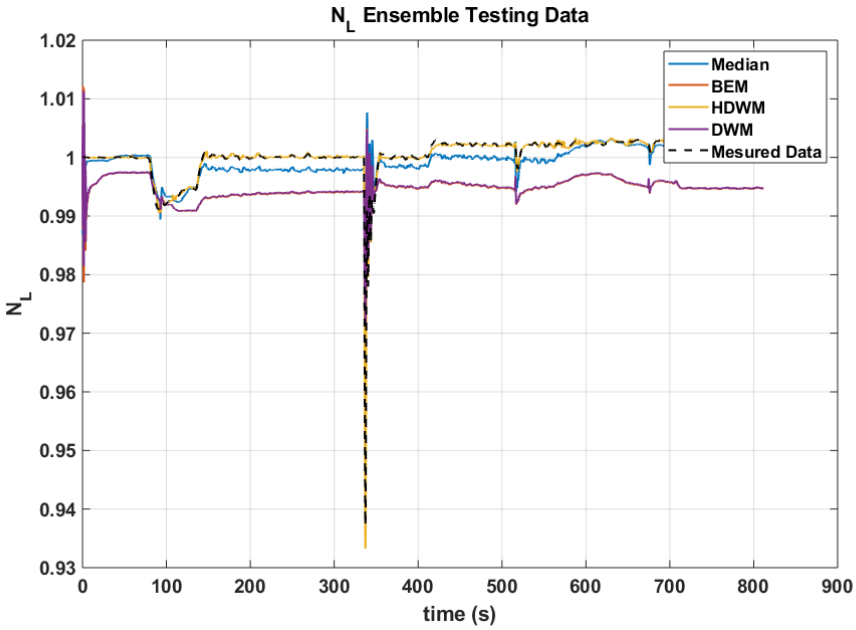

Figure 16: N_L prediction with ensemble by $TS1_{exp}$ data set.

Table 8: The controller performance comparison under load rejection test.

Output parameter	Controller	TS (s)	OS (%)	Time constant (s)
PW	NNGPC	3.22	2.6	1.16
	min-max	3.01	11.2	0.01
N_H	NNGPC	4.99	n/a	2.7
	min-max	3.1	6.17	0.19
N_I	NNGPC	3.76	1.2	2.68
	min-max	3.12	5.94	0.17
TGT	NNGPC	4.66	1.75	3.18
	min-max	9.24	n/a	1.62
P_{30}	NNGPC	1.5	2.97	0.31
	min-max	4.1	23.56	0.23
T_{30}	NNGPC	6.1	3.59	2.25
	min-max	12.91	n/a	3.1

Gh & Rasti (2019). However, the NNGPC controller, with the current design constraint, has the potential to keep WF within its limits.

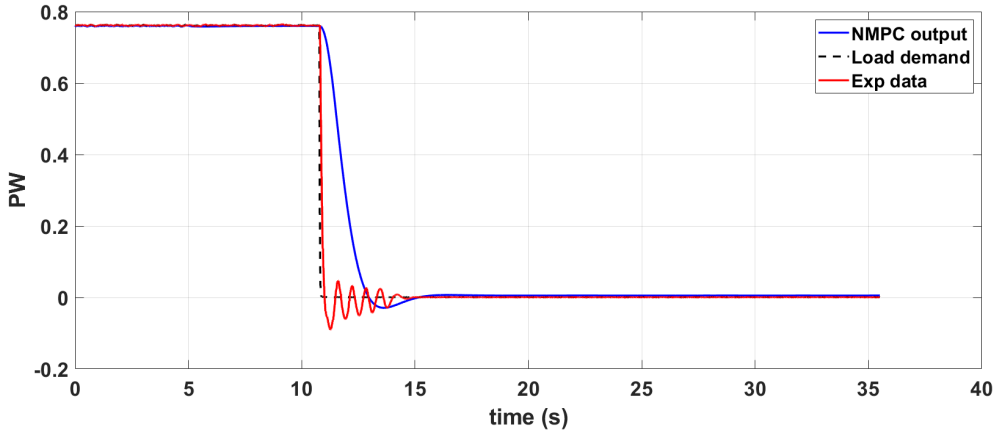

Figure 17: Comparison between NNGPC controller and min-max controller performance during load disturbance - PW.

