Multivariable Predictive Control Design with Application to a Gas Turbine Power Plant

Abdallah Ali Omar

Industrial Control Centre
Department of Electronic and Electrical Engineering
University of Strathclyde
Graham Hills Building
50 George Street
Glasgow, G1 QE
Scotland, UK.

This thesis is submitted to the
University of Strathclyde
for the degree of
Master of Philosophy (M.Phil)
July 1997

Dedication:

This thesis is dedicated to my parents, my wife and my children.

Abstract

The application of Model-Based Predictive Control (MBPC) to Gas Turbine Power Plants is investigated. This begins with the discussion of the advantages and disadvantages of (MBPC). Three well-known algorithms namely, Model Algorithmic Control (MAC), Dynamic Matrix Control (DMC) and Generalised Predictive Control (GPC) are presented and compared.

A multivariable state space GPC algorithm is then presented and discussed. The general optimisation problem is addressed. Four types of constraints namely, constraints on control signal increments, constraints on control signal amplitude, constraints on process outputs and constraints on overshoot are extended to the multivariable state space GPC algorithm and the constrained GPC problem is solved as a Quadratic Programming (QP) optimisation problem. Analysis and design aspects of GPC control strategy are discussed. A method for multivariable state space GPC closed-loop formulation is developed to derive conditions for performance and stability robustness assessment of the control design. Simulation of a GPC controlled multivariable industrial process is presented and the results are analysed.

The operation of a single unit gas turbine power plant is discussed and a mathematical model is developed. The nonlinear model is implemented on SIMULINK and stabilised using PID controllers. A linear model is then derived and reduced using MATLAB and SIMULINK toolboxes for control design purposes.

A supervisory control system for the nonlinear power plant model is designed using the multivariable GPC algorithm. Closed-loop analysis is carried out and conditions for performance and the stability robustness are derived. Compromise among various design objectives is made for the final controller design. The performance of PID and GPC controllers is compared. A constrained supervisory control system is designed to handle process and actuator constraints.

A predictive controller in an adaptive (gain scheduling) manner is then proposed. The technique is applied to optimise the start-up and shut-down operations of the power turbine and to design a constrained adaptive predictive controller. For this purpose an identification experiment is designed and a recently developed

subspace-based identification algorithm is applied off-line to identify several linear models.

Acknowledgements

I would like to take this privilege to thank my supervisor Dr. Reza Katebi for his continuous support, his invaluable academic and technical comments and advice, and his regular feedback to achieve this research work.

I would like to thank all my colleagues and the staff in the Industrial Control Centre, especially Dr. Andrzej Ordys, Dr. Michael Johnson and Mr. Mahmoud El-Fandi for their help and useful discussions.

I am very grateful to my wife Nadya and my daughters for their patience, support and encouragement even in their most difficult times. I am also very grateful to my parents, my brothers and my sisters.

I would like to thank the project department in the Libyan Agip Oil Company for their trust and support. Also I would like to thank the Libyan Agip Oil Company for financial support.

Contents

List of Figures

List of Tables

Notation

Symbols

$\hat{y}_{k+j}$	Optimal outputs predictor
$M(\theta)$	Process model as a function process parameters θ
w_k	Reference input
y_k	Process output
k	Discrete time operator
ζ_k	A general disturbance term
w_{k+j}	The future reference input
e	Tracking error
h_j	The j *th* impulse response element
N_r	The order of FIR model
N_s	The number of step response coefficients of FSR model
$N1$	The minimum costing horizon
$N2$	The maximum costing horizon
Nu	The control horizon
u_{k+j}	The process input j steps ahead in future
g_j	The jth step response coefficient
Δ	The differencing operator $(1-q^{-1})$
d	The process time delay
dl	Process dead time
$B(q^{-1})$, $A(q^{-1})$,&	
$D(q^{-1})$ & $C(q^{-1})$	Polynomials in backward shift operator
u_{k-1}	Delayed input of the process
A	n×n state (system) matrix
B	$n \times p$ input matrix
C	$r \times n$ output matrix
D	$r \times p$ direct transmission matrix
E	n×n diagonal disturbance matrix

$E(.)$	Expectation operator
Q_w	Process noise covariance matrix
R_v	Measurement noise covariance matrix
$\hat{x}_k$	Estimate state vector
γ	Weighting on tracking error
x_k	$n \times 1$ state vector
w_{wk} & v_k	Uncorrelated discrete white noise sequences with zero mean and variance σ^2
$D(k)$	Drift disturbance polynomial
ζ_{k1}	A filtered drift disturbance
ζ_{k2}	A filtered known measurable disturbance
ζ_{k3}	A stochastic disturbance
$vm(k)$	A known measurable disturbance
q^{-1}	The backward shift operator
p_j	The free response of the system
$G_j(q^{-1})$	A polynomial of step response coefficients in the backward shift operator $=g_1q^{-1}+\cdots+g_jq^{-j}$
G	A lower triangle matrix of step response coefficients
λ	Penalising factor for the excessive increment of the incremental control action
τ	Time constant
τ_s	The sampling time
$E_j(q^{-1})$, $F_j(q^{-1})$ & $\Gamma_j(q^{-1})$	Polynomials that are uniquely defined
J	Denotes a cost function
Δu	A vector of increments of control signals
Δu_{pk}	The increment of the pth control input
$\hat{y}_{rk+j}$	The rth predicted output
Hm	A matrix contains step response coefficients for $i \geq 0$
Γ & Λ	Diagonal weighting matrices

N_{lu} & N_{uu}	The lower and the upper control constraint horizons
L	A lower triangle matrix
$N_{l\Delta u}$ & $N_{u\Delta u}$	The lower and the upper control increment constraint horizons
N_{ly} & N_{uy}	The lower and the upper output constraint horizons
N_{lo} and N_{uo}	The lower and the upper overshoot constraint horizons
Hmy	A submatrix of Hm
Fy	A submatrix of F
Hmo	A submatrix of Hm
Fo	A submatrix of F
$\sigma_i(A)$	Denotes the singular values of a matrix $A \in R^{m\times n}$
A^*	A complex conjugate of a matrix A
$\overline{\sigma}(A)$ & $\underline{\sigma}(A)$	Denote the maximum and the minimum singular values of the matrix A
$\lVert A \rVert_2$	The 2-norm of matrix A
K_{gpc}	GPC gain matrix
F	A constant gain matrix which can be regarded as a state feedback gain matrix
$K_{\hat{x}u}$ & $K_{\hat{x}y}$	System matrices that represent the effects of inputs and Outputs on Kalman filter
T_{yd}	Sensitivity transfer function matrix
T_{yn}	Complementary sensitivity transfer function matrix
$T_{\Delta uw}$	Control sensitivity transfer function matrix
M_p	A transfer function matrix seen by the additive perturbation
M_i & M_o	Transfer function matrices seen by the input and the output multiplicative perturbations
w_f -	Fuel mass flow
k_{ff} -	Fuel system gain constant
τ_f -	Fuel system time constant

e_1 -	Valve position
e_2 -	Internal signal
MF -	Minimum fuel signal
k_f -	Feedback coefficient
F_d -	Fuel demand signal
τ_d -	Fuel system pure time delay
ω -	Rotation speed of the turbine
w_a -	Air mass flow into the compressor
A_o -	Compressor exit flow area
$\eta_{\infty c}$ -	Compressor polytropic efficiency
ρ_i -	Inlet air density
p_{cin} -	Inlet air pressure
m_a -	Polytropic index
r_c -	Pressure ratio (outlet/inlet)
$\gamma_a = (c_{pa} / c_{va})$	Ratio of specific heats for air (constant)
c_{pa} -	Specific heat at constant pressure for air
c_{vv} -	Specific heat at constant volume for air
T_{cout} -	Outlet air temperature
T_{cin} -	Inlet air temperature
p_c -	Compressor power consumption
Δh_I -	Isotropic enthalpy
η_c -	Overall compressor efficiency
η_{trans} -	Transmission efficiency from turbine to compressor
R_a -	Air gas constant
w_f -	Fuel mass flow
w_{is} -	Injection steam mass flow
c_{pg} -	Specific heat of combustion gases(constant)
T_{Tin} -	Turbine inlet gas temperature
Δh_{25} -	Specific enthalpy of reaction at a temperature 25° C
c_{ps} -	Specific heat of steam (constant)

T_{is} -	Temperature of injected steam
p_{Tin} -	Pressure of combustion gases at turbine inlet
Δp -	Combustion chamber pressure loss
k_1, k_2 -	Pressure loss coefficients
R_{cg} -	Universal gas constant for combustion gases
A_m -	Combustion chamber mean cross-sectional area
g'_{cco} -	Mass flow of CO
τ_m -	Measurement delay
ρ_{Tin} -	Inlet gas density
m_{cg} -	Combustion gases polytropic index
A_{To} -	Turbine exit flow area
$\eta_{\infty T}$ -	Turbine polytropic efficiency
w_G -	Inlet gas flow from combustor
T_{Tout} -	Gas temperature at exit of turbine
η_T -	Overall turbine efficiency
p_T -	Mechanical power delivered by turbine
p_c -	Power required to drive the compressor
p_{mech} -	Net available mechanical power
Δh_I -	Isentropic enthalpy
h_{Tout} -	Exhaust gas enthalpy
H_{eq} -	Inertia constant for machine
p_{mech} -	Mechanical shaft power
p_{el} -	Real electrical load power
D_{eq} -	Electrical damping coefficient
wo -	Synchronous frequency
k_p	Proportional gain constant
k_i	Integral gain constant
k_d	Differential gain constant
PID	Proportional Integral Differential control action
δ_{kl}	Kronecker index

s	Stands for stochastic
d	Stands for deterministic
x_k^d	A deterministic state sequence
y_k^d	A deterministic output
x_k^s	A stochastic state sequence
$U_{o/i-1}$	Input block Hankel matrix
$Y_{o/i-1}$	Output block Hankel matrix
$Y_{o/i-1}^s$	Stochastic subsystem output block Hankel matrix
Γ_i	The extended (means $i > n$) observability matrix
Δ_i^d	The reversed extended controllability matrix
H_i^d	Block Toeplitz lower triangular matrix
L_i^s	Block Toeplitz covariance matrix
H_i^s	Block Toeplitz cross covariance matrix
Z_i	Denotes orthogonal projection
$\hat{X}_i$ $\hat{X}_{i+1}$	State sequences that can be interpreted as the states of a bank of non-steady state Kalman filters
Γ_{i-1} & $\underline{\Gamma_i}$	Extended observability matrix formed by deleting the last *r* rows of Γ_i
R_r	R factor of RQ decomposition
$Q_r{}^t$	Q factor of RQ decomposition
τ_r	Process rise time
ω_s	Sampling frequency in (rad/sec)
ω_b	The 3dB band width
f_s	Sampling frequency in Hz

Preface

Introduction:

In process industry and electrical power plants it is required to optimise the operations of gas turbines to improve their performance and to achieve fast response to load demand changes.

The objective of this thesis is to design predictive controllers in an adaptive manner (gain scheduling) to optimise the start-up, steady state and shut-down operations of gas turbines power plant and to achieve smooth transition from one operating point to another. The advantage of this technique is that the regulator parameters can be changed quickly in response to the process changes. This advantage is exploited to design a controller to minimise the time required for start-up and shut-down of the gas turbines. The Control design technique adopted in this thesis is the Model Based Predictive Control (MBPC) technique,(specifically GPC) which has found a wide acceptance in process industry. MBPC is characterised by its capability to cope with a wide range of processes. It is effective with, processes which are simultaneously nonminimum-phase and open-loop unstable, processes with unknown or variable dead-time, processes with over- and under- parametrized models and processes with variable model orders.

The main advantage of this control design technique is, however, the systematic way of handling process and actuator constraints. To achieve a stable control design with good performance, constraints imposed on the process variables must be accounted for during the design stage of the controller.

To design a MBPC controller a model of the process is required. Thus, dynamic modelling of the power gas turbines using the basic laws of physics as well as those based on system identification are considered.

Achievements of the Research:

1. The author has implemented a nonlinear model of a gas turbine power plant on SIMULINK and MATLAB. This model was used by the author to investigate and optimise the operations of start-up and shut-down of an industrial gas turbine plant. This original research has been accepted for conference presentation.
2. Full presentation of multivariable state space GPC has been given including an analysis of its closed-loop structure. The author developed ways to incorporate operational constraints for multivariable systems and study performance and stability robustness. Contributions original to the author were to use the subspace algorithm for the production of linear models within the multivariable GPC algorithm, and a method of gain scheduling multivriable GPC control for start-up and shut-down of industrial power plant gas turbine.
3. Detailed performance and stability robustness studies were carried out to design controllers to meet the control design specifications.

