Paper: ASAT-14-211-CT

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

A Neural Predictive Control Scheme for Small Turbojet Engines

I.M. Atia[*] and A.M. Bayoumy[†]

Abstract: Artificial Neural Networks (NN) is a well-known tool among artificial intelligence techniques that are able to reproduce arbitrary nonlinear relationships existing between input and output variables. Model based Predictive Control (MPC), or simply predictive control, is a family of control schemes that uses a model from the plant as a predictor of the future plant outputs and hence optimizes the future control inputs for the minimum future errors and minimum control energy. Among this family Generalized Predictive control (GPC) is one of the most famous.

In another part of this work [5],, a neural network representation is shown to be suitable for modeling a small gas turbine engine (SR-30). In the present paper, this model is used in a model-based predictive control scheme. The results of this controller are compared with a classical Proportional-Integral-Derivative (PID) controller tuned offline with a genetic optimization technique. Both are tested on the SR-30 turbojet engine model.

PID controller cannot cope with model changes in the whole operating range of the engine and therefore a predictive control scheme is then proposed as a solution to this problem. A neural model is used as a predictor for the calculation of GPC parameters. The nonlinear system free response is obtained by recursive future predictions while the dynamic response matrix is obtained by instantaneous linearization of the input /output relation.

The results illustrate the improvements in control performance that could be achieved with a neural predicative scheme compared to that of a classical PID controller.

Keywords: Small turbojet engines, artificial intelligence, neural networks, predictive controller, PID, GPC.

Nomenclature

ARX	AutoRegressive with eXternal input
F	Vector of predicted free response
G	system impulse response matrix
G_f	Fuel flow rate (kg/s)
K_d	derivative gain
K_i,	
K_p,	proportional gain, integral gain
N	Engine revolution speed,(rpm)
N_1	lower value of predicting horizon
N_2	Higher value of predicting horizon
NN	Neural networks

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

NNGPC	Neural network generalized predictive controller
N_u	Control horizon
PID	Proportional, integral and derivative controller
T	Sampling time
t_r	Rise time
t_s	Settling time
$\hat{\mathbf{y}}$	Vector of predicted outputs for prediction horizon
$\tilde{\mathbf{u}}$	Vector of future control increments for the control horizon
$\mathbf{w}$	Vector of future references
λ	Weighting factor for control increments

Introduction

Model Based Predictive Control (MBPC), or simply Predictive Control, is a family of algorithms with common strategy. MBPC appeared in the decade of 1970s and had got a good reputation in the chemical industries and process control [1].

The main strategy of the MBPC is as follows, (**Error! Reference source not found.**): A model of the controlled system is used to predict its behaviour in the future. A known required reference trajectory is then given for certain prediction horizon. Then an optimisation algorithm is used to find the optimum control sequence for certain number of steps in the future that minimise a certain cost function which includes future predicted errors and control increments. A receding horizon technique is then applied where only the first control signal of the optimum future control sequence is applied to the controlled system.

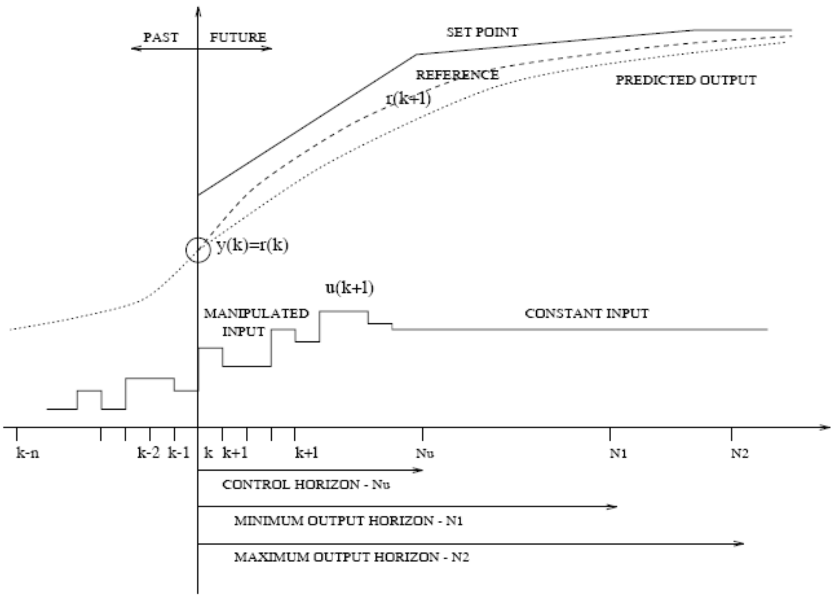


Fig. 1 Prediction strategy

Generalized Predictive Control (GPC) was developed by Clarke et al. in 1987 [2]. The GPC uses ideas from Generalized Minimum Variance (GMV) [3] and is perhaps one of the most popular methods at the moment.

Since the last three decades predictive control has shown to be successful in control industry. Generalized Predictive Control (GPC) was one of the most famous linear predictive algorithms. The control law of GPC contains two parameters that describe the system dynamics: system free response (f) and system impulse response matrix (G). Often these parameters are calculated from the discrete linear model. For nonlinear systems, either a nonlinear system model is instantaneously linearized or a nonlinear optimization is used. The validity of the linear model is the shortcoming of the first one and the possibility of non-uniqueness of local minimum is that for the second. The neural network (NN) model is used as a predictor to calculate these parameters for GPC.

The nonlinear system free response is obtained instantaneously while dynamic response is linearized every batch of time. This method [4] is tested on a benchmark nonlinear model. Results are compared with that of other neural predictive techniques found in previous literature. Also, this method in[4] is applied and validated on a realistic multivariable aircraft model. The simulation results show that this method has some good advantages over others neural predictive techniques. In one hand, the system dynamics parameters are calculated more accurately directly from the nonlinear NN model. And in the other hand, the used linear GPC has a cost function with only one global minimum. The method in [4], as a trade-off between nonlinear neural predictive control (NPC) and instantaneous linearization approximate neural linear predictive control (APC), is promising for control of nonlinear systems.

A. Watanabe et al [7] 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.

R. Andoga et al [8] discussed digital electronic control of a small turbojet engine. They stated that the main purpose of control of gas turbine was increasing its safety and efficiency. Their engine was controlled by PIC 16 F84A microcontroller, which manipulating the fuel flow valve.

M.Lichtsinder et al. [9] worked on development of a simple real-time transient performance model for AMT jet engine. The fast model is obtained using the Novel Generalized Describing Function, proposed for investigation of nonlinear control systems. They presented the Novel Generalized Describing Function definition and then discusses the application of this technique for the development a fast turbine engine simulation suitable for control and real-time applications.

In another part of this work [5], a neural networks representation is shown to be suitable for modeling a small gas turbine engine (SR-30). In the present work, this model is used in a model-based predictive control scheme. This model is linearized at different engine design points, this linearized model is used in design of a classical PID controller. The PID controller is tuned offline with a genetic optimization technique. Both controllers are tested on the SR-30 turbojet engine model and comparison is made between the results from these controllers with the same input.

Turbojet Engine Controller Design

For a gas-turbine engine, particularly for a jet engine, the speed n control is one of the most important aspects (even most important than the engine temperature control) and it is currently realized by some specific hydro-mechanical or electro-mechanical controllers.

The engine speed is the most important operating parameter, especially for the multi-spool engines, because it represents the parameter which assures the most accurate co-relation with the engine thrust amount, as well as with the engine fuel consumption; meanwhile, the speed *n* offers an image about the dynamic load of the engine mobile parts (compressor blades and disks, turbine blades and disks, shafts), as well as an indirect image about the thermal charge of the engine hot parts (combustor, turbine(s), exhaust nozzle).