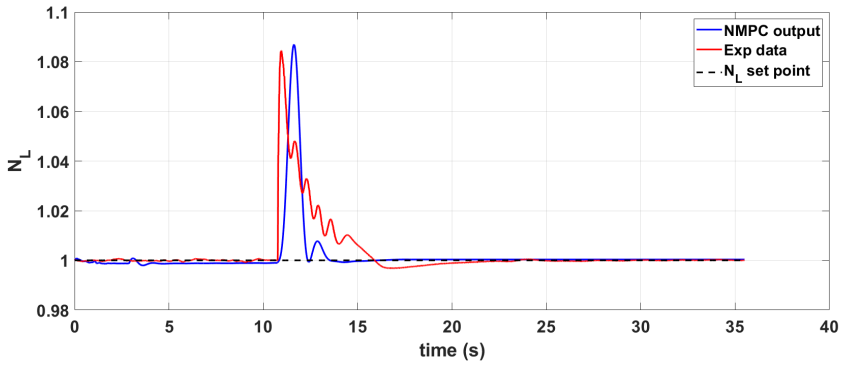

Figure 18: Comparison between NNGPC controller and min-max controller performance during load disturbance - N_L.

Finally, the computational efforts required for the NNGPC to execute the calculations of the controller during this test is 5.153 s, knowing that, the test simulation time is 35.5 s. Therefore, the computation time required to solve an optimization problem was sufficiently faster than the sampling rate (T_s = 0.01 s) by applying Hildreth's quadratic programming algorithm. This efficient algorithm would allow NNGPC to be implemented via real-time optimization for gas turbine power plants in a fast and robust manner.

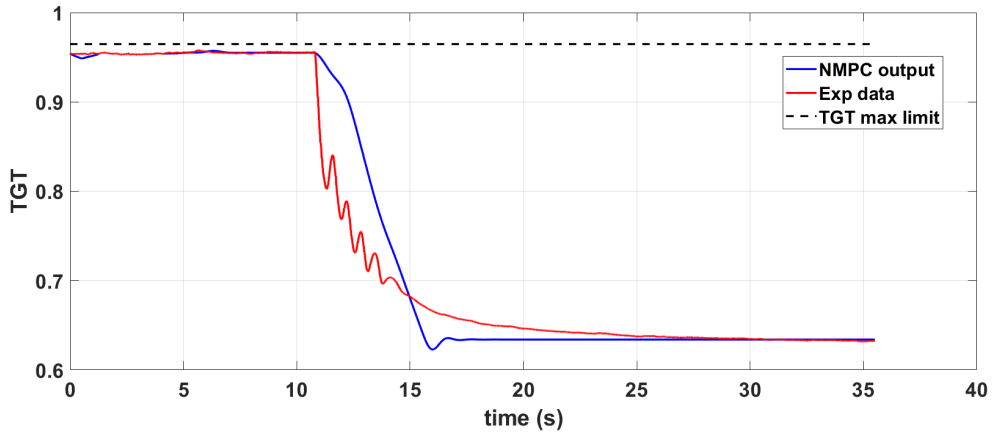

Figure 19: Comparison between NNGPC controller and min-max controller performance during load disturbance - TGT.

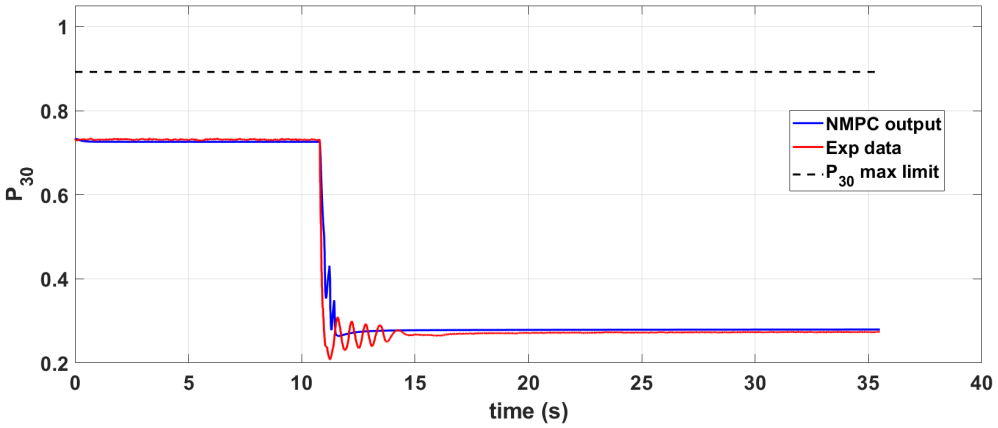

Figure 20: Comparison between NNGPC controller and min-max controller performance during load disturbance - P_{30}.