Publications Arising:

1. Omar, A. A. and Katebi, M R: Multi-layer Control Design for Gas Turbine Power Plants, Euraco Workshop II, Herrsching, Germany, pp 125-129, October, 1996.
2. Omar, A. A. and Katebi, M. R.: Optimisation of start-up and shut-down operations for power gas turbines. To be published in the Proceedings of Int. Conf. on Systems Engineering, Coventry, September, 1997.

Thesis Outlines:

The research of this thesis is organised as follows:

Chapter 1

This chapter presents the features and design aspects of Model Based Predictive Control (MBPC) strategies. Some of the commonly used process and disturbance models are described. Three well-known MPBC methods are presented and compared to demonstrate their similarities and differences.

Chapter 2

In this chapter the formulation of a multivariable unconstrained and constrained state space GPC algorithms are presented and the controller tuning parameters selection guidelines are summarised. Closed-loop formulation of the control design is presented and the performance and stability robustness issues are analysed. Tuning procedure for the controller is investigated by simulation of a multivariable process.

Chapter 3

A Dynamic model of a gas turbine power plant is presented. PID controllers are designed to stabilise the nonlinear model at a nominal operating point. A linearized model is derived and its order is reduced. The responses of the nonlinear and the linearized models are compared. Performance of the control design is investigated.

Chapter 4

This chapter is devoted to the design and the analysis of GPC controllers. A second layer predictive controller is designed for the gas turbine for to optimally move the set-points of the regulator loops to new operating conditions. Detailed analysis of GPC closed-loop system is carried out. The performance and the stability robustness are investigated through frequency domain indicators. A constrained predictive controller is designed to take into account process and actuator constraints and to investigate the possible improvements upon the unconstrained control design.

Chapter 5

This chapter is divided into two parts. Part one is concerned with the identification of linear models from the nonlinear model. One of the recently developed subspace state space algorithms is selected to identify the linear models. The second part is concerned with the design of an adaptive (gain scheduling) predictive controller. Both the constrained and the unconstrained adaptive controllers are designed to optimise the gas turbine operations.

Chapter 6

In this chapter, the main results of the research are summarised. Comments are made on the advantages and disadvantages of the techniques used in the thesis to optimise the operation of gas turbine power plants. Future work is also discussed.

Chapter One

Model Based Predictive Control

1.1 Introduction

In this Chapter some of the well-known Model Based Predictive Controllers (MBPC) algorithms will be reviewed. Their advantages and disadvantages will be stated. MBPC design aspects will be addressed. Some of the process and disturbance models that are commonly used in predictive control design will be reviewed. A comparison between three MBPC algorithms will be made to illustrate the differences and similarities among them and to select an algorithm to be used in later Chapter.

Model based control is a control strategy where, a model of the process is explicitly used to design a controller. The family of controllers belongs to this design strategy are called Model Based Predictive Controllers (MBPC) where, future process outputs and some times future process inputs are predicted over a finite horizon (range). This concept allows these controllers to deal with pre-assigned reference inputs.

Predictive control methods, also known as Long Range Predictive Control (LRPC) were firstly developed and introduced in France by Richalet et al., [8]. The first LRPC algorithm was called IDentification and COMmand (IDCOM) algorithm which has been known later as Model Algorithmic Control (MAC) [9]. In 1980 another algorithm was introduced in the United States by Cutler and Ramaker [10] and was called Dynamic Matrix Control (DMC) where, a Finite Step Response (FSR) model was used to describe the process. This design approach was successfully applied to many industrial applications such as large scale computer controlled petrochemical plants.

Other MBPC algorithms have been derived and applied successfully in process industry such as, Extended Horizon Adaptive Control (EHAC) by Ydstie [11], Extended Predictive Self Adaptive Control (EPSAC) by De Keyser and Van Cauwenberghe [12], and Generalized Predictive Control (GPC) algorithm which was developed by Clarke [1, 2].

1.2 Advantages and Disadvantages

Model Based Predictive Controllers have many features in common that make them attractive to both control engineers and plant operators [13]. Their features are summarised here:

- They are relatively simple to tune.

- They are capable of handling both Single-Input Single-Output (SISO) and Multi-Input Multi-Output (MIMO) processes.

- Process and actuator constraints can be handled in a systematic way during design stage. However, the ability of MBPC algorithm to handle constraints mainly depends on the quality of the predictor. The process model used for prediction must be valid irrespective of the choice of the data within the complete set. Also to produce sensible predictions it is required that the model used for prediction must be representative and also remains representative over the period of prediction [92].

- They have been applied to nonlinear processes [22, 23].

- Their predictive nature gives them the ability to deal with pre-programmed reference signals.

- It is easy to make the algorithm adaptive by including a suitable recursive identification algorithm.

- It can deal effectively with simple plants (open loop stable plants) as well as complex plants such as, nonminimum phase plants, open loop unstable plants, plants with unknown orders and with variable or unknown time delays [1, 2, 3, 13].

- Various filters and transfer functions can be added to the design to broaden the range of performance and to improve robustness against unmodelled dynamics and disturbances [4].

- Feed forward can be incorporated in predictive control design for compensation against the measurable disturbances [19].

- Some of the predictive controllers such as, DMC and GPC can provide feed forward action as a result of the feed forward term in their control laws [24]. This can reduce set-point interactions.

The disadvantage of MBPC is that a proper model of the process must be obtained. The other problem with predictive control is that, many controllers can be designed within the same framework. This makes it difficult to select which predictive controller is more suitable to solve a certain problem [13].

1.3 MBPC Design Aspects

The design of predictive controllers is based on the selection of the process model $M(\theta)$, the reference input w_k, the optimisation method, and the criterion function [4]. Selection of a proper linear model structure is important in designing the controller since, the prediction of the process output depends on the model. The process model is used to predict future process outputs $\hat{y}_{k+j}$ at time k using the past known inputs and measured process outputs. The predicted response is composed of two parts, free and forced responses. In general the model dynamics must be rich enough to include the essential process modes. The model should, of course, be controllable and observable [6].

Several process models are used with MBPC [6, 13]. One of the simplest process models is the Finite Impulse Response (FIR) model which has been used in Model Algorithmic Control (MAC) [9]. The other process model that found its application in predictive control is the Finite Step Response (FSR) Model which has been used in Dynamic Matrix Control (DMC) [10]. Transfer function model has some advantages over the above mentioned models thus, it is widely used in MBPC [6]. State space model is also widely used in predictive control. These models will be discussed later.

Disturbances should also be modelled to take into account its effect during output prediction. The disturbance term ζ_k is added at the input of the process. This disturbance may be deterministic or stochastic. Deterministic disturbance is exactly known a priori and can be described analytically or predicted. However, stochastic disturbances are random in nature and cannot be exactly described or predicted. They can be either stationary or non-stationary. The disturbances should therefore be approximately modelled.

To design a model based predictive controller an optimisation problem is solved over a finite horizon at each sampling time instant to find the optimum controller output sequence. The optimisation is applied to minimise a certain cost function or criterion function. The type of a cost function to be optimised is important because it determines whether it is easy or difficult to solve the problem. The cost functions based on design objectives such as rise time, settling time, overshoot and damping ratio are difficult to optimise due to highly nonlinear relationship between these objectives and the controller parameters. Thus a mathematically suitable criterion must be selected [13]. The criterion that is commonly used to design a predictive controller is of quadratic type. A simple form of the criterion is given by:

$$J(e,N2)=\sum_{j=0}^{N2-1}[e]^2=\sum_{j=0}^{N2-1}[\hat{y}_{k+j}-w_{k+j}]^2 \tag{1.1}$$

where, $N2$ is the maximum costing horizon (the period over which the predictions are made), $\hat{y}_{k+j}$ is the predicted process model output (free and forced responses), w_{k+j} is reference input j steps ahead and e is being the tracking error. The result of minimising the above cost gives a predictive control law (consists of the best future control signals) that minimises the tracking error between the predicted process output and the reference input [1]. Figure 1.1 shows a general block diagram of the predictive controller [6].

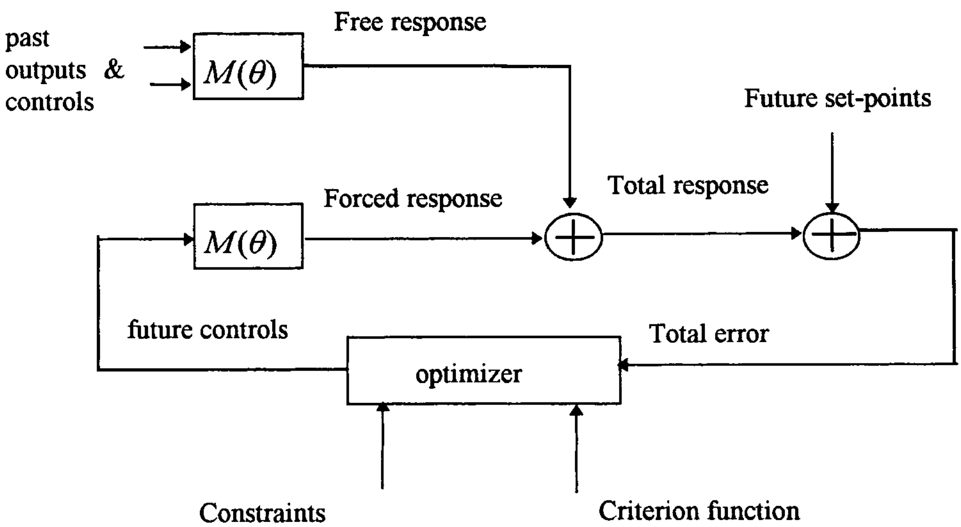


Figure 1.1: Block diagram of MBPC

1.4 Process and Disturbance Models

1.4.1 Process Models

To study or to control a process, a set of mathematical equations relating the process variables to each other, called a model, is required. A model may be either linear or non-linear.

In many cases model linearization around a certain desired operating point is required. Models of dynamic processes can be obtained theoretically by applying the laws of physics to describe the process or experimentally (system identification) using the data obtained by either of the following methods [14, 16]:

- Process transient response methods using different test signals, such as impulse and step test signals.

- Frequency response methods using sinusoidal excitation input at different frequencies.

- Identification using least squares theory.

A brief review of some of the models that are used in predictive control design is presented next their advantages and disadvantages are stated.

1.4.1.1 Finite Impulse Response Model

FIR model is characterised by its simple calculations, no restrictions about its order need to be made, and its effectiveness with plants that are open loop stable. It is described by the following equation:

$$y_k = \sum_{j=0}^{N_r-1} h_j u_{k-j-1} \tag{1.2}$$

where, h_j is the $j\,th$ impulse response element, N_r is the order of the model, u_{k-j-1} is the process input $j-1$ in the past, and y_k is the current process output. This model is unsuitable for unstable processes and requires many parameters to be estimated. It has been used in MAC [9].

1.4.1.2 Finite Step Response Model

This model is also often used in predictive control. It has been used in DMC and has

the same advantages and disadvantages as FIR model. FSR model has the following mathematical form:

$$y_k = \sum_{j=0}^{N_s-1} g_j \, \Delta u_{k-j-1} \tag{1.3}$$

where y_k is the current process output, g_j is the jth step response coefficient, N_s is the number of step response coefficients and Δ is the differencing operator $(1-q^{-1})$. For undisturbed processes the coefficients of the model are obtained by applying a step input to the process [10].

1.4.1.3 Transfer Function Model

Transfer function model is suitable for both stable and unstable processes, needs a minimum number of parameter to represent a process and it does not have the disadvantages of FIR and FSR models. It is described by:

$$y_k = q^{-d} \frac{B(q^{-1})}{A(q^{-1})} u_{k-1} \tag{1.4}$$

where d is the process time delay, $B(q^{-1})$ and $A(q^{-1})$ are the process polynomials in backward shift operator and u_{k-1} is the delayed input of the process. To describe a process by a transfer function an assumption about the order of the process must be made. The prediction of the process output described by this model is complex.

1.4.1.4 State Space Model

State space approach is a procedure to represent a process in time domain. It is valid for both SISO and MIMO systems and is widely used to analyse and to design control systems. Transformation methods from and to state space are available. The general form of state space representation is written as:

$$x_{k+1} = A x_k + B u_k + w_{wk} \tag{1.5}$$

$$y_k = C x_k + v_k \tag{1.6}$$

where x_k is the state vector of dimension $n \times 1$, A is an $n \times n$ state (system) matrix, B is an $n \times p$ input matrix, C is an output vector of dimension $r \times n$, w_{wk} and v_k are the white noise sequences.

1.4.2 Disturbance Models

Disturbances usually are inevitable in most real processes. They are one of the main reasons for using control. Moreover they are an important source of information about the nature of the properties of a system. They put fundamental limitations on the performance of a control system. The effects of the disturbances can be eliminated in many ways using feedback, feedforward or prediction [14].