An aircraft engine operates at various flight regimes, that means at various flight speed and flight altitudes, which means that the engine thrust variation must follow the aircraft flight dynamics necessities, therefore the engine speed (and thrust) must be strictly controlled, because of its important operating role.

The engine speed is one of the engine operating parameters, which is the easiest to measure, both for steady state regimes and for dynamical regimes. That fact represents an advantage and promotes the engine speed as the most important controlled engine parameter.

In this paper one has studied an engine speed controller with fuel flow rate as a regulating parameter. The controller design was based on engine neural networks model.

Discrete PID Based on Engine NN Model

The discrete PID controller was used with the NN model of the SR-30 turbo jet engine. Now, the tuning of the PID is achieved by using genetic algorithm. The GA is carried out using a MATLAB built-in routine so called Simulink Response Optimization (SRO) Toolbox as shown in Fig. 2. The SRO, automatically, formulates an optimization problem and calls a genetic algorithm and direct search toolbox, as an optimization routine to solve the problem.

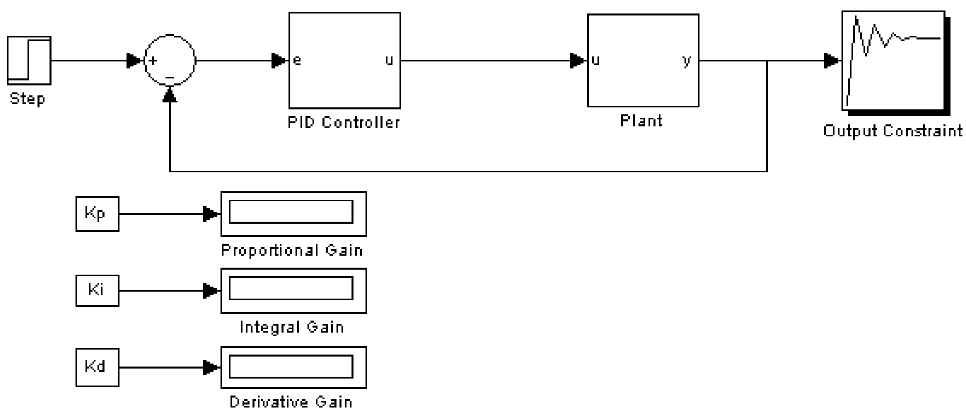


Fig. 2 Simulink Response Optimization (SRO) Toolbox

A classical discrete PID control system can be described as shown. The input-output relation of the PID controller is expressed mathematically by equation (1)[13].

$$u(t) = K_p \cdot e(t) + TK_i \sum_{t=0}^{N} e(t) + K_d \frac{e(t) - e(t-1)}{T} \tag{1}$$

where, $u(t)$ is the control signal, $e(t)$ is the error signal, and K_p, K_i, and K_d denotes the proportional gain, integral gain and derivative gain respectively, T is the sampling time and N is the number of samples, $u_1(t)$ represents the output of the controller at the sampling point (t).

If the sampling period is short enough, the approximate calculation by equation (1) can get an accurate result and the discrete control process is close to the continuous control process.

The digital PID Controller transfer function as a function of z has the following form [13]

$$C(z) = K_p + K_i T\left(\frac{z}{z-1}\right) + \frac{K_d}{T}\left(\frac{z-1}{z}\right) \quad (2)$$

where: K_p, K_i and K_d are the proportional, integral and derivative parameters of the controller respectively and 'T' the sampling time. The required step response characteristics of the engine are rise time (t_r) = 0.872 s, settling time (t_s) = 4 s and maximum overshoot (M_p) = 2%.

PID controller is tuned, based on the linearized neural model at different operating points, with step change from 41050 to 82000 rpm. The optimal PID parameters are shown in Table (1) and the resulted step response shown in **Error! Reference source not found.** represents the engine response at step input from n=41050rpm to n= 82000 rpm. This input covers a wide range of engine speeds.

In contrast, if the same controller is used with an input of smaller amplitudes as shown in **Error! Reference source not found.** and **Error! Reference source not found.**, the response of the engine with the full range PID controller has a high over shoot response compared with the scheduled PID controller. This is due to the fact that the PID controller is a linear controller. It is thus not capable of dealing optimally with a nonlinear constrained system across its whole operating range.

Gain-scheduling PID controllers are proposed and their parameters are recalculated and shown in Table (1) using small-amplitude step inputs, to cover the engine operating ranges in which the data used for the estimation and validation are available.

Table (1) PID parameters at different step changes based on linearized neural network models and ARX model

Model	Step changes (rpm)	k_p	k_i	k_d	MSE
Linearized neural models at certain design points	41050-46050	22.3465	11.7934	2.6964	0.002554
	46050-51060	9.5792	8.7627	0.99334	0.00336
	51060-56050	6.1426	8.6645	1.2238	0.00371
	56050-61050	6.3598	9.2752	1.1889	0.003231
	61050-66060	8.3136	12.1784	0.99342	0.002649
	66060-71060	5.1877	8.9899	1.6088	0.003409
	71060-76100	7.479	13.4361	1.3721	0.00245
	76100-82000	4.6472	10.6291	0.70902	0.002762
Linearized model at n_o=61050 rpm	41050-82000	7.4465	11.5974	1.2223	0.02024

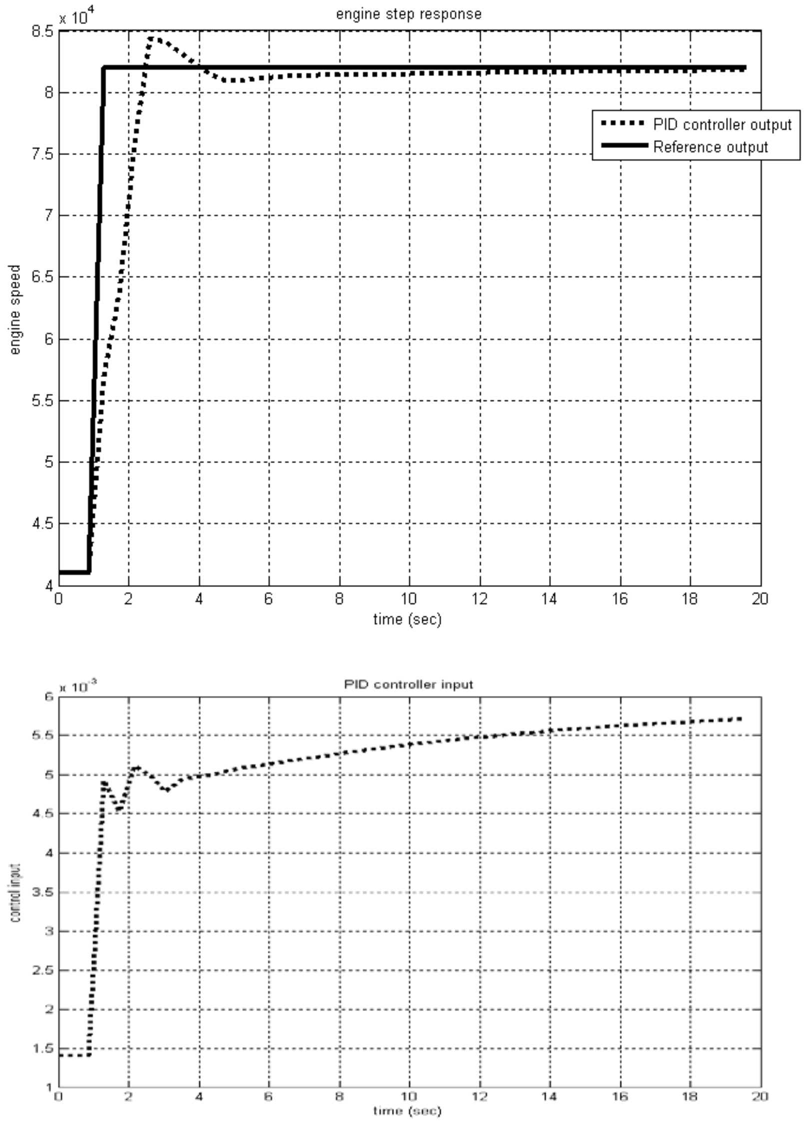
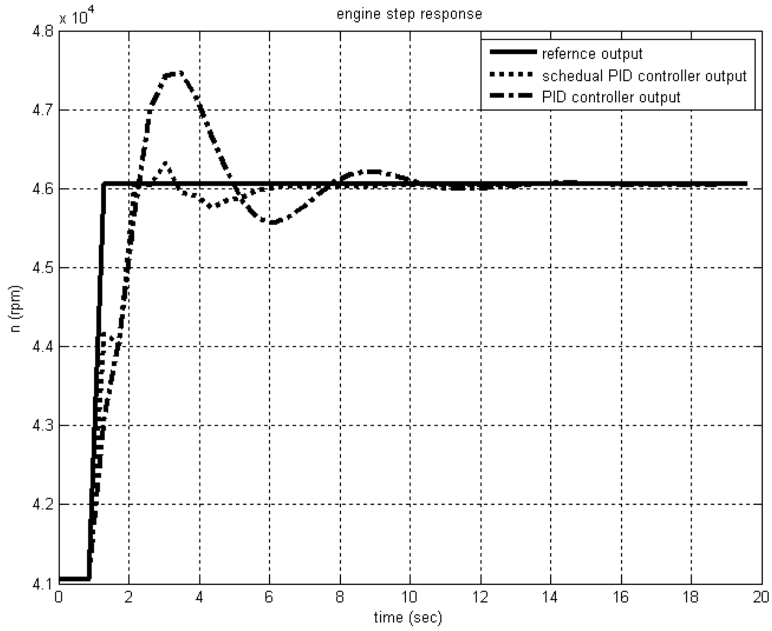