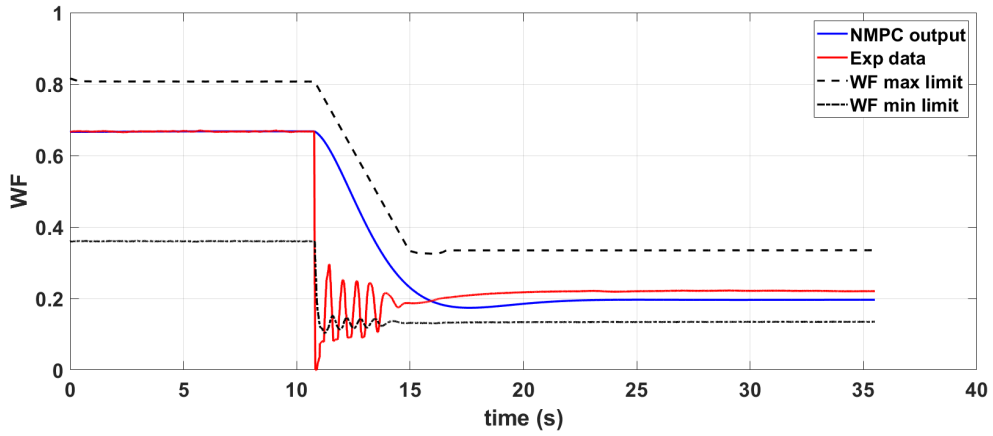

Figure 21: Comparison between NNGPC controller and min-max controller performance during load disturbance - WF.

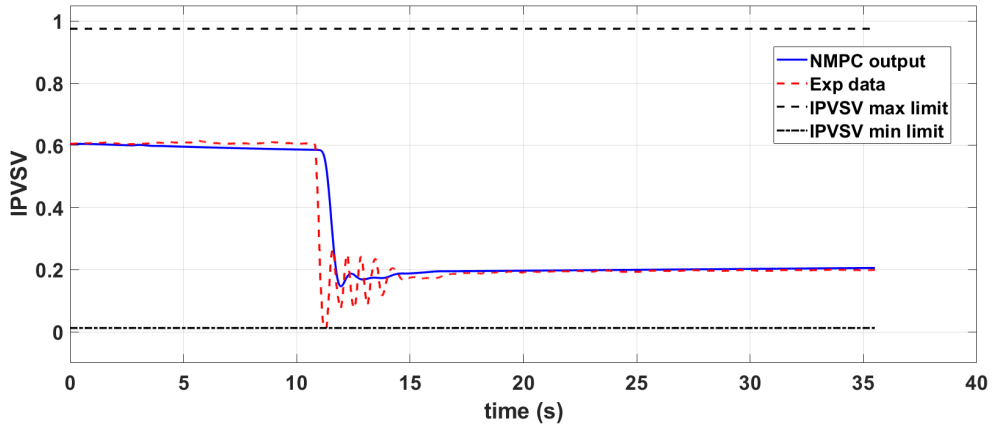

Figure 22: Comparison between NNGPC controller and min-max controller performance during load disturbance - $IPVSV$.

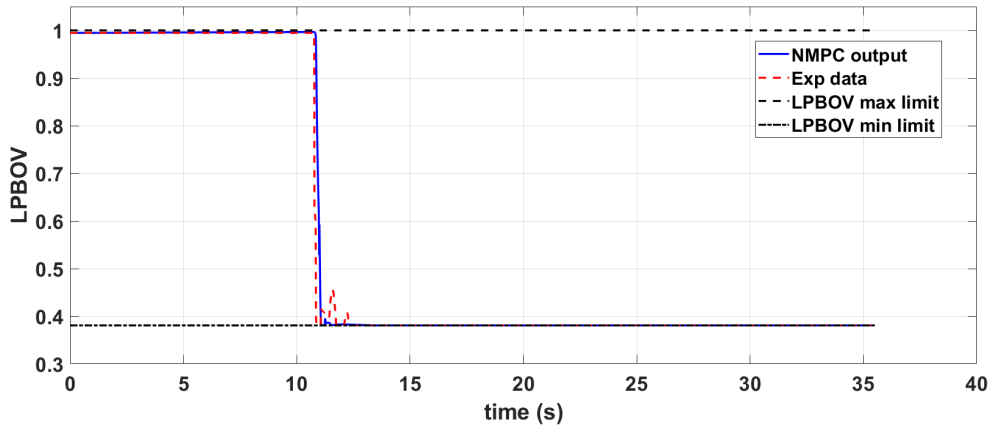

Figure 23: Comparison between NNGPC controller and min-max controller performance during load disturbance - $LPBOV$.