The effect of measurable disturbances on a system response can be analysed and eliminated by generating and feeding it to the system in a closed-loop form. For non-measurable disturbances a disturbance model is required to allow a reasonable formulation of prediction to reduce the effect of the predictable part using feedforward technique.

One of the objectives of a control system is to compensate for the disturbance effects on the process output and since one of the advantages of predictive control is the simplicity of incorporating disturbance models thus, more attention will be paid here to examine different disturbance models. Two types of deterministic disturbances can be distinguished. Those which can be predicted exactly such as, offset (constant value), drift D_k ($D_k = d_0 + d_1 t + d_2 t^2 + \ldots + d_k t^k$),ramp, parabola,...etc., which are given by and those which cannot be predicted exactly [13, 15]. Drift disturbances can be represented in the following filtered form:

$$\zeta_{k1} = \frac{D_k}{A(q^{-1})} \tag{1.7}$$

Measurable disturbances can be represented as:

$$\zeta_{k2} = \frac{D(q^{-1})}{A(q^{-1})} vm_k \tag{1.8}$$

where, vm_k is a known measurable disturbance, the polynomials $A(q^{-1})$ and $D(q^{-1})$ constitute a discrete time transfer function (a filter) which acts on vm_k. The separation of this equation into past (known) and future (unknown) terms is realised by solving a Diophantine equation [13].

Stochastic disturbances can be separated into two unknown terms using Diophantine equations, future term which cannot be calculated and past term which can be calculated using the available data. The general form of these disturbances can be represented mathematically as follows [13, 15]:

$$\zeta_{k3} = \frac{C(q^{-1})}{A(q^{-1})} v_k \tag{1.9}$$

where, v_k is a discrete white noise sequence with zero mean and variance σ^2. $C(q^{-1})$ and $A(q^{-1})$ are polynomials which result in different stochastic disturbance models by assigning different settings for them. The above mentioned disturbance models can be combined in a single model which is valid to represent nearly most of disturbance signals:

$$\zeta_k = \frac{D_k}{A(q^{-1})} + \frac{D(q^{-1})}{A(q^{-1})} vm_k + \frac{C(q^{-1})}{A(q^{-1})} v_k \tag{1.10}$$

The stochastic disturbance represented by equation (1.9) when combined with the process model of equation (1.4) results in some of the well-known process models as summarised in Table 1.1.

Table 1.1 Stochastic process models.

Model	Poly. C	Poly. A	Description	Application to MBPC
MA	C	1	Moving Average (all zeros). It represents an FIR filter	MAC
ARMA	C	A	Auto Regressive and Moving Average (all poles and zeros). It is applied in many practical problems	EHAC and DMC
CARIMA	C	AΔ	Controlled Auto Regressive and Integrated Moving Average. It is widely used in processes where disturbances are random drifts.	EPSAC and GPC

Poly: Stands for polynomial

1.5 Review of Some MBPC Algorithms

Predictive controllers have the same framework and they follow the same concept (receding horizon) but their differences are as a result of the use of different process models, different prediction models, different performance indices and different reference inputs. Three well known predictive control methods namely, DMC, MAC and GPC will be summarised and their advantages and disadvantages will be stated. A comparison of different predictive control methods can be found in [7, 9, 13, 17].

1.5.1 Dynamic Matrix Control

Dynamic matrix control algorithm makes use of a Step Response model [10, 17] which is given by equation (1.3):

$$y_k = \sum_{i=1}^{\infty} g_i \Delta u_{k-i}$$

Including the disturbance (noise) term ζ_k, which represents the effects of the unmodelled dynamics and the disturbances acting on the process, in equation (1.3) then DMC model can expressed as follows:

$$y_k = \sum_{i=1}^{\infty} g_i \ \Delta u_{k-i} + \zeta_k \tag{1.11}$$

The j steps ahead output is given by:

$$y_{k+j} = \sum_{i=1}^{j} g_i \Delta u_{k+j-i} + \sum_{i=j+1}^{\infty} g_i \Delta u_{k+j-i} + \zeta_{k+j} \tag{1.12}$$

The first term (1.12) represents the forced response, the second term is the free response and the last term is the future disturbance.

For $\zeta_{k+j} = \zeta_k$ and from equation (1.11) and equation (1.12) the predicted output can be expressed as:

$$\hat{y}_{k+j} = \sum_{i=1}^{j} g_i \Delta u_{k+j-i} + \sum_{i=j+1}^{\infty} g_i \Delta u_{k+j-i} + y_k - \sum_{i=1}^{\infty} g_i \Delta u_{k-i} \tag{1.13}$$

which can be rewritten as:

$$\hat{y}_{k+j} = G_j(q^{-1}) \Delta u_{k+j} + p_j \tag{1.14}$$

where $G_j(q^{-1}) = g_1 q^{-1} + \cdots + g_j q^{-j}$, the first term of equation (1.14) is the forced response and p_j represents the free response of the system and is given by:

$$p_j = y_k + \sum_{i=1}^{\infty} (g_{j+i} - g_i) \Delta u_{k-i} \tag{1.15}$$

$$J = (w - \hat{y})^T (w - \hat{y}) + \lambda \Delta u^T \Delta u$$

$$J = (e_0 - G\Delta u)^T (e_0 - G\Delta u) + \lambda \Delta u^T \Delta u$$

$$J = \Delta u_k^T [G^T G + \lambda I] \Delta u_k - 2e_0^T G \Delta u_k + e_0^T e_0 \tag{1.16}$$

where $e_o^T = [w_{k+N1} - p_{N1}, \ldots\ldots, w_{k+N2} - p_{N2}]$, w_{k+j} is the future reference input, $\Delta u^T = [\Delta u(k),\ \Delta u(k+1),\ \cdots,\ \Delta u(k+Nu-1)]$ and λ is included to penalise the excessive increment of the incremental control action. N1 is the minimum output horizon, N2 is the maximum output horizon and Nu is the control horizon. The optimum control sequence resulted from minimising (1.16) is given by:

$$\Delta u(t) = [G^T G + \lambda I]^{-1} G^T e_o \tag{1.17}$$

where G is a lower triangle matrix of dimension $(N2 - N1) \times Nu$ which contains the step response parameters of the process.

The DMC algorithm has been used successfully in over 200 industrial applications [7]. Its main advantages are:

- Simple calculations are required to implement the process model and the algorithm.

- It is easy to understand.

- No precautions about the order of the process [7].

- It can be made adaptive.

- It is applied to multivariable processes.

- Constraints on the manipulated and controlled variables can be handled. However, the output constraints should be handled with care, that is safety margins must be introduced to prevent constraints violation.

Its disadvantage is that open loop unstable processes cannot be modelled or controlled.

1.5.2 Model Algorithmic Control (MAC)

Model Algorithmic Control (MAC) is suitable to control multivariable processes represented by their impulse responses [8, 9, 21]. Using a finite impulse response (FIR), equation (1.4), to build a model makes this algorithm valid only for stable processes.

In MAC the reference trajectory is represented by a first order model and computed over the prediction horizon $N2$ as:

$$w_{(k+i)mr} = \alpha\, w_{(k+i-1)mr} + (1-\alpha) w_k \tag{1.18}$$

where $i = 1, 2, \ldots\ldots, N2,\ \alpha = e^{-\tau/\tau_s}$, τ is the time constant of the first order reference model, τ_s is the sampling time, w_k is the set-point with the initial value of $w_{(k)mr} = y_k$ and the subscript mr refers to model reference. Predictions of the outputs j steps ahead in the future are computed from equation (1.2) and can be written as:

$$y_{(k+j)im} = \sum_{ii=0}^{N_{r-1}} h_{ii} u_{k+j-ii-1}$$

In a vector form:

$$y_{(k+j)im} = h^T u_{k+j} = u_{k+j}^T h \tag{1.19}$$

with:

$$h^T = [h(1)\ h(2) \cdots h(N_r)];$$

$$u_{k+j}^T = [u_{k+j-1}\ u_{k+j-2} \cdots u_{k+j-N_r}]$$

The subscript im refers to internal model.

The future inputs are assumed to remain unchanged that is:

$\Delta u_{k+j-1} = u_{k+j-1} - u_{k-1} = 0$ for $i=1, 2, \ldots, N2$

$$Y_{(k)im} = \begin{bmatrix} y_{(k+1)im} \\ y_{(k+2)im} \\ \vdots \\ y_{(k+N2)im} \end{bmatrix} = h^T u_k \tag{1.20}$$

To take into account the unknown disturbances the predicted outputs are modified to include the measured output y_k of the system over the prediction horizon N2:

$$Y^*_{im} = \begin{bmatrix} y^*_{(k+1)im} \\ y^*_{(k+2)im} \\ \vdots \\ y^*_{(k+N2)im} \end{bmatrix}, \text{ with}$$

$$y^*_{(k+i)im} = y_{(k+i)im} + [y_k - y_{(k)im}] \tag{1.21}$$

The prediction error is given by:

$$e = Y_{mr} - Y^*_{im}$$

For allowed changes in the input variable over the prediction horizon the prediction error can be expressed as:

$$e = Y_{mr} - Y^*_{im} - H\Delta\tilde{u} \tag{1.22}$$

where:

$$\Delta\tilde{u}^T = [\Delta u_k \ \Delta u_{k+1} \cdots \Delta u_{k+N2-1}]$$

$$H = \begin{bmatrix} h_1 & 0 & \cdots & 0 \\ h_2 & h_1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ h_{N2} & h_{N2-1} & \cdots & h_1 \end{bmatrix}$$

The quadratic cost function is given by :

$$J = e^T e + \lambda^2 \Delta\tilde{u}^T \Delta\tilde{u} \tag{1.23}$$

where λ is a penalising factor for the variations in the input signal. The optimal control sequence that minimises the error e, is obtained from the minimisation of the quadratic cost function with respect to $\Delta\tilde{u}$, is given by:

$$\Delta\tilde{u} = (H^T H + \lambda^2 I)^{-1} H^T (Y_{mr} - Y^*{}_{im}) \tag{1.24}$$

Only the first element of the control increments will be selected at each sample to give:

$$u_k = u_{k-1} + \Delta u_k$$

1.5.3 Generalised Predictive Control (GPC)

The input-output CARIMA model in a backward shift operator is given by [1, 5]:

$$A(q^{-1})y_k = B(q^{-1})u_{k-1} + C(q^{-1})v_k / \Delta \tag{1.25}$$

where polynomials $A(q^{-1})$ and $B(q^{-1})$ are the plant polynomials, q^{-1} is the backward shift operator, $C(q^{-1})$ is a user defined design polynomial serves as an observer, Δ is a differencing operator $(1-q^{-1})$ which ensures the integral action to cancel the effect of offset disturbance and v_k is a white noise disturbance with zero mean value. Equation (1.25) can be written as:

$$A(q^{-1})y_k = B(q^{-1})u_{k-1} + C(q^{-1})v_k / \Delta \tag{1.26}$$

Define a Diophantine equation as follows:

$$E_j(q^{-1})A(q^{-1})\Delta + q^{-j}F_j(q^{-1}) = C(q^{-1}) \tag{1.27}$$

where the polynomials $E_j(q^{-1})$ and $F_j(q^{-1})$ are uniquely defined and are of orders $j-1$ and n_f respectively. Multiplying equation (1.26) by $q^j\ E_j(q^{-1})\ \Delta$ gives:

$$\begin{aligned} q^j E_j(q^{-1})\Delta A(q^{-1})y_k &= q^j B(q^{-1})E_j(q^{-1})\Delta u_{k-1} \\ &+ q^j C(q^{-1})E_j(q^{-1})v_k \end{aligned} \tag{1.28}$$

Substitute equation (1.27) into equation (1.28) to get:

$$\begin{aligned} y_{k+j} &= E_j(q^{-1})\frac{B(q^{-1})}{C(q^{-1})}\Delta u_{k+j-1} + \frac{F_j(q^{-1})}{C(q^{-1})}y_k \\ &+ E_j(q^{-1})v_{k+j} \end{aligned} \tag{1.29}$$

where y_{k+j} is the plant output j steps ahead. Taking the expected value of both sides of the above equation and from the assumption that the disturbance term has a zero mean value, thus the last term is zero. This gives the following optimal predictor:

$$\hat{y}_{k+j} = F_j(q^{-1})\,/\,C(q^{-1})y_k + E_j(q^{-1})B(q^{-1})\,/\,C(q^{-1})\Delta u_{k+j-1} \tag{1.30}$$

$\hat{y}_{k+j}$ is a function of the measured current output y_k and future information in terms of future control inputs. This future information can be separated into past known and future signals using another Diophantine equation:

$$E(q^{-1})B(q^{-1}) = G_j(q^{-1})C(q^{-1}) + q^{-j}\Gamma_j(q^{-1}) \tag{1.31}$$