Fig. 3 Engine response with step change from n=41050 to n=82000 rpm.
(a) Engine step response, (b) PID controller input



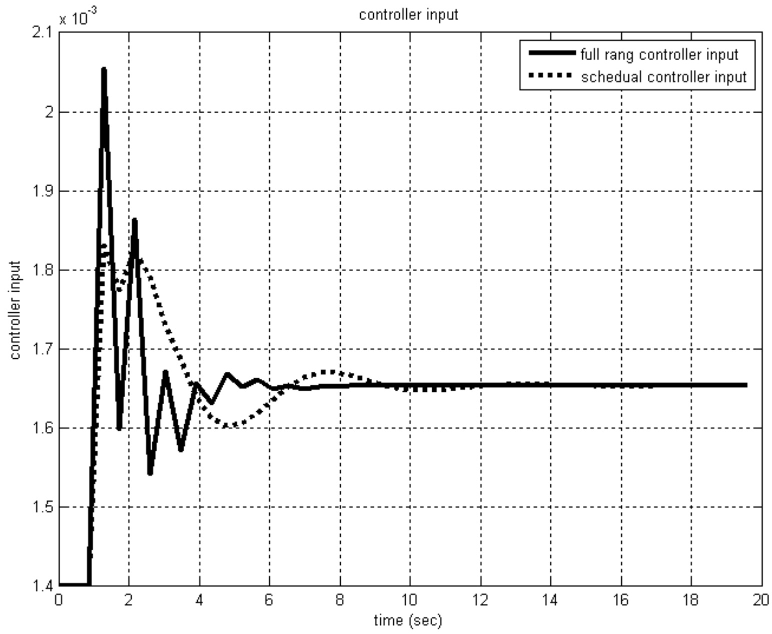


Fig. 4 Engine response with step change from n=41050 to n=46050
(a) Engine step response, (b) PID controller input

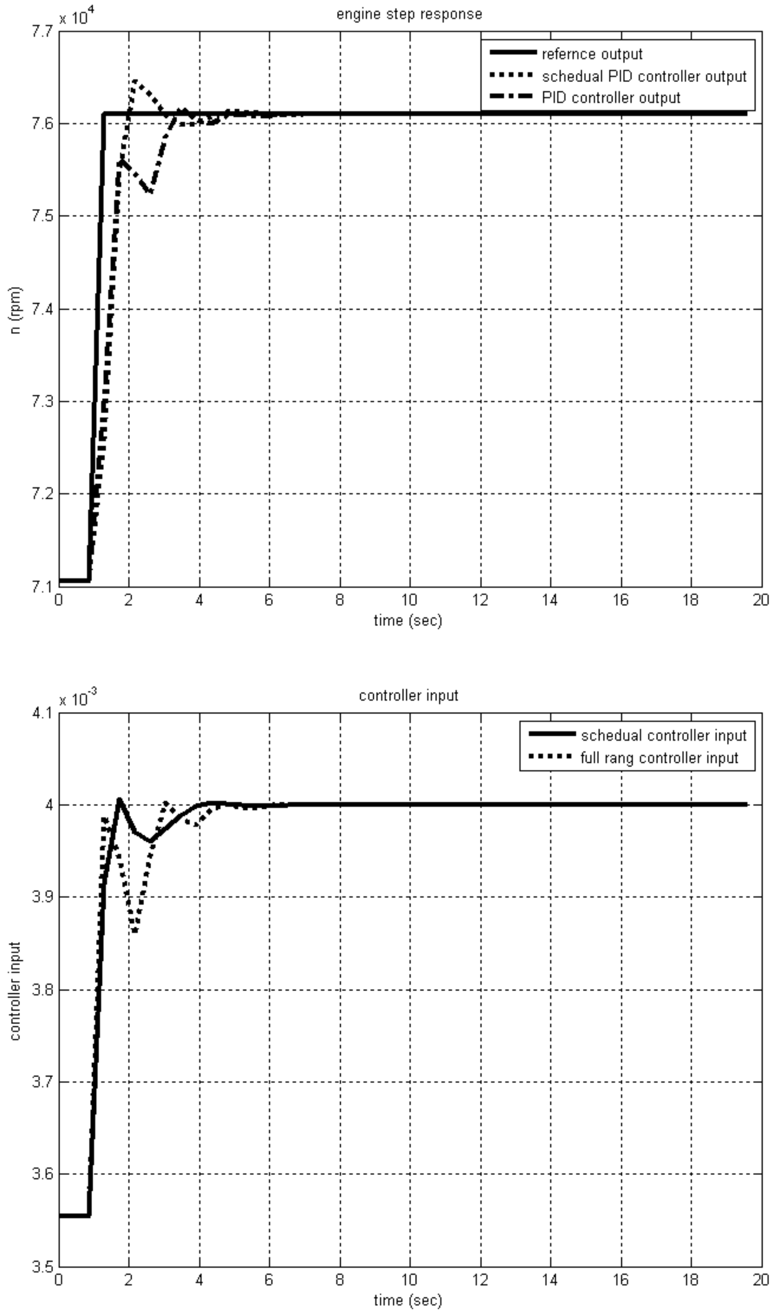


Fig. 5 Engine response with step change for n=71060 to n=76100
(a) Engine step response (b) PID controller input

Error! Reference source not found. represents the output from a neural model controlled with PID controller tuned at n=61050 rpm, the curve shows that the engine response became better as engine speed became near to the n_o= 61050 rpm and the error increased as the point became far away from the design point n_o.