6 CONCLUSIONS

This paper focuses on solving important challenges in the area of gas turbine engine's controller design to ensure the safe operation of the engine and at the same time get the maximum performance. The system identification and advanced control algorithms based on NN methodology were developed to control an ADGTE used in power generation application under loading and unloading condition. The main results can be summarized as follows;

1) A novel methodology for the development of real time data driven based model of ADGTE was presented in 3. An ensemble of multiple MISO-NARX neural models was introduced to predict the ADGTE output parameters in real time. Inspired by the way biological neural networks process information and by their structure which changes depending on their function, multiple-input single-output (MISO) NARX models with different configurations were used to represent each of the ADGTE output parameters with the same input parameters.

2) Estimation of the NN model order by generating different ARX models and estimation of the input/output delay, before generation of NN model, are very important steps. These steps save more iterations required to find the best structure of the NN and consequently saving more time required for NN model generation. In addition, data cleaning and resampling step significantly reduced training time. The lower sampling rate reduces the number of data points, which reduces the computation time during training operation and reduces data collinearity.

3) Usage of a single neural network to represent each of the system output parameters may not be able to provide an accurate prediction for unseen data and as a consequence, provides poor generalization. To overcome this problem, an ensemble of MISO NARX models was used to represent each output parameter. The major challenge of the ensemble generation is to decide how to combine results produced by the ensemble's components. A novel hybrid dynamic weighting method (HDWM) was proposed to perform this task.

4) This paper was proposing a novel method to estimate the free and forced responses of the GPC based

on the NN model of the plant each sampling time. It reduces the NNGPC optimization problem to a linear optimization problem at each sampling step. Therefore, the optimization problem can be solved using quadratic programming instead of a nonconvex and non-linear programming problem, that will improve the computation time and reliability of the solution. In addition, the Hildreth's quadratic programming algorithm was used to solve the quadratic optimization problem within the NNGPC controller, which offers simplicity and reliability in real-time implementation. Furthermore, Hildreth's method may be useful to implement on non-PC platforms like programmable logic controllers or embedded machine which do not support linear algebra libraries.

In addition, a larger prediction horizon can make a faster response and better control quality. However, it will also greatly increase the calculation and affect real time performance of NNGPC controller. So that, as an indication of forthcoming research, it is intended to use a more systematic approach to perform NMPC tuning and find the best tuning parameters based on the desired performance characteristics. NMPC tuning is usually performed ad-hoc based on some experience. Most of the time, several simulation runs are performed to check if the chosen tuning parameters are suitable. Automated tuning procedures using genetic algorithm can be applied to reduce the manual operations.

As the only available experimental data from the real engine testing was collected at a specific operation conditions, future work will involve looking at the NNGPC controller performance over a different operation conditions and different transient scenarios and determining what modifications are required to obtain acceptable robust performance. Also, additional design and analysis will need to be performed taking into account compressor working line, surge margin, fuel valve micro motion stability criteria, and over speed analysis of the system. Once the design phase is completed the results will have to then be validated by implementing it on Programmable Logic Controller (PLC) with the focus of eventually being able to test it on a real life engine. Thus, even though this novel NNGPC controller presents promising first iteration results, there is great scope to improve and tune this controller.

7 PERMISSION FOR USE

NOMENCLATURE

ADGTE Aero-derivative gas turbine engine.

ARX Autoregressive with exogenous inputs.

LPC Low pressure compressor.

IPC Intermediate pressure compressor.

HPC High pressure compressor.

HPT High pressure turbines

IPT intermediate pressure turbines

LPT Low pressure turbines

PT Power turbine

HPBOV Number of opened HPC bleed off valves .

IPVSV Variable stator vanes position % .

IPBOV Number of opened IPC bleed off valves.

LPBOV Low pressure bleed-off valves position %.

LHV Fuel heating value (kJ/kg).

N_L The LPC speed.

N_I The IPC speed.

N_H The HPC speed.

P_{amb} Ambient pressure.

P_{30} The HPC exit pressure.

PW The engine shaft power.

RMSE Root mean square error.