Hence:

$$\begin{aligned} \hat{y}_{k+j} &= G_j(q^{-1})\Delta u_{k+j-1} + \Gamma_j(q^{-1})\Delta u_k\,/\,C(q^{-1}) \\ &+ F_j(q^{-1})y_k\,/\,C(q^{-1}) \end{aligned} \tag{1.32}$$

$$\hat{y}_{k+j} = G_j(q^{-1})\,\Delta u_{k+j-1} + \hat{y}_{k+j/k} \tag{1.33}$$

where:

$\hat{y}_{k+j/k} = \Gamma_j(q^{-1})\Delta u_k \,/\, C(q^{-1}) + F_j(q^{-1})y_k \,/\, C(q^{-1})$

The term $\hat{y}_{k+j/k}$ represents the free response prediction. This equation has the following vector form:

$F=[\,\hat{y}_{k+1/k}\,,\ \hat{y}_{k+2/k}\,,\ \hat{y}_{k+3/k}\,, \ldots\ldots, \hat{y}_{k+N2/k}\,]$

Next define the vector of future control increments $\overline{U}$ (giving that $\Delta u_{k+j}=0$ for $j \geq Nu$):

$\overline{U}=[\Delta u_k\,, \Delta u_{k+1}\,, \Delta u_{k+2}\,, \Delta u_{k+3}\,, \ldots\ldots, \Delta u_{k+Nu}\,]$

Finally define the vector of the plant predicted outputs:

$\overline{Y}=[\,\hat{y}_{k+1}\,,\ \hat{y}_{k+2}\,,\ \hat{y}_{k+3}\,, \ldots., \hat{y}_{k+N2}\,]$

The prediction equation (1.33) can be written in a vector form:

$$\overline{Y}=G\overline{U}+F \tag{1.34}$$

where G is a matrix of dimension $N2 \times Nu$ and it is composed of the step response parameters g_i of the plant model $B\,/\,A\Delta$ and is given by:

$$G=\begin{bmatrix} g_0 & 0 & 0 & \ddots & 0 \\ g_1 & g_0 & 0 & \ddots & 0 \\ g_2 & g_1 & g_0 & \ddots & 0 \\ \ddots & \ddots & \ddots & \ddots & 0 \\ g_{N2} & g_{N2-1} & g_{N2-2} & \ddots & g_{N2-Nu} \end{bmatrix}$$

The cost function to be minimised is given by:

$$J_{GPC} = \sum_{j=N1}^{N2} [\hat{y}_{k+j} - w_{k+j}]^2 + \sum_{j=1}^{Nu} \lambda[\Delta u_{k+j-1}]^2 \tag{1.35}$$

Substituting equation (1.34) and equation (1.35) to get:

$$J_{GPC} = (\overline{Y}\text{-W})^T (\overline{Y}\text{-W}) + \lambda\, \overline{U}^{\,T}\, \overline{U} \tag{1.36}$$

This can be minimised with respect to the controller output sequence vector $\overline{U}$ to get the optimum controller output sequence:

$$\overline{U} = (G^T G + \lambda I)^{-1} G^T (W\text{-}F)$$

(2.19) This equation gives the optimal future control increments for times t up to *k+Nu-1*. At time t the first element of the vector $\overline{U}$ i.e., Δu_k is applied to the process. At time *k+1* the whole process is repeated using the information available. Then the process is repeated in the next time instants. This procedure is known as the receding horizon.

1.6 Comparison of the Methods

The strategy followed to derive these three methods can be summarised as follows: Each of these methods makes use of a process and disturbance model. The process models are used for output prediction. Future errors between the predicted outputs and future reference inputs are calculated over the assumed prediction horizons. A quadratic cost function is optimised to calculate a vector of optimal control signals that minimises the future tracking errors. It can be seen that these three algorithms follow the same strategy to derive the optimal control sequences. However, there are several differences among these algorithms:

- Each algorithm makes use of a different process and disturbance model that is, in MAC the process is described by its impulse response model, in DMC the process is described by a step response model and in GPC the process is described by a CARIMA model. The type of the model determines the type of the process that can be controlled. For instance unstable plants can not be modelled by finite responses.

- There is a difference in the way the disturbances are handled. In GPC the offsets due to load disturbances are eliminated by the assumed integral action whereas in MAC and DMC correcting the difference between the actual and the predicted outputs eliminates the offsets due to load disturbances.

- Different reference trajectories are used.
- The number of tuning parameters is different in the three algorithms.

The following comparison illustrates the differences between the three well-known MBPC algorithms and Table 1.2 summarises the main parameters of the cost functions for each algorithm [3, 13]:

Control Algorithm	N1	N2	Nu	λ
GPC	✓	✓	✓	✓
DMC	✓	✓	✓	✓
MAC	1	✓	Nu=N2	✓

Table 1.2: A summary of the tuning parameters of the three algorithms.

It can be seen that for MAC, N1 is fixed to one and Nu to N2, and so it has less tuning parameters than GPC and DMC. MAC is unsuitable for nonminimum-phase plants [3]. DMC is suitable for open-loop stable processes but open-loop unstable process cannot be controlled [7]. GPC is the only algorithm that can handle processes that are simultaneously nonminimum-phase and unstable [3]. Moreover many MBPC algorithms can be seen as GPC special cases. For instance, EHAC and EPSAC are its special cases for N1=N2=d>dl , d= the process time delay, dl= the process dead time, Nu=1 and $\lambda = 0$ and N1=1, N2=d>dl, Nu=1 and $\lambda = 0$, respectively. It can be seen tha GPC algorithtm can deal with a wider range of processes (complex processes).

1.7 Conclusions

Model Based Predictive Control algorithms have been successfully applied in the field of process industry as well as other applications such as robotics, aviation etc. These algorithms share common ideas such as receding horizon principle, minimisation of a certain cost function, and the use of a process model.

In this Chapter the advantages and disadvantages of MBPC algorithms were summarised and some of the commonly used process and disturbance models were discussed. Three of the well-known MBPC algorithms were discussed and their advantages and disadvantages were stated. The algorithms were compared to illustrate the differences and similarities between them. It was concluded that GPC algorithm has more advantages over other MBPC algorithms in the sense that it can deal with wider range of processes namely, those which are simultaneously open-loop unstable and nonminimum-phase. This algorithm will be discussed in details in the next Chapter.

Chapter Two

Robust Multivariable Predictive Control

2.1 Introduction

GPC algorithm is characterised by its simplicity, flexibility (due to its large number of tuning and design parameters), its applicability to a wide range of processes, simple as well as complex ones, and within its formulation many other algorithms can be obtained as its special cases. This has made GPC popular and it has found many applications in process industry.

GPC algorithms are readily formulated both in polynomial and state space representations for SISO and MIMO systems. The first GPC algorithm due to Clarke and his co-workers [1, 2] was formulated for SISO systems in polynomial form using CARIMA process model. Based on a multivariable CARIMA process model multivariable Diophantine equations were solved to derive a multivariable control algorithm [18].

A SISO state space description for GPC controllers was presented by Ordys and Clarke [20]. The assumed state space process model includes, the increment of the control signal Δu, to provide integral action, and the disturbance term v_k which affects future state and current output of the process is assumed to be white Gaussian noise with zero mean value and variance σ^2. The extension of SISO state space GPC algorithm to a multivariable algorithm is also considered [20, 25, 26].

A state space representation of a process that could be either minimal or non-minimal is not unique that is, within a similarity transformation several descriptions can be derived. Several forms of state space models are discussed in [20]. State space representation has some advantages over the polynomial descriptions [27].

- The extension of SISO state space systems to MIMO systems is straight forward.
- State space models are more suitable for digital computers.
- State space equations are first order differential equations. This makes them simple to manipulate.
- Multivariable systems and advanced optimal and robust control techniques are usually formulated in state space and use state space based algorithms for their numerical solutions.
- Time varying and nonlinear systems are more naturally represented in state space.

For these reasons, the multivariable GPC algorithm that will be discussed here is formulated in state space and hence the analysis and the design that will follow will be also in state space throughout the following Chapters. Both the unconstrained and constrained GPC algorithms will be discussed. The constraints that have been presented in [29] will be extended for the multivariable state space GPC algorithm. GPC closed-loop formulation will be used to derive conditions for performance and stability robustness of the closed-loop system. Simulation example of a multivariable GPC controller will be presented to demonstrate the tuning procedure of the controller.

2.2 Multivariable State Space GPC Algorithm Formulation

State space GPC algorithms for both SISO and MIMO processes have been presented in [20, 25, 26]. The algorithm that is discussed here is based on the formulation given in [26]. The first step in deriving a GPC algorithm is to select a suitable process model, the second is to construct an optimal output predictor, the third is to calculate a vector of state estimates using a state estimator, this is needed in the case of state space formulation, and finally, a quadratic objective function should be minimised to calculate the control law.

2.2.1 Process Model

The assumed discrete-time state space process model is the general state space model given by:

$$x_{k+1} = Ax_k + B\Delta u_k + Ew_{wk} \tag{2.1}$$

$$y_k = Cx_k + D\Delta u_k + v_k \tag{2.2}$$

where A is $(n \times n)$ system matrix, B is $(n \times p)$ input matrix, C is $(r \times n)$ output matrix, D is $(r \times p)$ direct transmission matrix, E is the $(n \times n)$ input disturbance matrix, w_{wk} is the process noise which includes all errors due to modelling and the stochastic disturbances, v_k is being the output measurement noise vector. An integral action is being added to the general state space model by using the control increment $\Delta u_k = u_k - u_{k-1}$ in place of the control signal u_k. The integral action affects the state matrix A and makes some of the eigenvalues of A equal to one [20]. The assumed noises w_{wk} and v_k are uncorrelated and modelled as Gaussian distributed, white, stationary random processes with the following properties:

$$E\{w_{wk}\} = 0, \qquad E\{w_{wi}w_{wk}^T\} = \begin{cases} 0 & i \neq k \\ Q_w & i = k \end{cases}$$

$$E\{v_k\} = 0, \qquad E\{v_i v_k^T\} = \begin{cases} 0 & i \neq k \\ R_v & i = k \end{cases} \tag{2.3}$$

$$E\{w_{wk} v_k{}^T\} = 0$$

where $E(.)$ is being the expectation operator, Q_w is the process noise covariance matrix and R_v is the measurement noise covariance matrix.

2.2.2 The Optimal Output Predictor

To formulate the GPC algorithm, an output predictor is required. This is done first, by constructing a j-step ahead output signal through recursive substitution of the state equation (2.1) into the output equation (2.2) for j times to give:

$$\begin{aligned} y_{k+j} &= CA^j x_k + \sum_{i=0}^{j-1} CA^{(j-i-1)} B\Delta u_{k+i} + D\Delta u_{k+j} \\ &+ \sum_{i=0}^{j-1} CA^{(j-i-1)} Ew_{wk+i} + v_{k+j} \end{aligned} \tag{2.4}$$

Then, by taking the expected value of equation (2.4) an optimal output predictor is obtained:

$$\begin{aligned} \hat{y}_{k+j} &= E\{y_{k+j} / y_k\} \\ &= CA^j E(x_k / y_k) \\ &+ \sum_{i=0}^{j-1} CA^{(j-i-1)} B\Delta u_{k+i} + D\Delta u_{k+j} \\ &+ \sum_{i=0}^{j-1} CA^{(j-i-1)} E\{E(w_{wk+i})\} + E(v_{k+j}) \end{aligned} \tag{2.5}$$

This predictor can be simplified using the noise properties given in equations (2.3) and denoting $\hat{x}_k = E(x_k / y_k)$:

$$\hat{y}_{k+j} = CA^j \hat{x}_k + \sum_{i=0}^{j-1} CA^{(j-i-1)} B\Delta u_{k+i} + D\Delta u_{k+j} \tag{2.6}$$

$\hat{x}_k$ is the state vector estimate. This can be obtained using a steady state Kalman filter.

2.2.3 Kalman Filter

Kalman filter is an optimal state observer suitable for stochastic processes. The discrete time Kalman filter that is used here for state estimation is a linear stationary filter. It requires the state and the observation equations to be linear in state to obtain the state estimate covariance equations. Kalman filter is designed to minimise the variance of the error e_k between the true state x_k and the estimated state $\hat{x}_k$. Besides state estimation Kalman filter can be interpreted as a disturbance filter like the T-polynomial filter given in [1, 2] for polynomial GPC design. Moreover the use of Kalman filter leads to a systematic design procedure for any disturbance model even in the case of unstable processes [25]. The steady state Kalman filter estimator is given by:

$$\hat{x}_{k+1} = A\hat{x}_k + B\Delta u_k + M[y_k - C\hat{x}_k - D\Delta u_k] \tag{2.7}$$

M represents the Kalman gain matrix which adapts the estimated model states to the measured process outputs.

$$M = APC^T[CPC^T - R_v]^{-1} \tag{2.8}$$

where R_v is the measurement noise covariance matrix and P is the unique symmetric non-negative definite solution of a discrete-time algebraic Riccati equation given by:

$$[A - MC]P[A - MC]^T - P + GQ_wG^T + MRM^T = 0 \tag{2.9}$$

MATLAB control toolbox provides a solution to the problem of steady state Kalman filter.