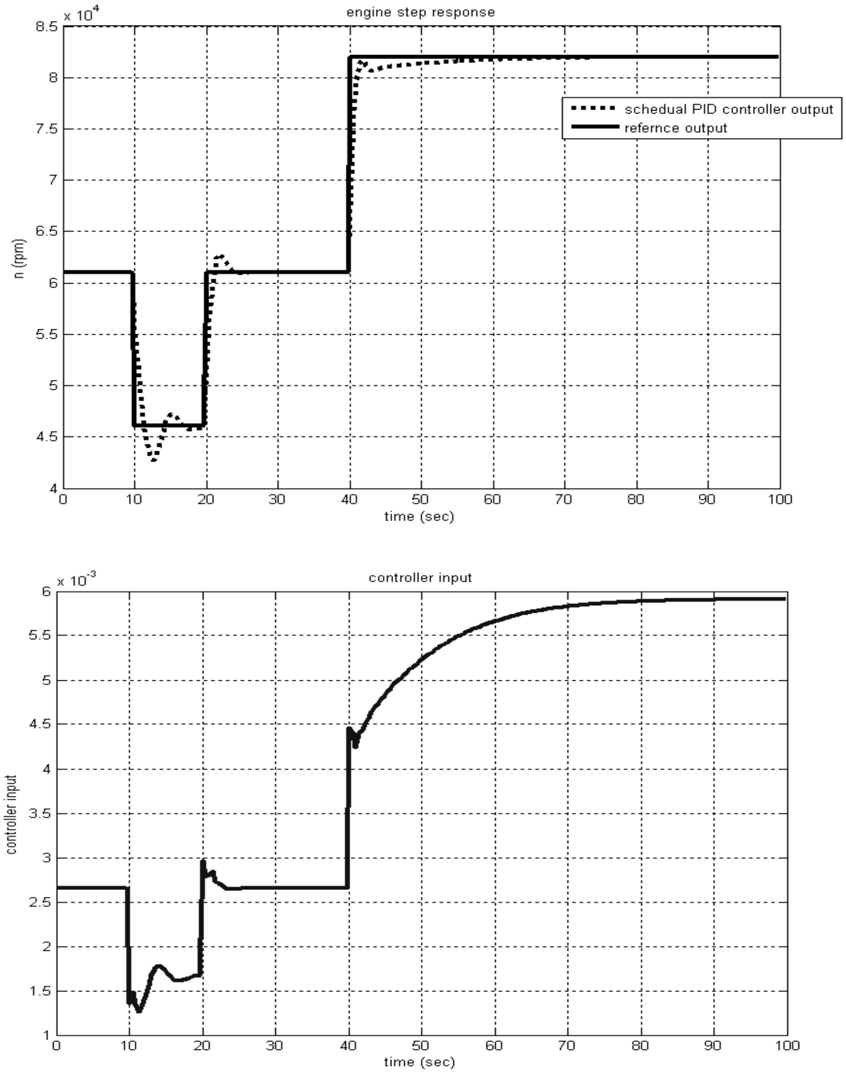


Fig. 6 Engine response with different steps change from n=61050 to n=82000
(a) Engine step response (b) PID controller input

The system nonlinearity is well illustrated if an increasing amplitude square pulse signal is given to the system. **Error! Reference source not found.** shows the response of the engine in case of square pulse signal input with the PID controller. There is an over shoot in the engine response.

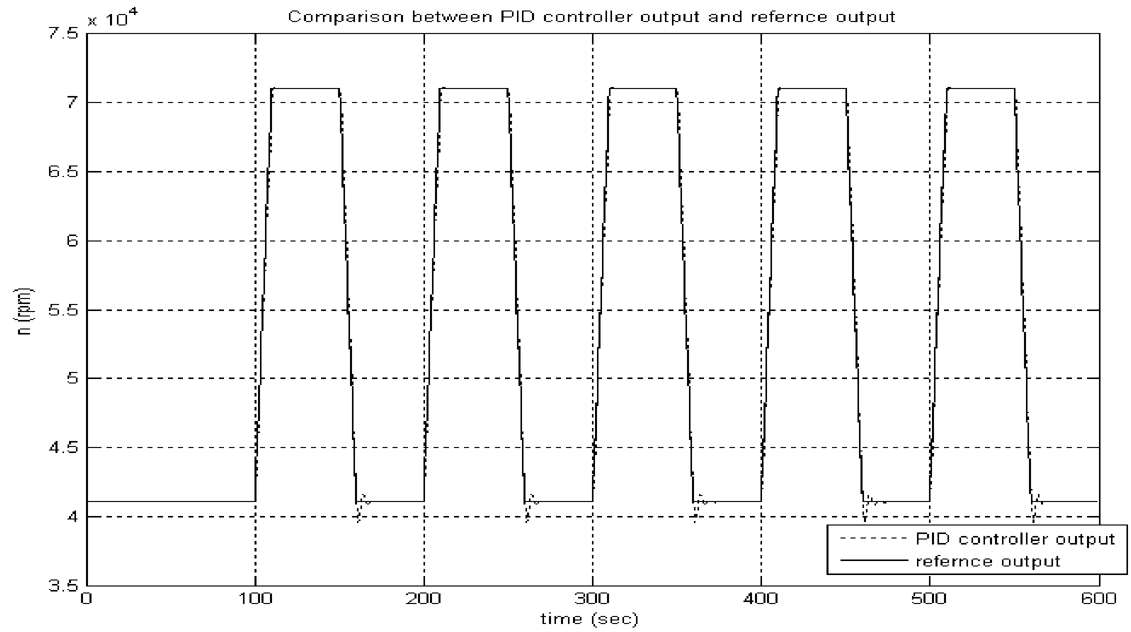


Fig. 7 The response of the engine in case of square pulse signal input with the PID controller

Predictive controller design

In order to implement the predictive controller strategy, the basic structure shown in **Error! Reference source not found.** is used. A model is used to predict the future plant outputs, based on past and current values and on the proposed optimal future control actions. These actions are calculated by the optimizer taking into account the cost function (where the future tracking error is considered) as well as the constraints.

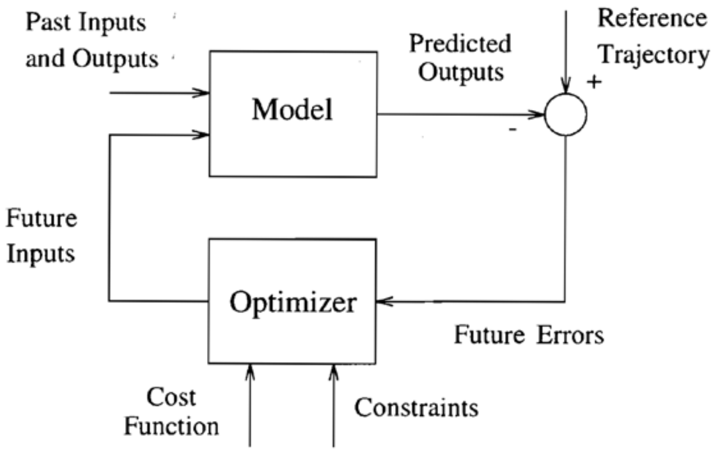


Fig. 8. Basic structure of MPC

The basic idea of GPC is to calculate a sequence of future control signals in such a way that it minimizes a multistage cost function defined over a prediction horizon. The index to be optimized is the expectation of a quadratic function measuring the distance between the predicted system output and some predicted reference sequence over the horizon plus a quadratic function measuring the control effort. GPC provides an explicit solution (in the absence of constraints), it can deal with unstable and non minimum phase plants and incorporates the concept of control horizon as well as the consideration of weighting of control increments in the cost function. The general set of choices available for GPC leads to a greater variety of control objectives compared to other approaches, some of which can be considered as subsets or limiting cases of GPC.