T_{amb} Ambient temperature (C).

P_{amb} Ambient temperature (kPa).

T_{30} The HPC exit temperature .

TGT The intermediate turbine exit temperature.

trainbr Bayesian regularization training algorithm.

trainlm Levenberg-Marquardt training algorithm.

trainscg Scaled conjugate gradient training algorithm.

VIGV Variable inlet guide vanes position %.

WF Fuel flow rate (kg/s).

η_{th} Thermal efficiency.

8 REFERENCES

Alves, V. A. O., de Godoy, R. J. C. & Garcia, C. (2013), Optimal time delay estimation for system identification, *in* '2013 American Control Conference', IEEE, pp. 95–100.

Aly, A. & Atia, I. (2012), Neural modeling and predictive control of a small turbojet engine (sr-30), *in* '10th International Energy Conversion Engineering Conference', p. 4242.

Amozegar, M. & Khorasani, K. (2016), 'An ensemble of dynamic neural network identifiers for fault detection and isolation of gas turbine engines', *Neural Networks* **76**, 106–121.

Asgari, H., Chen, X. & Sainudiin, R. (2013), 'Modelling and simulation of gas turbines', *International Journal of Modelling, Identification and Control* **20**(3), 253–270.

Bahlawan, H., Morini, M., Pinelli, M., Spina, P. R. & Venturini, M. (2017), Development of reliable narx models of gas turbine cold, warm and hot start-up, *in* 'ASME Turbo Expo 2017: Turbomachinery Technical Conference and Exposition'.

Beale, M. H., Hagan, M. T. & Demuth, H. B. (2015), Neural network toolbox™ user's guide.

Brown, G., Wyatt, J., Harris, R. & Yao, X. (2005), 'Diversity creation methods: a survey and categorisation', *Information Fusion* **6**(1), 5–20.

Camacho, E. F. & Alba, C. B. (2013), *Model predictive control*, Springer Science & Business Media.

CASTRO, E. (2018), 'Diagram of how the brain works elegant gemütlich diagram brain parts and functions bilder menschliche', dreamdiving.
URL: *http://www.dreamdiving-resort.com/diagram-of-how-the-brain-works*

Clarke, D. & Mohtadi, C. (1987), 'Properties of generalized predictive control', *IFAC Proceedings Volumes* **20**(5), 65–76.

Clarke, D. W., Mohtadi, C. & Tuffs, P. (1987*a*), 'Generalized predictive control—part i. the basic algorithm', *Automatica* **23**(2), 137–148.

Clarke, D. W., Mohtadi, C. & Tuffs, P. (1987*b*), 'Generalized predictive control—part ii extensions and interpretations', *Automatica* **23**(2), 149–160.

Cybenko, G. (1989), 'Approximation by superpositions of a sigmoidal function', *Mathematics of control, signals and systems* **2**(4), 303–314.

de Sousa, J. F., Moreira, J. M., Soares, C. M. & Jorge, A. (2012), 'Ensemble approaches for regression: A survey'.

H.I.H. Saravanamuttoo, G.F.C. Rogers, P. S. H. C. A. (2017), *Gas Turbine Theory*, 7 edn, Pearson.
URL: *http://gen.lib.rus.ec/book/index.php?md5=e522f34d4e94f04313d3074c1e7ef784*

Ibrahem, I. A., Akhrif., O., Moustapha., H. & Staniszewski., M. (2019), Neural networks modelling of aero-derivative gas turbine engine: A comparison study, *in* 'Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics - Volume 1: ICINCO,', INSTICC, SciTePress, pp. 738–745.

Imani, A. & Montazeri-Gh, M. (2017), 'Improvement of min–max limit protection in aircraft engine control: An lmi approach', *Aerospace Science and Technology* **68**, 214–222.

Jafari, S. & Nikolaidis, T. (2018), 'Turbojet engine industrial min–max controller performance improvement using fuzzy norms', *Electronics* **7**(11), 314.

Jaw, L. & Mattingly, J. (2009), 'Aircraft engine controls'.

Khan, K. (2018), 'Surprising discovery of genetic mosaic in brain', About Islam.
URL: *http://aboutislam.net/science/health/surprising-discovery-genetic-mosaic-brain/*

Kim, J. S., Powell, K. M. & Edgar, T. F. (2013), Nonlinear model predictive control for a heavy-duty gas turbine power plant, *in* '2013 American control conference', IEEE, pp. 2952–2957.