2.2.4 Cost Function and The Optimal Control Law

The cost function is of the following quadratic form:

$$
\begin{aligned}
J = &\sum_{j=N1}^{N2} \left[w_{k+j} - \hat{y}_{k+j}\right]^T \gamma \left[w_{k+j} - \hat{y}_{k+j}\right] \\
&+ \sum_{j=0}^{Nu-1} \left[\Delta u_{k+j}\right]^T \lambda \left[\Delta u_{k+j}\right]
\end{aligned}
\tag{2.10}
$$

where $N1$ is the minimum output horizon, $N2$ is the maximum output horizon, Nu is the control horizon, γ is the weighting on the tracking error, and λ is the weighting on the control increments Δu_k. This cost function includes two terms, the first one is the summation of the weighted squares of the tracking error over the horizon N1 to N2 and the second term is the summation of the weighted squares of the control increments. The control increments Δu_{k+j} are assumed to be constant for $j \geq Nu$.

The vector representation of the cost function is,

$$J = (W - \hat{y})^T \Gamma (W - \hat{y}) + \Delta\tilde{u}^T \Lambda \Delta\tilde{u} \tag{2.11}$$

where,

$W = [w_{k+N1}^T, \cdots, w_{k+N2}^T]^T$, W is a vector of future set-points,

$\Delta\tilde{u} = [\Delta u_k^T, \cdots, \Delta u_{k+Nu-1}^T]^T$, $\Delta\tilde{u}$ is the vector of control increments and

$\Delta u^T = [\Delta u_{k1}, \Delta u_{k2}, \cdots, \Delta u_{kp}]$

Δu_{kp} is the control increment of the pth control input.

The output predictor in a vector form is given by:

$$\hat{y} = F\hat{x} + Hm\Delta\tilde{u} \tag{2.12}$$

$\hat{y} = [\hat{y}_{k+N1}^T, \cdots, \hat{y}_{k+N2}^T]^T$

$$\hat{y}_{k+j}^T = [\hat{y}_{(k+j)1}, \cdots, \hat{y}_{(k+j)r}] \quad j = N1, N2$$

$\hat{y}_{(k+j)r}$ is the rth predicted output and the first term of the output predictor is the free response and the second is the forced response,

$$F = \begin{bmatrix} CA^{N1} \\ \vdots \\ \vdots \\ CA^{N2} \end{bmatrix}, \quad Hm = \begin{bmatrix} h_{N1-1} & \cdots & \cdots & h_{N1-Nu} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ h_{N2-1} & \cdots & \cdots & h_{N2-Nu} \end{bmatrix}, \quad h_i = \begin{cases} CA^i Bm & i \geq 0 \\ Dm & i = -1 \\ 0 & i < -1 \end{cases}$$

Hm contains the step response coefficients for $i \geq 0$, Γ and Λ are diagonal weighting matrices as follows:

$$\Gamma = \begin{bmatrix} \gamma & & 0 \\ & \ddots & \\ 0 & & \gamma \end{bmatrix}, \text{ where } \quad \gamma = \begin{bmatrix} \gamma_1 & & 0 \\ & \ddots & \\ 0 & & \gamma_r \end{bmatrix}$$

$$\Lambda = \begin{bmatrix} \lambda & & 0 \\ & \ddots & \\ 0 & & \lambda \end{bmatrix}, \text{ where } \quad \lambda = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_p \end{bmatrix}$$

γ_r is the weighting on the rth tracking error and λ_p is the weighting on the pth control increment.

Now we can substitute equation (2.12) into (2.11) to get:

$$\begin{aligned} J &= \Delta\tilde{u}^T [Hm^T \Gamma Hm + \Lambda]\Delta\tilde{u} - 2(W - F\hat{x})^T \Gamma Hm \Delta\tilde{u} \\ &\quad + (W - F\hat{x})^T \Gamma (W - F\hat{x}) \end{aligned} \tag{2.13}$$

Minimise with respect to $\Delta\tilde{u}$:

$$\frac{dJ}{d\Delta\tilde{u}} = 2[Hm^T \Gamma Hm + \Lambda]\Delta\tilde{u} - 2Hm^T \Gamma (W - F\hat{x}) = 0$$

This leads to the following optimal control law:

$$\Delta\tilde{u} = [Hm^T \Gamma Hm + \Lambda]^{-1} Hm^T \Gamma (W - F\hat{x}) \tag{2.14}$$

Only the first element of the vector $\Delta\tilde{u}$ is used to obtain the control signal:

$$u_k = u_{k-1} + \Delta u_k$$

In the next time step the procedure is repeated. This leads to open loop feedback control action. GPC control law equation (2.14) contains two terms. The first term can be considered as a feedforward part and the second term as a state feedback part.

2.3 GPC Tuning Parameters Selection Rules

The guidelines of choosing the tuning parameters are summarised here:

- $N1$: The minimum (lower) output horizon and is associated with the process dead time dl. When the value of the dead time is known $N1$ set equal to $dl+1$ otherwise $N1$ is set to one. Higher values of $N1$ reduces the amount of computations and results in smoother control action.

- $N2$: The maximum (upper) output horizon is recommended to be equal to the plant rise time [3]. In case of nonminimum-phase plant, which has an initial negative going response, the value of $N2$ is chosen such that some of the positive going response samples are included in the costing. For stable processes, when $N2$ is chosen high enough (equals to the process settling time), it gives a closed-loop response speed nearly equal to that of the open-loop.

- Nu: The control horizon is set to one for simple plant. For complex plants the value of Nu is set at least equal to the number of unstable poles or the poles near the stability boundary. Increased values of Nu results in a faster control action and hence more active manipulated variable.

- λ: The control costing (control-weighting) penalises the control increments. It prevents the output signal from increasing and is used for fine tuning the damped control modes. For numerical stability reasons the values of λ are recommended to be chosen very small [26].

- γ_i : The weighting on the tracking errors. It is recommended to have values in the range [62]:

$$0 < \gamma_i \leq 1, \qquad i = 1,2,....,r.$$

with r refers to the rth output.

2.4 Constrained GPC

The real time processes are subject to constraints. The process variables should stay within specified boundaries (limits), due to design requirements, physical limitations and safety regulations. For instance, in certain plants it may be needed to have some variables such as, temperature, power, pressure, level, etc., to be governed to pre-specified values. In this case, the controller must be capable of handling such constraints. GPC algorithms having these capabilities have been derived and applied to processes with different types of constraints. The general optimisation problem can be stated as follows.

Optimisation is either a minimisation or a maximisation problem. Solving an optimisation problem usually requires an iterative procedure. However if the objective function, to be minimised or maximised, is quadratic then an analytical solution is available. A quadratic objective function $J(u, e)$, where u is the process input and e is the tracking error, can be minimised when there is no constraints by equating its first derivative (e.g. with respect to u) to zero. On the other hand if there exist constraints on the process variables the optimisation of the objective function is no longer analytically possible and an iterative procedure is required (search method).

When there is no constraints GPC optimisation problem is solved analytically as already discussed in Section 2.2.4. However, in the presence of constraints the problem is difficult to solve analytically and numerical optimisation methods are required. Several works dealing with constrained GPC problem have been presented over the past ten years. Tsang and Clarke [30] have derived GPC algorithm with input constraints (amplitude and rate constraints) by minimising a multi-stage cost function using a Quadratic Programming, (QP) method. They stated that other constraints, such as output and state constraints, can also be handled by their algorithm. Their algorithm is suitable for a wide range of low speed control application (i.e. chemical industry). However, it is not suitable for fast systems due to the large amount of computations required by this method. De Madrid et al., [91] have suggested the application of dynamic programming method to solve GPC constrained problem in order to reduce the computational effort required during the optimisation process. Kuznetsov and Clarke [29] have minimised GPC cost function subject to four different types of constraints namely, constraints on control increments, constraints on control signal amplitude, constraints on overshoot and constraints on the outputs as a problem of quadratic programming optimisation.

2.5 Constrained GPC Formulation

Violation of constraints may result in poor performance of the control system, loss of production and equipment damage. Thus, constraints must be taken into consideration during design stage of the control system and their violation must be predicted and corrected by the control system [28]. Amplitude limitations on control signals, slew rate limits of actuator (limitations on control increments), limitations on the output signals (normally imposed for safety reasons) and limiting overshoot are the most commonly types of constraints experienced in process industry. The four constraints given in [29] are extended here to the multivariable state space GPC algorithm:

1. **Constraints on control signal**: The amplitude of the current and future control signals can be limited to upper u_{max} and lower u_{min} limits. It follows that,

$$\sum_{j=1}^{i} \Delta u_{k+j-1} = u_{k+j-1} - u_{k-1} \tag{2.15}$$

where,

$$\Delta u_{k+j-1} = [\Delta u_{(k+j-1)1}, \ldots, \Delta u_{(k+j-1)p}]^T$$

$$u_{k+j-1} = [u_{(k+j-1)1}, \ldots, u_{(k+j-1)p}]^T$$

$$u_{k-1} = [u_{(k-1)1}, u_{(k-1)2}, \ldots, u_{(k-1)p}]^T$$

The upper and the lower limits on the control signal, over the horizon $j = N_{lu}, N_{uu}$ with N_{lu} and N_{uu} represent the upper and lower constraint horizons, are expressed as:

$$u_{\min} \leq u_{k+j-1} \leq u_{\max} \tag{2.16}$$

$$u_{\max} = [u_{(k+j-1)\max 1}, \ldots, u_{(k+j-1)\max p}]^T$$

$$u_{\min} = [u_{(k+j-1)\min 1}, \ldots, u_{(k+j-1)\min p}]^T$$

Equation (2.16) can be written in terms of the control increments Δu_{k+j-1} as follows:

$$u_{\min} - u_{k-1} \leq \sum_{j=1}^{i} \Delta u_{k+j-1} \leq u_{\max} - u_{k-1} \tag{2.17}$$

In vector form:

$$U_{\min} \leq \overline{U} \leq U_{\max} \tag{2.18}$$

$$U_{\min} = [(u_{\min} - u_{k-1})^T, \ldots, (u_{\min} - u_{k-1})^T]^T$$

$$\overline{U} = [\Delta u_{k+N_{lu}-1}^T, \ \ldots, \ \Delta u_{k+N_{uu}-1}^T \ ]^T$$

$$U_{\max} = [(u_{\max} - u_{k-1})^T, \ldots, \ (u_{\max} - u_{k-1})^T]^T$$

Hence, the constraints on the control signals as inequalities are:

$$L\overline{U} \leq U_{\max}, \quad L\overline{U} \leq -U_{\min} \tag{2.19}$$

L is a lower triangular matrix given by:

$$L = \begin{bmatrix} \phi & 0 & . & 0 \\ \phi & \phi & . & 0 \\ . & .. & . & . \\ \phi & \phi & . & \phi \end{bmatrix}, \ \phi = I_{p \times p} \tag{2.20}$$

where I is an identity matrix of dimension $p \times p$, p represents the number of inputs, and L is of dimension $(p \times Nu) \times (p \times Nu)$ for $N_{lu} = 1$ and $N_{uu} = Nu$.