The GPC algorithm consists of applying a control sequence that minimizes a multistage cost function J:

$$J = \sum_{j=N_1}^{N_2} \left[\hat{y}\left(k+j\right) - w\left(k+j\right)\right]^2 + \lambda \sum_{j=1}^{N_u} \Delta u\left(k+j-1\right)^2 \tag{3}$$

Subject to: $\Delta u\left(k+j-1\right) = 0$ for $N_u < j < N_2$,where N_1 denotes the minimum prediction horizon, N_2 the maximum prediction horizon and N_u the control horizon, λ is a weight factor penalizing changes in the control input to obtain smooth control input signals and d is the system time delay.

Then the predictor equation becomes in matrix form

$$\hat{\mathbf{y}} = \mathbf{G} \cdot \tilde{\mathbf{u}} + \mathbf{f} \tag{4}$$

where:

$$\hat{\mathbf{y}} = \left[\hat{y}\left(k+N_1\right) \quad \hat{y}\left(k+N_1+1\right) \quad \cdots \quad \hat{y}\left(k+N_2\right)\right]^T$$

$$\mathbf{G} = \begin{bmatrix} g_{N_1} & g_{N_1-1} & \cdots & g_1 & 0 & \cdots & 0 \\ g_{N_1+1} & \ddots & & & \ddots & \ddots & \vdots \\ \vdots & & \ddots & & & g_1 & 0 \\ \vdots & & & \ddots & & & g_1 \\ g_{N_2} & g_{N_2-1} & & & \ddots & & g_{N_2-N_u+1} \end{bmatrix}$$

$$\tilde{\mathbf{u}} = \left[\Delta u\left(k\right) \quad \Delta u\left(k+1\right) \quad \cdots \quad \Delta u\left(k+N_u-1\right)\right]^T$$

$$\mathbf{f} = \left[f\left(k+N_1\right) \quad \cdots \quad f\left(k+N_2\right)\right]^T$$

Then J could be written in matrix form as:

$$J = \left(\hat{\mathbf{y}} - \mathbf{w}\right)^T \cdot \left(\hat{\mathbf{y}} - \mathbf{w}\right) + \lambda \tilde{\mathbf{u}}^{\mathbf{T}} \cdot \tilde{\mathbf{u}} \tag{5}$$

where:

$$\mathbf{w} = \left[w\left(k+N_1\right) \quad \cdots \quad w\left(k+N_2\right)\right]^T$$

Minimize J to get optimum $\tilde{\mathbf{u}}$ we get:

$$\tilde{\mathbf{u}}^* = \left(\mathbf{G}^T \cdot \mathbf{G} + \lambda \mathbf{I}\right)^{-1} \cdot \mathbf{G}^T \cdot \left(\mathbf{w} - \mathbf{f}\right) \tag{6}$$

Taking the first element of the control sequence (as the receding horizon principle)

$$\Delta u^*\left(k\right) = \mathbf{H} \cdot \left(\mathbf{w} - \mathbf{f}\right) \tag{7}$$

where:

$$\mathbf{H} = \left[1 \quad 0 \quad 0 \quad \cdots \quad 0\right]^T \left(\mathbf{G}^T \cdot \mathbf{G} + \lambda \mathbf{I}\right)^{-1} \cdot \mathbf{G}^T$$

For a linear time-invariant system the parameter $\mathbf{H}$ is unchanged over the time. But the free response $\mathbf{f}$ should be calculated every time step.

The incremental controller ensures zero offsets even with non-zero constant disturbance. The choices of parameters (N_1, N_2, N_u and λ) determine the stability and performance of the GPC controller. Some guidelines for selecting them exist in [6-10].

Free System Response (f)

To get the free system response the prescribed NN is given a zero increment vector $\hat{\tilde{\mathbf{u}}}$ then the output predicted vector $\hat{\mathbf{Y}}$ will be the system free response $\mathbf{f}$.

Impulse system response (G)

The impulse response of the system is calculated using trained NN model with a linearization around the current operating point. To get the first column of the G matrix a small value for $\Delta\hat{u}_{k+1}$ is assumed as small value ε, where $\varepsilon \ll 1$ and the corresponding output prediction is obtained.

$$\hat{\tilde{\mathbf{u}}} = \begin{bmatrix} \varepsilon & 0 & 0 & 0 \end{bmatrix} \tag{8}$$

Then the first column will be

$$\mathbf{G}_{(1)} = \frac{1}{\varepsilon} \cdot (\hat{\mathbf{Y}} - \mathbf{f}) \tag{9}$$

It will be easy after that to form the special shape of $\mathbf{G}$ matrix then calculate $\mathbf{H}$ vector.

Predictive controller scheme

The proposed control scheme **Error! Reference source not found.** consists of a nonlinear neural network model in the form of NN model and a linear GPC controller. The neural model is trained off-line within the complete range of system input. After performing the training, the network is then used by the GPC controller to calculate the free response of nonlinear system every time step. Every batch of time the impulse response matrix is calculated through linearization.

The control law **Error! Reference source not found.** is computed every time-step to get the next control increment.

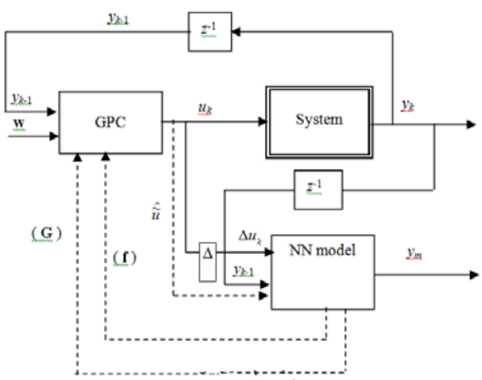


Fig. 9 Proposed Neural Network GPC Control Scheme

Application to the SR-30 NN model

The predictive controller parameters are $N_1 = 1$, $N_2 = 4$, $N_u = 1$, $\lambda = 0.05$

The system nonlinearity is well illustrated if an increasing amplitude square pulse signal is given to the system. The Simulations results and Comparison with the PID controller with the same input will be illustrated below.

Error! Reference source not found. shows the response of the engine in case of enhancing the full-step response from 41050 to 82000 rpm. It is clear that the oscillation around final position eliminated and the rise and settling time are reduced