Lazar, M. & Pastravanu, O. (2002), 'A neural predictive controller for non-linear systems', *Mathematics and Computers in Simulation* **60**(3-5), 315–324.

Lu, F., Wu, J., Huang, J. & Qiu, X. (2019), 'Aircraft engine degradation prognostics based on logistic regression and novel os-elm algorithm', *Aerospace Science and Technology* **84**, 661–671.

Mehrpanahi, A., Hamidavi, A. & Ghorbanifar, A. (2018), 'A novel dynamic modeling of an industrial gas turbine using condition monitoring data', *Applied Thermal Engineering* **143**, 507–520.

Merz, C. J. & Pazzani, M. J. (1999), 'A principal components approach to combining regression estimates', *Machine learning* **36**(1-2), 9–32.

Montazeri-Gh, M. & Rasti, A. (2019), 'Comparison of model predictive controller and optimized min-max algorithm for turbofan engine fuel control', *Journal of Mechanical Science and Technology* **33**(11), 5483–5498.

Mu, J., Rees, D. & Liu, G. (2005), 'Advanced controller design for aircraft gas turbine engines', *Control Engineering Practice* **13**(8), 1001–1015.

Oviedo, J. J. E., Vandewalle, J. P. & Wertz, V. (2006), *Fuzzy logic, identification and predictive control*, Springer Science & Business Media.

Pandey, A., de Oliveira, M. & Moroto, R. H. (2018), Model predictive control for gas turbine engines, *in* 'ASME Turbo Expo 2018: Turbomachinery Technical Conference and Exposition', American Society of Mechanical Engineers Digital Collection.

Pires, T. S., Cruz, M. E., Colaço, M. J. & Alves, M. A. (2018), 'Nonlinear model predictive control applied to transient operation of a gas turbine', *Journal of Sustainable Development of Energy, Water and Environment Systems* **6**(4), 770–783.

Poursafar, N., Taghirad, H. & Haeri, M. (2010), 'Model predictive control of non-linear discrete time systems: a linear matrix inequality approach', *IET control theory & applications* **4**(10), 1922–1932.

Richter, H. (2011), *Advanced control of turbofan engines*, Springer Science & Business Media.

Rooney, N. & Patterson, D. (2007), 'A weighted combination of stacking and dynamic integration', *Pattern recognition* **40**(4), 1385–1388.

Rusnak, A., Fikar, M., Najim, K. & Mészáros, A. (1996), 'Generalized predictive control based on neural networks', *Neural Processing Letters* **4**(2), 107–112.

Saleh, E. B. E. (2017), Modeling and Simulation of A Double Spool Turbofan Engine Using SIMULINK®, PhD thesis, Zagazig University Zagazig.

Salehi, A. & Montazeri-Gh, M. (2018), 'Black box modeling of a turboshaft gas turbine engine fuel control unit based on neural narx', *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment* .

Salehi, A. & Montazeri-GH, M. (2020), 'Design and hil-based verification of the fuel control unit for a gas turbine engine', *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* p. 0954410020910593.

Soares, S. G. & Araújo, R. (2015), 'A dynamic and on-line ensemble regression for changing environments', *Expert Systems with Applications* **42**(6), 2935–2948.

Tayarani-Bathaie, S. S., Vanini, Z. S. & Khorasani, K. (2014), 'Dynamic neural network-based fault diagnosis of gas turbine engines', *Neurocomputing* **125**, 153–165.

Van Essen, H. & De Lange, H. (2001), 'Nonlinear model predictive control experiments on a laboratory gas turbine installation', *J. Eng. Gas Turbines Power* **123**(2), 347–352.

Wang, L. (2009), *Model predictive control system design and implementation using MATLAB®*, Springer Science & Business Media.

Wen, L., Dong, Y. & Gao, L. (2019), 'A new ensemble residual convolutional neural network for remaining useful life estimation', *Math. Biosci. Eng* **16**, 862–880.

Xu, Y., Xu, R. & Yan, W. (2017), Power plant performance modeling with concept drift, *in* '2017 International Joint Conference on Neural Networks (IJCNN)', IEEE, pp. 2096–2103.

Yu, B. & Shu, W. (2017), Research on turbofan engine model above idle state based on narx modeling approach, *in* 'IOP Conference Series: Materials Science and Engineering', Vol. 187, IOP Publishing, p. 012002.

Zhang, C. & Ma, Y. (2012), *Ensemble machine learning: methods and applications*, Springer.