2. **Constraints on control increments**: Each control increment of Δu_{k+j-1} can be limited to a maximum $\Delta u_{\max}$ and a minimum $\Delta u_{\min}$ limit over a lower $N_{l\,\Delta u}$ and an upper $N_{u\Delta u}$ horizons thus we can write:

$$\Delta u_{\min} \leq \Delta u_{k+j-1} \leq \Delta u_{\max} \qquad \text{for } j = N_{l\Delta u}, N_{U\,\Delta u} \tag{2.21}$$

$$\Delta u_{\max} = [\Delta u_{(k+j-1)\max 1}, \ldots, \Delta u_{(k+j-1)\max p}]^T$$

$$\Delta u_{\min} = [\Delta u_{(k+j-1)\min 1}, \ldots, \Delta u_{(k+j-1)\min p}]^T$$

Let,

$$\overline{U} = [\Delta u_{k+N_{l\Delta u}-1}^T, \ldots, \Delta u_{k+N_{u\Delta u}-1}^T]^T$$

$$\overline{U}_{\max} = (\Delta u^T{}_{\max},\ldots,\Delta u^T{}_{\max})^T$$

$$\overline{U}_{\min} = (\Delta u^T{}_{\min},\ldots,\Delta u^T{}_{\min})^T$$

The constraint on control increments can be expressed by the following inequalities:

$$\mathrm{I}\overline{U} \le \overline{U}_{\max} \text{ and } -\mathrm{I}\overline{U} \le -\overline{U}_{\min} \tag{2.22}$$

where I is a diagonal matrix of dimensions:

$$(p \times (N_{u\Delta u} - N_{l\Delta u})) \times (p \times (N_{u\Delta u} - N_{l\Delta u})).$$

3. **Output constraints**: The outputs of a process can be limited by introducing upper and lower output constraints $y_{\max}$ and $y_{\min}$. The output constraints have the following form.

$$y_{\min} \le y_{k+j} \le y_{\max}, \quad \text{over the horizon } j = N_{ly},\ N_{uy} \tag{2.23}$$

$$y_{\max} = [y_{(k+j-1)\max 1},\ldots,\ y_{(k+j-1)\max r}]^T$$

where N_{ly} and N_{uy} are the lower and the upper constraint horizons for the output. Equation 2.23 will be met only if we have a perfect model of the process to be able to replace y_{k+j} with the optimal output predictior $\hat{y}_{k+j}$.It is assumed here that the modelling errors are negligible. Thus, Substituting from the prediction equation (2.12) for y_{k+j}, and using vector notations, results in:

$$-Hmy\overline{U} \le -Y_{\min} + Fy\hat{x}, \quad Hmy\overline{U} \le Y_{\max} - Fy\hat{x}, \tag{2.24}$$

$$Y_{\max} = [y^T{}_{\max},\ \ldots,\ y^T{}_{\max}]^T$$

$$Y_{\min} = [y^T{}_{\min},\ldots,y^T{}_{\min}]^T$$

where Hmy is a submatrix of Hm and Fy is a submatrix of F, with dimensions $(r)(N_{uy} - N_{ly}) \times (p)(Nu)$ and $(r)(N_{uy} - N_{ly}) \times (n)$ respectively, and are given by:

$$Hmy = \begin{bmatrix} h_{N_{ly}-1} & \cdots & \cdots & h_{N_{ly}-Nu} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ h_{N_{uy}-1} & \cdots & \cdots & h_{N_{uy}-Nu} \end{bmatrix}, \quad h_i = \begin{cases} CA^i Bm & i \geq 0 \\ Dm & i = -1 \\ 0 & i < -1 \end{cases}$$

$$Fy = \begin{bmatrix} CA^{Nly} \\ \vdots \\ \vdots \\ CA^{Nuy} \end{bmatrix}$$

4. **Overshoot constraint**: Undesired overshoots in process outputs can be limited by introducing overshoot constraints. For the set-point w greater than the current output y_k and for the set-point w less than the current output y_k, a set of constraints can be written respectively as:

$$y_{k+j} \leq w, \qquad j = N_{lo}, N_{uo} \qquad 1 \leq N_{lo} \leq N_{uo} \tag{2.25}$$

$$-y_{k+j} \leq -w, \qquad j = N_{lo}, N_{uo} \qquad 1 \leq N_{lo} \leq N_{uo} \tag{2.26}$$

where N_{lo} and N_{uo} are the lower and the upper constraint horizons for overshoot. Using the prediction equation (2.12) to replace for y_{k+j} we get:

$$Hmo\overline{U} \leq w_{\max} - Fo\hat{x} \qquad \text{for } w > y_k \tag{2.27}$$

$$-Hmo\overline{U} \leq -w_{\max} + Fo\hat{x} \qquad \text{for } w < y_k \tag{2.28}$$

where Hmo is a submatrix of the matrix Hm and Fo is a submatrix of F, with dimensions $(r)(N_{uo} - N_{lo}) \times (p)(Nu)$ and $(r)(N_{uo} - N_{lo}) \times (n)$ respectively, and are given by:

$$Hmo=\begin{bmatrix} h_{N_{lo}-1} & \cdots & \cdots & h_{N_{lo}-Nu} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ h_{N_{uo}-1} & \cdots & \cdots & h_{N_{uo}-Nu} \end{bmatrix}, \quad h_i=\begin{cases} CA^i Bm & i \geq 0 \\ Dm & i=-1 \\ 0 & i<-1 \end{cases}$$

$$Fo=\begin{bmatrix} CA^{Nlo} \\ \vdots \\ \vdots \\ CA^{Nuo} \end{bmatrix}$$

All the inequality constraints can be expressed in the following compact form:

$$\mathrm{N}\Delta U \leq \mathrm{g} \tag{2.29}$$

$$\mathrm{N}=(I \;\; \text{-}I \;\; L \;\; \text{-}L \;\; Hmy \;\; -Hmy \;\; \pm Hmo)^{\mathrm{T}} \tag{2.30}$$

$$\mathrm{g}=(\overline{U}^T \;\; \text{-}\overline{U}^T \;\; \overline{U}^T_{\max} \;\; \text{-}\overline{U}^T_{\min} \;\; (Y_{max}\text{-}Fy\hat{x})^T \;\; (\text{-}Y_{min}+Fy\hat{x})^T \;\; \pm(w_{max}\text{-}Fo\hat{x})^{\mathrm{T}})^{\mathrm{T}} \tag{2.31}$$

When the cost function equation (2.13) is combined with the constraints (2.29) and (2.30), a nonlinear quadratic optimisation problem is formulated:

$$\min_{\Delta U}\left\{ \Delta U^T H \Delta U + h^T \Delta U \right\} \text{ subject to } \mathrm{N}\Delta U \leq \mathrm{g} \tag{2.32}$$

$$H=[Hm^T \Gamma Hm + \Delta] \tag{2.33}$$

$$\text{and } h^T = -2(w-f)^T \Gamma Hm \tag{2.34}$$

This optimisation problem can be solved using a QP algorithm. Other constraints such as, non-minimum phase behaviour constraints, actuator nonlinearities constraints and terminal state constraints [18] can be taken into consideration during the optimisation of GPC cost function. As the number of constraints increases the amount of computations required for the optimisation process will also increase.

2.6 Properties of GPC

Properties of a control system can be described in terms of how well the controller deals with the unwanted signals (the control system performance) and how good it regulates changes in, parameter variations, internal behaviour of the controlled system (the control system robustness) [14]. A controller must have the ability to handle a wide range of problems that are commonly found in process industry such as unmeasurable disturbances, measurement noise, interactions from one loop to another, actuator limitations, process dead times, time delays, nonminimum-phase behaviour, unmodelled process dynamics, etc. There is no specific controller that can handle or solve all these problems at the same time. However, it is possible to design a controller that can handle many of these problems efficiently. Properties of GPC have been examined in many works [1, 2, 3, 18, 31, 32, 33] and some stability theorems for special cases of GPC have been presented [2, 3]. Closed-loop Analysis is used to investigate the controller stability and the way in which it can be improved and how disturbance models can be added to improve the regulation of the controller performance [32, 33, 34].

2.6.1 GPC Performance and Stability Robustness

A robust control system is a system that has a good disturbance rejection, gives good regulation in the presence of disturbance, and low sensitivity to internal changes of plant parameters. Also a control system is said to be robust if it has adequate stability margins. It is important that a control system to be robust, with respect to measurement error, plant disturbances, and modelling errors. A control strategy that is commonly used to reduce the effects of disturbances, parameter uncertainties, and unpredictable changes in the behaviour of a controlled system is feedback technique. Model uncertainties are due to process model approximation. A control system must be robust against these model uncertainties because processes usually are described by their approximate models especially those used for control purposes. The robustness of GPC to model over- and under-parametrization and fast sampling has been investigated in [2, 34]. The desired speed of response is achieved by adjusting different GPC tuning parameters. Simulation examples and experimental results have confirmed the theoretical analysis and proved the ability of GPC to yield a consistent closed-loop behaviour for large process variations [32]. Closed-loop formulation and analysis of input output GPC have been carried out by Clarke *et al.*, [2, 3]. They studied the effects of different tuning parameters, user specified polynomials (pre-filter polynomials), over- and under-parametrization and unknown or variable time delay on the closed-loop performance. Robinson and Clarke [33] examined the effects of the pre-filter polynomials $T(q^{-1})$ and $P(q^{-1})$ on the closed-loop stability and robustness of two special cases of GPC (i.e. mean level and dead beat GPC) in the presence of model mismatch.. Banerjee *et al.*, [34] examined the effects of tuning parameters, model mismatch, and filter polynomials on the performance and stability as well as the robustness of GPC in frequency domain using small signal theorem. In the above mentioned analysis it is obvious that GPC algorithm has a reasonable closed-loop performance and robustness. A new performance and stability robustness method for multivariable state space GPC will be presented in the next Section. The method is similar to the one used for H_∞ control design and analysis.

2.6.2 Multivariable State Space GPC Closed-Loop Formulation

In this section the closed-loop formulation of GPC is presented. Frequency domain indicators are derived to assess the system stability and performance robustness and to tune Kalman filter such that the robustness is improved. The linear state space quadruple (A, B, C, D) of a process is used for designing both GPC and Kalman filter state estimator. The closed-loop linear system model is shown in Figure 2.1.

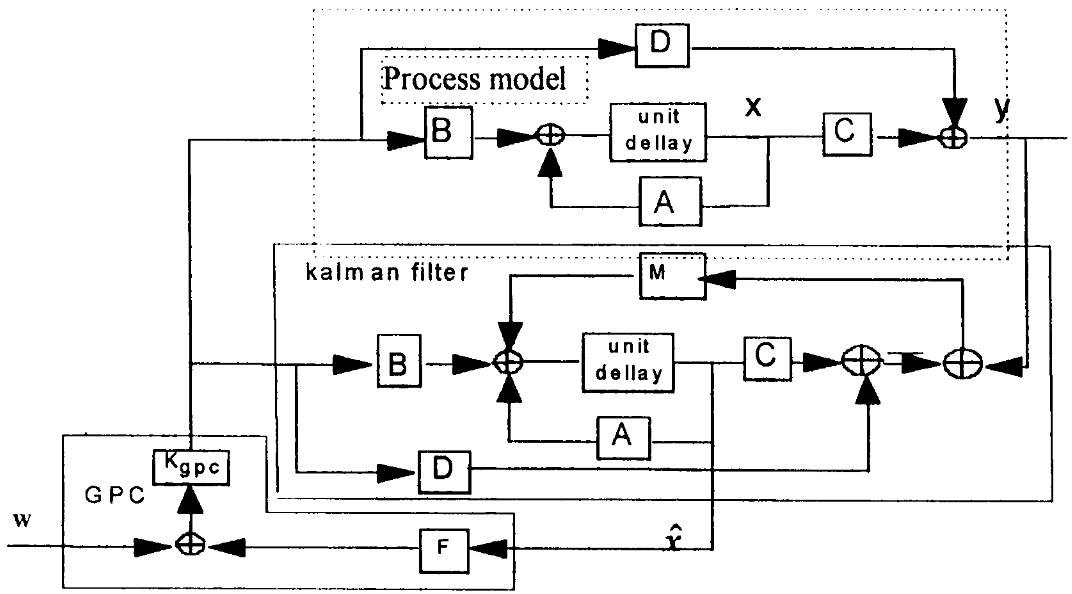


Figure 2.1: GPC closed-loop linear model.

For zero initial conditions equation (2.7) can be written in z-domain as:

$$z\hat{x}(z) = A\hat{x}(z) + B\Delta u(z) + M(y(z) - C\hat{x}(z) - D\Delta u(z)) \tag{2.35}$$

$$\{Iz - (A - MC)\}\hat{x}(z) = (B - MD)\Delta u(z) + My(z)$$

$$\hat{x}(z) = [Iz - (A - MC)]^{-1}(B - MD)\Delta u(z) + [Iz - (A - MC)]^{-1} My(z) \tag{2.36}$$

$$K_{\hat{x}u} = \hat{x}(z) / \Delta u(z)\big|_{y(z)=[0]} = [Iz - (A - MC)]^{-1}(B - MD)$$

Define the quadruple model of Kalman filter as:

$$K_{\hat{x}u} = \begin{pmatrix} A - MC & B - MD \\ I & 0 \end{pmatrix} \tag{2.37}$$

$$K_{\hat{x}y} = \hat{x}(z) \big/ y(z)\big|_{\Delta u(z)=[0]} = [Iz - (A - MC)]^{-1} M$$

Define,

$$K_{\hat{x}y} = \begin{pmatrix} A - MC & M \\ I & 0 \end{pmatrix} \tag{2.38}$$

$K_{\hat{x}u}$ and $K_{\hat{x}y}$ are the system matrices that represent the effects of inputs and outputs on the Kalman filter.

The process transfer function is defined as:

$$G_c = C[Iz - A]^{-1}B + D$$

or in a quadruple form:

$$G_C = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \tag{2.39}$$

The matrices K_{gpc} and F are constant matrices representing GPC gain matrix and F can be considered as a state feed back gain matrix.

$$K_{gpc} = \begin{pmatrix} 0 & 0 \\ 0 & K_{gpc} \end{pmatrix} \tag{2.40}$$

$$F = \begin{pmatrix} 0 & 0 \\ 0 & F \end{pmatrix} \tag{2.41}$$

Using the new definitions for the process and the control transfer functions, the model of Figure 2.1 can be simplified to the one shown in Figure 2.2.