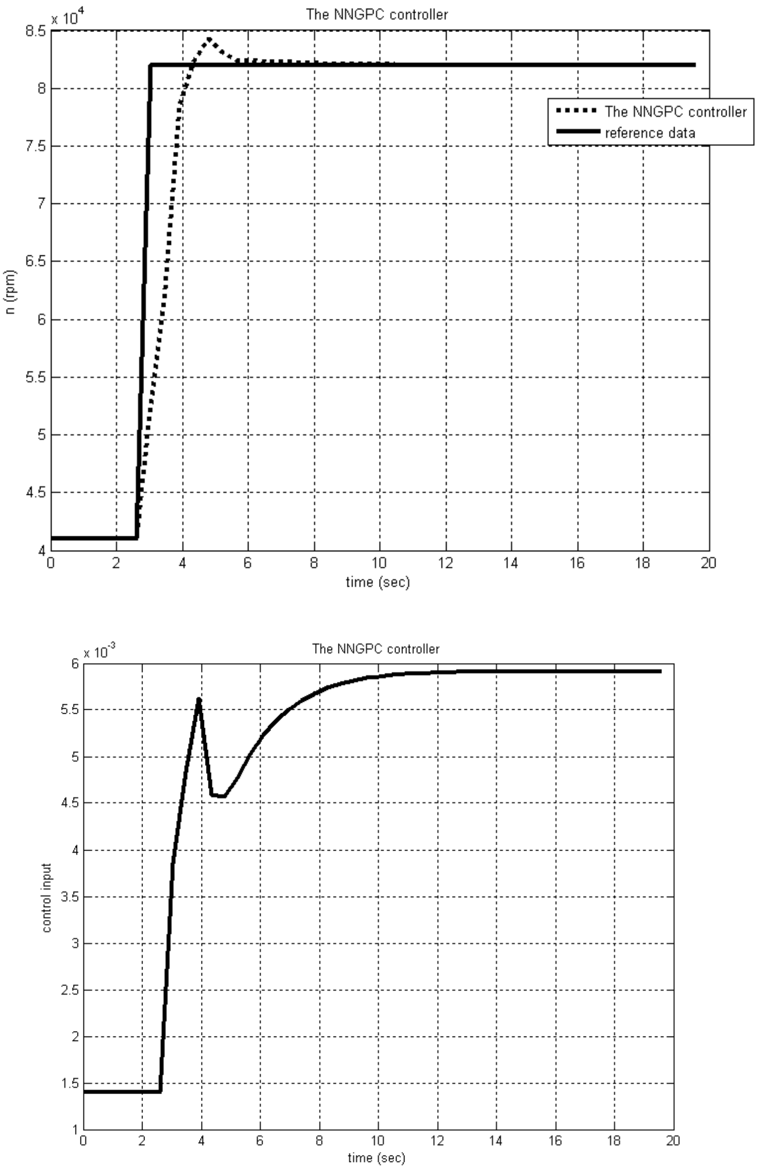


Fig. 10 Engine response with step change from 41050 to 82000 rpm
(a) engine step response, (b) predictive controller input

Error! Reference source not found. shows the engine step response with random step input from 61050 to 46050 finally to 82000 rpm. It is clear that the engine response with predictive controller is improved where the oscillations are reduced and the settling time is reduced.

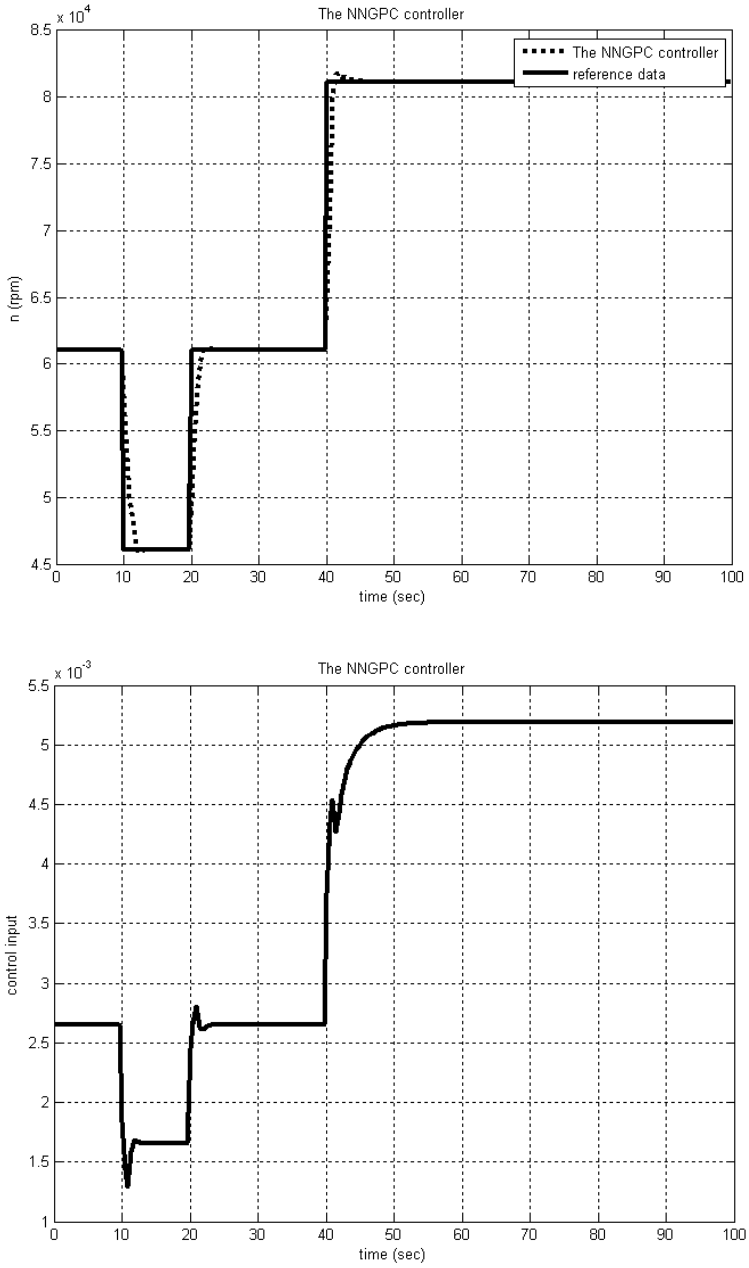


Fig. 11 Engine response with different steps change from n=61050 to n=82000
(a) engine step response, (b) predictive controller input

The system nonlinearly is well illustrated if an increasing amplitude square pulse signal is given to the system. **Error! Reference source not found.** the response of the engine in case of square pulse signal input with the predictive controller. It is clear that there is no over shoot in the engine response.

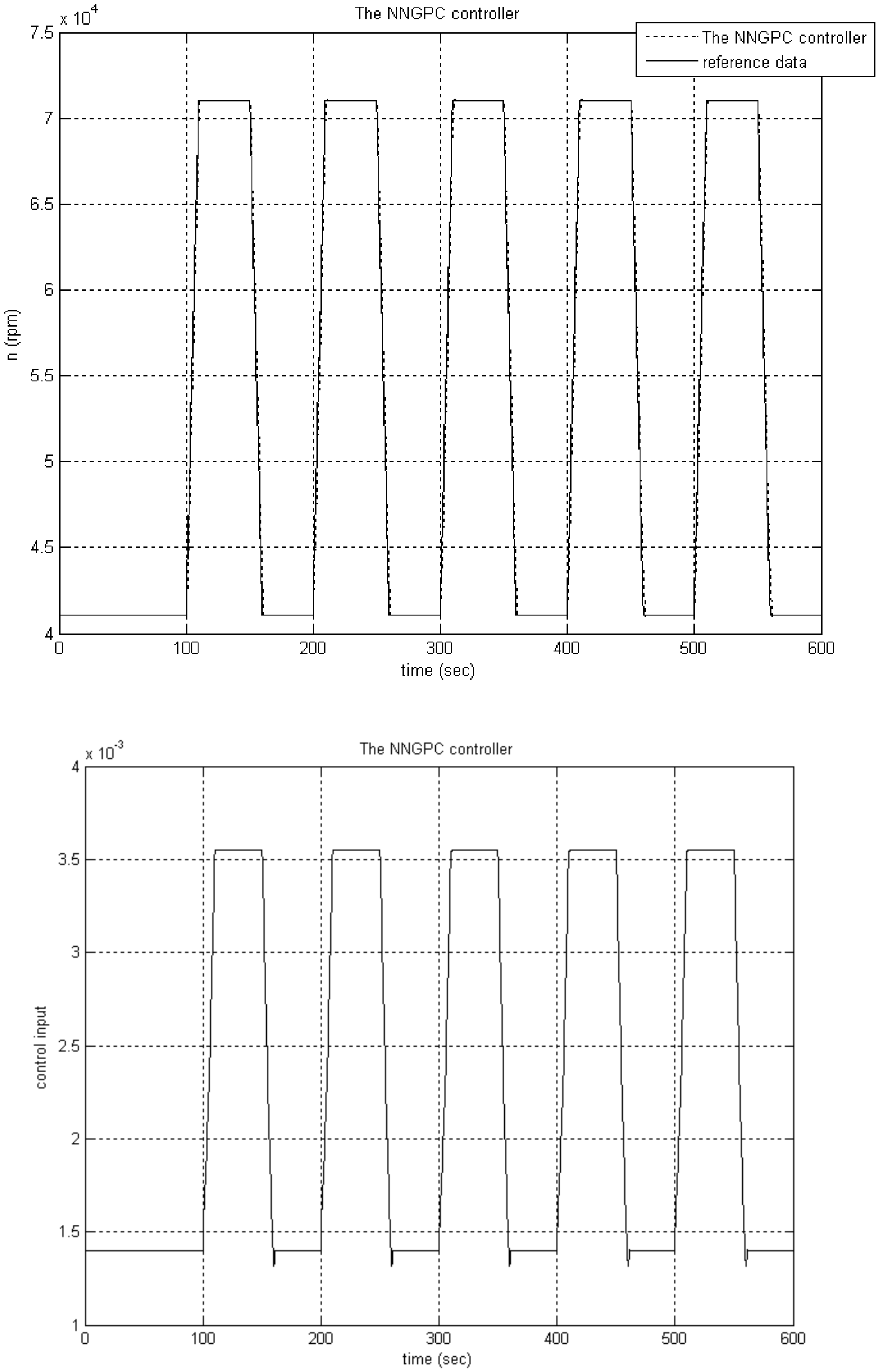
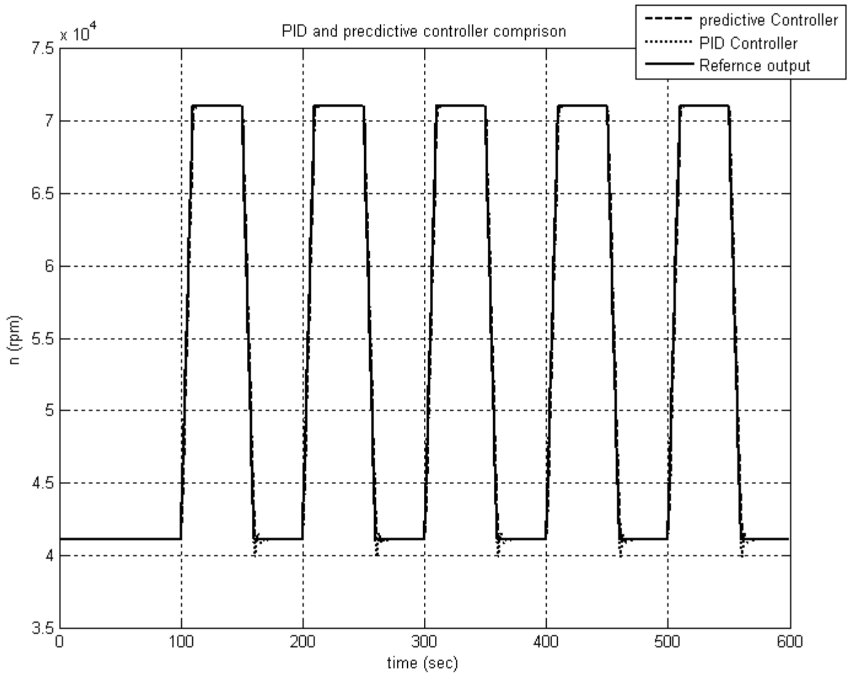