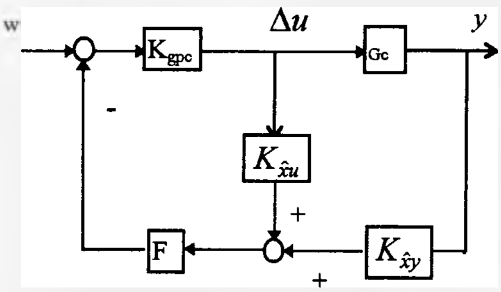


Figure 2.2: GPC closed-loop system block diagram.

This system can be reduced to the general feedback form as shown in Figure 2.3.

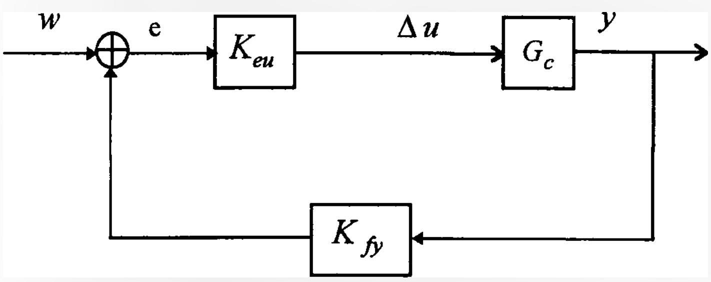


Figure 2.3: GPC closed-loop system in the general feedback form.

where:

$$K_{fy} = FK_{\hat{x}y} \tag{2.42}$$

$$K_{ue} = [I + K_{gpc}FK_{\hat{x}u}]^{-1}K_{gpc} \tag{2.43}$$

2.7 Singular Values

Singular values of a matrix $A \in C^{m\times n}$ are denoted by $\sigma_i(A)$ and defined as the k largest non-negative square roots of the eigenvalues of A^*A:

$$\sigma_i(A) = \sqrt{\lambda_i(A^*A)} \qquad i = 1,2,\cdots\cdots k$$

and ordered such that

$$\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \cdots\cdots\cdots \geq \sigma_k$$

where, $C^{m\times n}$ denotes a linear space of $m \times n$ complex matrices, A^* is the complex conjugate of the matrix A. The maximum and the minimum singular values of the matrix A are given respectively as:

$$\bar{\sigma}(A) = \sigma_1(A) \tag{2.44}$$

$$\underline{\sigma}(A) = \sigma_k(A) \tag{2.45}$$

where, $\bar{\sigma}(A)$ and $\underline{\sigma}(A)$ can be regarded as the upper and the lower bound of the gain of the system matrix A.

The importance of singular values in control theory is that they can be used to compute the frequency response of a multivariable system which can be seen as the extension of the SISO amplitude Bode plot to the multivariable systems. The stability and performance analysis of a multivariable system can be done using singular values of the closed-loop system to show whether improvements can be obtained when design parameters are modified.

2.8 Performance and Stability Robustness

Sensitivity functions (sensitivity, complementary sensitivity and control sensitivity) and unstructured additive and multiplicative perturbations are used for the assessment of the performance and stability robustness of the control design [40, 41]. The sensitivity functions of GPC closed-loop system are derived using the general multivariable feedback system shown in Figure 2.4 as follows:

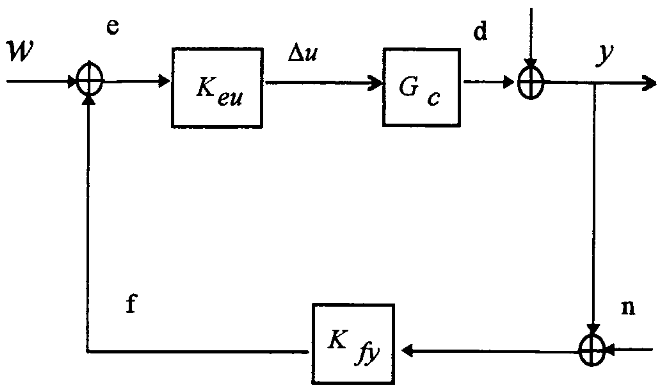


Figure 2.4: Reduced form of GPC closed loop system.

$$y = d + G_c\Delta u, \quad e = w - K_{fy}(y+n), \quad \Delta u = K_{eu}e$$

$$y = d + G_cK_{eu}(w - K_{fy}(y+n))$$

$$y = d + G_cK_{eu}w - G_cK_{eu}K_{fy}y + G_cK_{eu}K_{fy}n$$

Hence,

$$y = [I + G_cK_{eu}K_{fy}]^{-1}d + [I + G_cK_{eu}K_{fy}]^{-1}G_cK_{eu}w$$

$$+[I + G_cK_{eu}K_{fy}]^{-1}G_cK_{eu}K_{fy}n \qquad (2.46)$$

$$T_{yd} = [I + G_c K_{eu} K_{fy}]^{-1} \tag{2.47}$$

$$T_{yn} = [I + G_c K_{eu} K_{fy}]^{-1} G_c K_{eu} K_{fy} \tag{2.48}$$

The transfer function matrix from y to d denoted as T_{yd} is the sensitivity function and the transfer function matrix from y to n denoted as T_{yn} is the complementary sensitivity function.

The relation between the reference input w and the control signal Δu can be found as follows:

$$\Delta u = K_{eu}(w - K_{fy}(d + G_c \Delta u + n))$$

$$\Delta u = [I + K_{eu} K_{fy} G_c]^{-1} K_{eu} w - [I + K_{eu} K_{fy} G_c]^{-1} K_{eu} K_{fy} d - [I + K_{eu} K_{fy} G_c]^{-1} K_{eu} K_{fy} n$$

$$T_{\Delta uw} = [I + K_{eu} K_{fy} G_c]^{-1} K_{eu} \tag{2.49}$$

$T_{\Delta uw}$ is the control sensitivity function which is a measure of control activity and tracking properties of the control design.

A measure of systems closed-loop stability is the maximum singular value of the additive perturbation ΔGc. ΔGc is the transfer function matrix seen between the points $\times a \quad \times b$ as shown in Figure 2.5 and can be found as follows:

$$e = w - K_{fy} y, \quad y = b + G_c \Delta u, \quad \Delta u = K_{eu} e$$

$$\Delta u = K_{eu}(w - K_{fy}(b + G_c \Delta u))$$

$$\Delta u = [I + K_{eu} K_{fy} G_c]^{-1}(K_{eu} w - K_{eu} K_{fy} b)$$

Hence,

$$M_p = \Delta G_c = \frac{\Delta u}{b} = -[I + K_{eu} K_{fy} G_c]^{-1} K_{eu} K_{fy} \quad (2.50)$$

M_p is the transfer function matrix seen by the additive perturbation ΔGc.

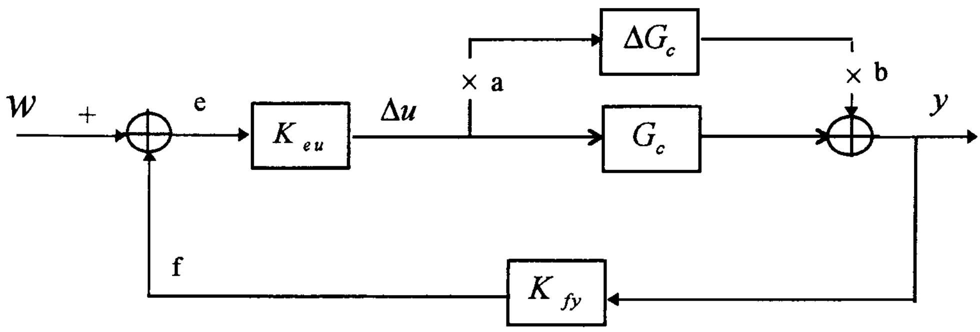


Figure 2.5: Additive perturbation.

Expressions for input and output multiplicative plant perturbations can be derived following the same procedure using Figure 2.6. The equivalent transfer function matrices seen by the input and output multiplicative perturbations are given respectively as:

$$M_i = -[I + K_{eu} K_{fy} G_c]^{-1} K_{eu} K_{fy} G_c \quad (2.51)$$

$$M_o = -[I + G_c K_{eu} K_{fy}]^{-1} G_c K_{eu} K_{fy} \quad (2.52)$$

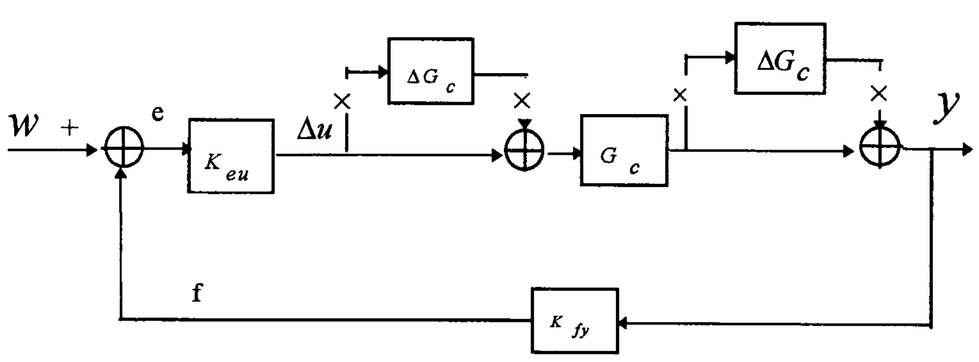


Figure 2.6 Input and output multiplicative perturbations.

The transfer function matrix M_p is the transfer function matrix seen by the additive perturbation. Its maximum singular value describes the magnitude of the maximum allowable unstructured additive uncertainty, thus a measure of robustness. M_i and M_o are the transfer function matrices seen by the input and output multiplicative perturbations respectively. Their maximum singular values describe the magnitude of the maximum allowable unstructured input and output multiplicative perturbations respectively, thus are measures of robustness.

2.9 Industrial Applications

Many practical applications have been reported in different publications. Clarke *et al.*, [3] have cited seven applications of GPC including a cement mill where a conventional PID controller has been replaced by GPC giving better performance, to an industrial spray dryer where a multivariable GPC resulted in reduced interactions, and to a robotics arm which gave good results.

Many other applications of GPC algorithm have been reported in literature. Linkens and Mahfouf [38] have reported an application of GPC algorithm in clinical Anaesthesia where, first a SISO GPC algorithm has been applied giving excellent regulation and robustness results. This work has been extended to the multivariable case which again has proved the effectiveness of the algorithm and its superiority over other control techniques which have been previously used in this field (PID and Pole-Placement). Dumor and Boucher [35] have applied GPC in the field of machine tools. Camacho and Berenguel [36] have applied the adaptive and gain-scheduling versions of GPC to a solar power plant under different operating conditions maintaining good performance in spite of the changes in the dynamics of the process. Lu and Hogg [37] have reported the application of self-tuning GPC to a 200 MW oil-fired thermal power plant to improve its dynamic response by controlling the steam pressure. Their results showed that the pressure is more stabilised than the conventional control over a wide range of operating conditions.

2.10 Simulation Example

2.10.1 Process Description

This simulation example is to demonstrate the tuning procedure of GPC controller and to examine the effects of each individual tuning parameter and the weighting matrices on the process outputs. The process model used in this demonstration is of a multi-stream distillation column given in [39]. The output variables y1, y2 and y3 represent overhead and two sidestream compositions. The inputs u1, u2 and u3 are the input manipulated variables which represent the flow rates. The system transfer function matrix of the process is a lower triangle with first order transfer functions. There are significant interactions between the input and output variables. The interactions are non-symmetrical which results in a non-symmetrical multivariable system. The continuous state space model of the plant has the following matrices:

$$A=\begin{bmatrix} -.0111 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.125 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.167 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.125 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.143 \end{bmatrix};$$

$$B=\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad C=\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}; \quad D=\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix};$$

where A is 6×6 diagonal system matrix, B is 6×3 inputs matrix, C is 3×6 outputs matrix. This system is both controllable and observable. The eigenvalues of the system matrix A show that the system is stable.

2.10.2 GPC Controller Design and Tuning Procedure

The first step in GPC controller design is to determine the step responses of the open-loop process and to check its modes. The second step is to calculate the longest rise and settling times of the process from the step responses. The third is to apply the guidelines given in Section 2.3 to find The controller tuning parameters. Next, a discretised linear model of the process and the tuning parameters are used for designing Kalman filter and calculating GPC control law. Finally fine tuning is required to get the desired performance.