Fig. 12 Engine response in case of square pulse signal input
(a) engine step response, (b) predictive controller input

Results of comparison between PID controller and predictive controller

In this section, a comparison is made between the PID controllers with the predictive controller with respect to the same input signal as shown in **Error! Reference source not found.**, 14 and 15.



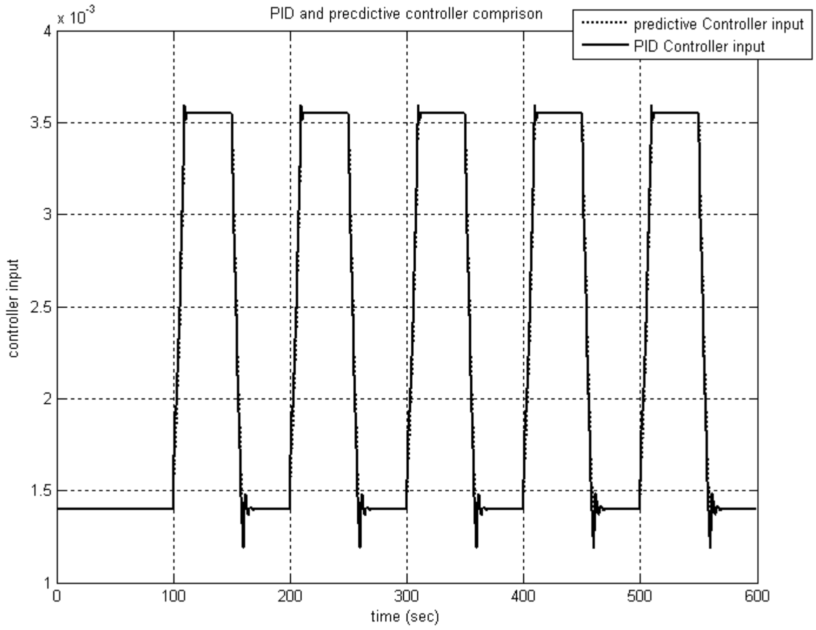
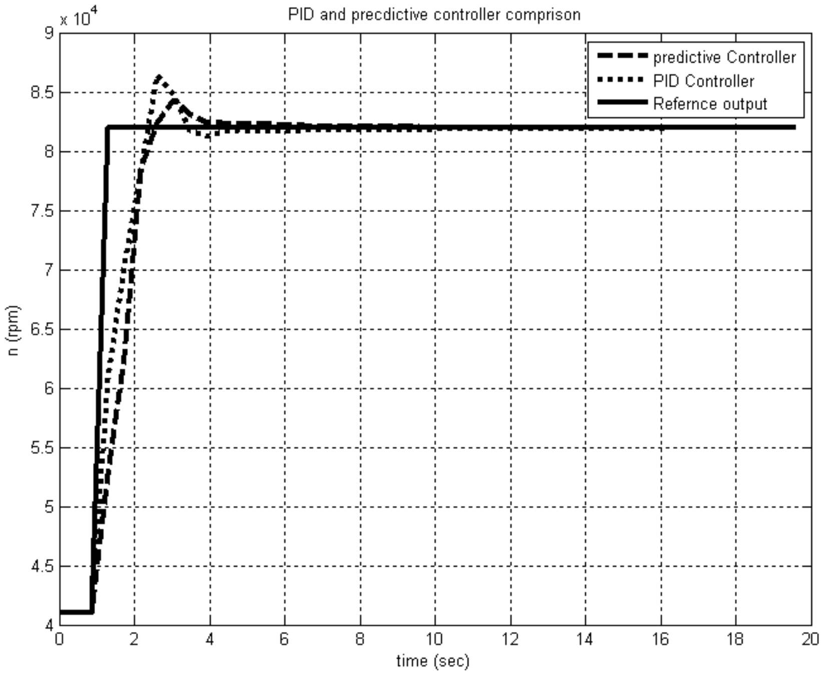


Fig. 13 Comparison between PID and predictive controllers in case of in case of square pulse signal input (a) engine step response, (b) controller input



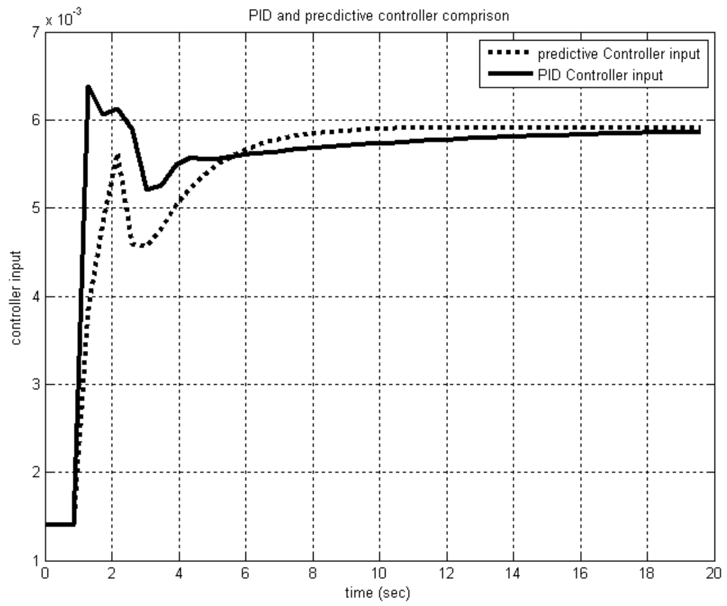


Fig. 14 Comparison between PID and predictive controllers in case of step input from 41050 to 82000 rpm (a) engine step response, (b) controller input

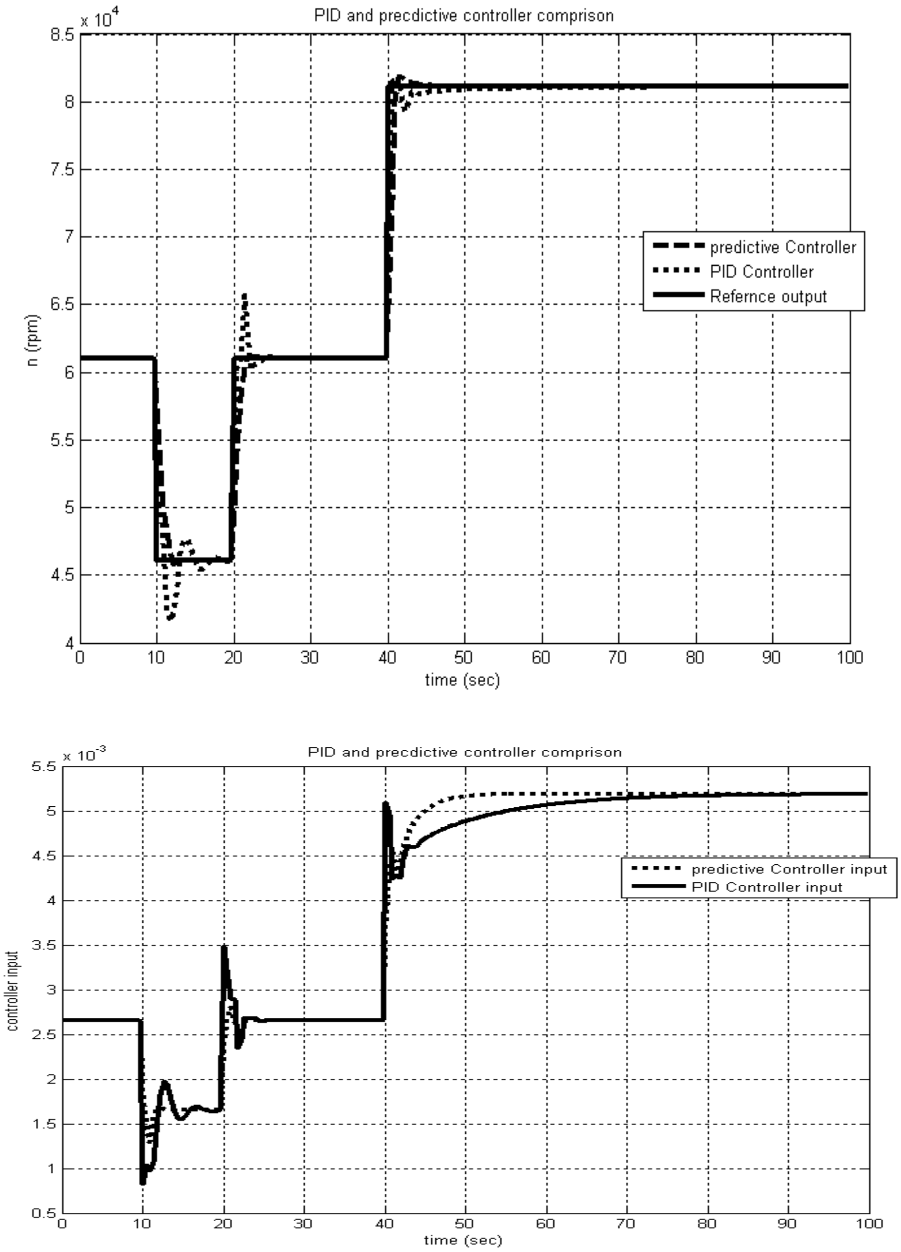


Fig. 15 Comparison between PID and predictive controllers in case of different steps input from 61050 to 82000 rpm (a) engine step response, (b) controller input

The Simulations results of the predictive controller and Comparison with the PID controller with the same input are illustrated above which show that the engine performance is improved with the predictive controller, the response oscillation and overshoot is smaller with the predictive controller rather than the PID controller, the engine rise time is also smaller. It can be therefore concluded that the parameters in the gain-scheduling PID controller need to be changed with the operating range, but using predictive controller enables a global controller to be implemented and provides the optimal control performance across the operating range. Predictive controller provides the best control performance against disturbances and model uncertainties.

Conclusion

A representative neural network SR-30 engine model was used to develop a PID controller and predictive controller. Results from the two controllers were compared and predictive controller found to be accurate for engine control during the full operating rang.

The following results are derived from our analysis:

1. PID controller was built based on the neural networks model of the SR-30 engine.
2. Tuning of the PID controller was performed with offline with a genetic optimization technique.
3. PID controller cannot cope with model changes in the whole operating range of the engine.
4. A neural model was used as a predictor for the calculation of GPC parameters. The nonlinear system free response was obtained by recursive future predictions while the dynamic response matrix was obtained by instantaneous linearization of the input /output relation.

As a conclusion, the results illustrate clearly the improvements in system performance that could be achieved with a neural predicative controller compared to that of a classical PID controller.

References

[1] Garcia C.E., Prett D.M., and M. Morari," *Model predictive control: theory and practice- a survey* ," Automatica, 25(3): 335-348, (1989).

[2] D.W. Clarke, C Mohtadi, and P.S. Tuffs. "*Generalized Predictive Control. Part I. The Basic Algorithm. Automatica*" 23(2):137-148, 1987.

[3] D.W. Clarke and P.J. Gawthrop. "*Self-tuning Control. Proceedings*" IEEE, 123: 633-640,1979.

[4] A. M. Bayoumy, J. Bordeneuve-Guibé , "*A Neural Predictive Control Scheme For Nonlinear Plants*", Department of Avionics and Systems, ENSICA, Toulouse, FRANCE, AIAA 2002-1541.

[5] I. M. Atia and A. M. Bayoumy "*Testing and Model Identification of a Turbojet Engine Using Neural Networks*". ASAT-14, 2011

[6] Clarke D.W., C. Mohtadi and P.S. Tuffs," *Generalized Predictive Control. Part2: Extensions and interpretations*," Automatica, vol.23, Nº2, pp. 149-160, (1987)

[7] 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.

[8] 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.

[9] Michael Lichtsinder, Yeshayahou Levy *"Jet Engine Model for Control and Real-Time Simulations"* Journal of Engineering for Gas Turbines and Power, 2006 by ASME, OCTOBER 2006, Vol. 128 / 745.

[10] Clarke D.W. and C. Mohtadi," *Properties of generalized predictive control*" Proceedings of the 10th triennial world congress of IFAC, Munich, FRG, pp. 65 -76, (1987).

[11] Clarke D.W. and R. Scattolini," *Constrained receding-horizon predictive control*" IEEE proceedings-D, vol.138, Nº4, pp.347-354, (1991)

[12] Rawlings J.B. and Muske K.R.," *The stability of constrained receding horizon control*" IEEE Trans. Automatic control, 38(10):1512 – 1516, (1993)

[13] Charles L Phillips, "*Digital control system analysis and design*".