The step responses of the process are shown in Figure 2.7. The longest measured rise and settling times are 23 and 35 respectively. N2 is selected to be 23 to reduce the amount of computations. From Figure 2.7 it is obvious that the process has dead time and the longest is 2 hence N1=3 (from larger plots of the step response). Since the process does not have unstable modes an arbitrary value of 5 for Nu is selected. The linear model of the distillation column is discretised and used with the tuning parameters for GPC control law calculation. After many trials, values for λ's and γ's are set to 0.8, 0.1, 0.43 and 0.01, 0.01, 0.001 respectively. The algorithm is implemented in MATLAB.

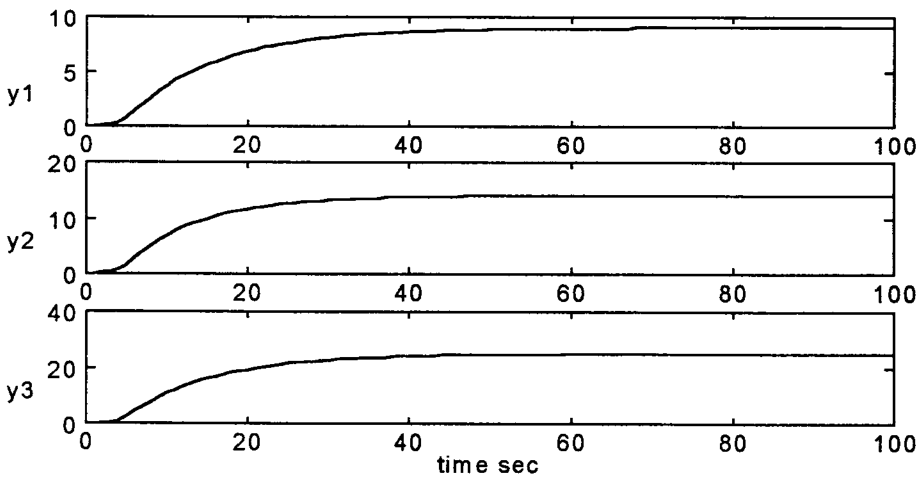


Figure 2.7: process open loop step responses.

2.10.3 Simulation Results

Simulation is performed at different sets of tuning parameters and weighting elements. Each tuning parameter is varied in a specified range with the others fixed to given values to investigate its effects on the process outputs.

2.10.3.1 Effects of N1

All the controller parameters are fixed except for N1 which is set to the following values N1=3 and N1=12. As the value of N1 is increased from 3 the process output responses became slower with longer rise and settling times and the overshoots are reduced. This can be seen in Figure 2.8.

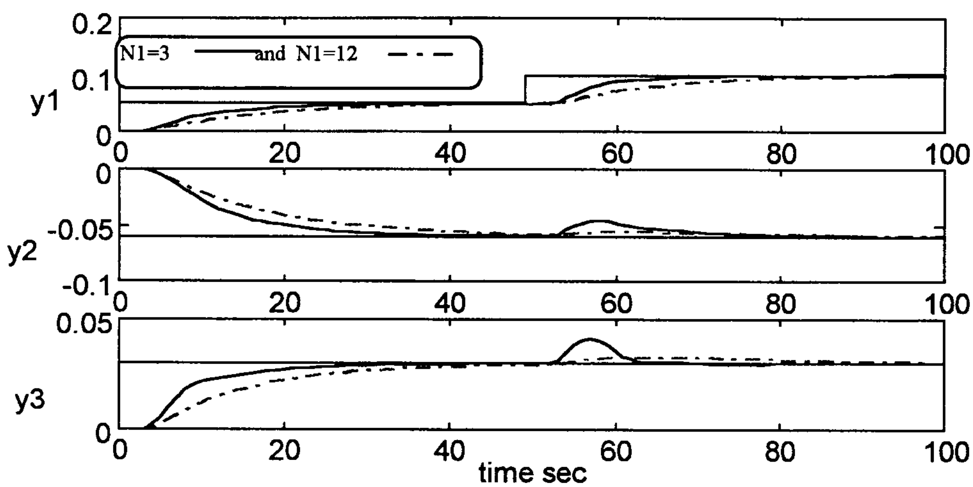


Figure 2.8: the effects of N1 on process outputs.

2.10.3.2 Effects N2

The speed of response in all the outputs increases as N2 is reduced from 23 to 15. Moreover the overshoots caused by set-points interaction are decreased as N2 is decreased. This is shown in Figure 2.9.

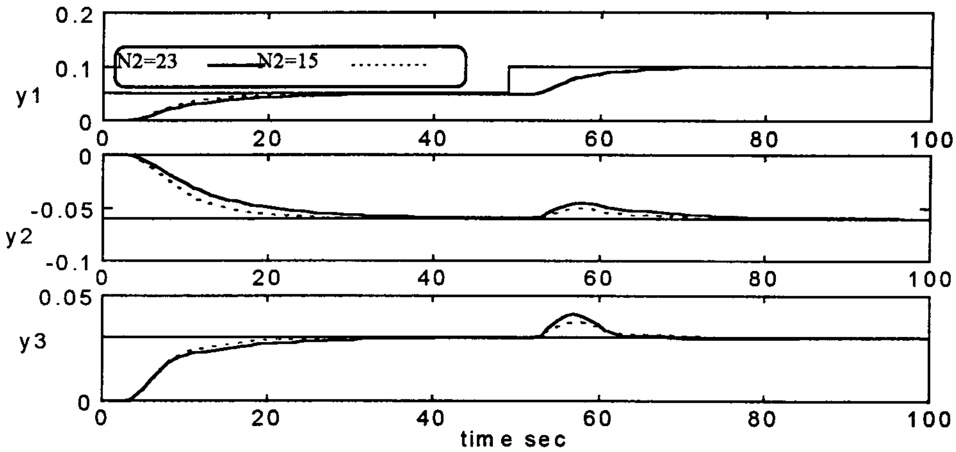


Figure 2.9: the effects of N2 on process outputs.

2.10.3.3 Effects of Nu

The control horizon Nu is increased from 3 to 15. The speed of all process outputs are increased with large overshoot in output y3 due to interaction. This can be seen from Figure 2.10.

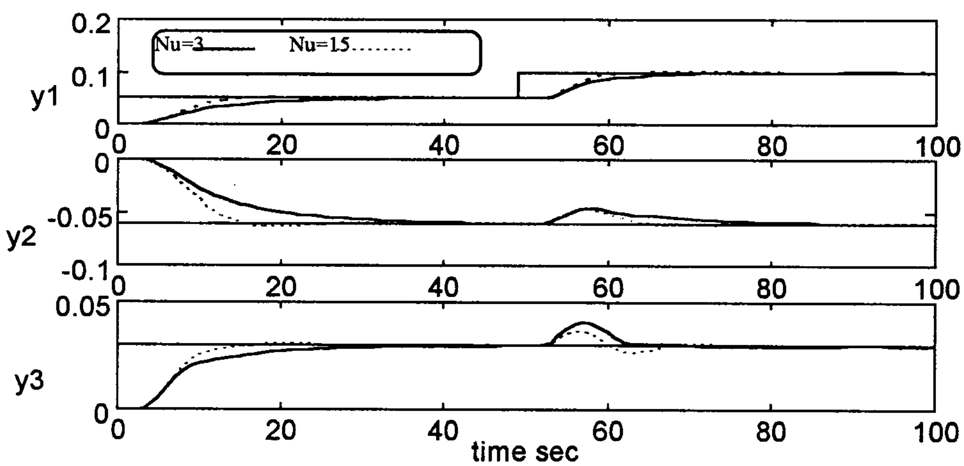


Figure 2.10: the effects of Nu on process outputs.

2.10.3.4 Effects of the Weighting on Control

The weighting $\lambda 1$ on the first input control signal has different effects on the process outputs. While the speed of response of y2 is increased, as a result of an increase in $\lambda 1$ from 0.01 to 0.1, the output y1 takes longer time to reach steady state. Overshoots due to set-points interaction are nearly eliminated as shown in Figure 2.11.

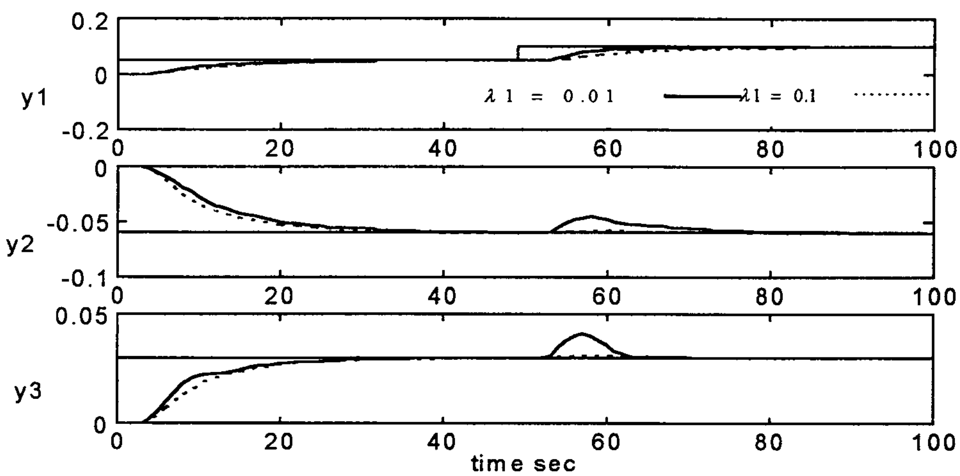


Figure 2.11: The effects of the weighting on the first input control signal.

The simulated plant model have interactions between its input and output variables and the application of the multivariable GPC controller to this plant resulted in decoupled outputs as can be seen from figure 2.11 and in good tracking performance. The tuning parameters and weighting elements to achieve a good performance are not unique. Many combinations can be obtained to fine tune the controller. However, for systems with strong interactions, it is not easy to find optimal settings easily for good performance. The controller has achieved set-point decoupling as predicted. This is due to the fact that the control law has a term that can be interpreted as a feed forward part.

2.11 Conclusions

The advantages of state space representation of industrial processes over polynomial descriptions were summarised. A multivariable state space GPC algorithm was discussed and the selection rules for the tuning parameters were summarised and discussed. The general optimisation problem was addressed and a multivariable state space GPC constrained problem was solved as a QP optimisation problem. Performance and stability robustness studies of GPC control design were reviewed and a new performance and stability robustness method for multivariable state space GPC was developed. The method is similar to the one used for H_∞ control design and analysis. Conditions for performance and stability robustness of GPC closed-loop system were derived. Simulation example of a multivariable industrial process was presented to demonstrate the tuning procedure of GPC controller.

Chapter Three

Gas Turbine Power Plant Modelling and Simulation

3.1 Introduction

In this chapter the operation of a gas turbine power plant and its subsystems will be briefly discussed. A single unit gas turbine will be modelled. The implementation and simulation of the nonlinear model on SIMULINK will be discussed and the simulation results will be presented. The nonlinear model will be controlled using conventional PID controllers. Controller tuning procedures will be presented and the results will be analysed. The closed-loop linear model of the plant is then derived and its order is reduced. The nonlinear and the linear model responses are then compared and analysed.

The models developed in this Chapter are used as a basis for designing a second layer of controllers for gas turbines to optimally manoeuvre the set-point to a new steady state condition. The second layer is designed using GPC control design technique.

Turbines are a means of producing mechanical power. Working fluids such as, water, steam, air, etc., are used to drive the turbines. Usually hot gases produced by boilers and nuclear reactors, bulky and expensive installations, are used indirectly to drive the turbine. Using hot gases to drive the turbine directly results in more compact, efficient, and economic units which are the gas turbines with net power of up to 200 MW [46]. Advanced gas turbines with pressure ratios up to 30:1, component efficiencies of 85% to 95%, and working temperature up to 1500 K are reported in [48].

3.2 Gas Turbine Power Plant Operation and Structure

3.2.1 Operation

The mechanical power produced by the gas turbines is obtained through a series of processes that is, compression, combustion, and expansion of the working fluid:

Compression of the working fluid (i.e. air which is supplied through guide vanes) to provide pressure ratio is accomplished by means of a single or a multi-stage compressor. Thermal energy is added to the working fluid by burning it with fuel which is supplied to the combustor through a fuel system. Mechanical power is extracted from the hot working gases through expansion process in the turbine (single or multi-stage). Part of the extracted power is used to drive the compressor and the rest can be used to drive an electric generator. These gases are then exhausted to the atmosphere through an exhaust system. The pollutants in the exhausted gas need to be at a minimum level.

Gas turbines can be operated in open or closed thermodynamic cycles. In an open cycle operation Figure 3.1 atmospheric air is supplied to the compressor to produce pressure ratio, then the compressed air is supplied to the combustor where fuels are added and combusted continuously (constant pressure) to add energy to the working fluid. Next the produced hot gases are expanded in the turbine and exhausted to the atmosphere.

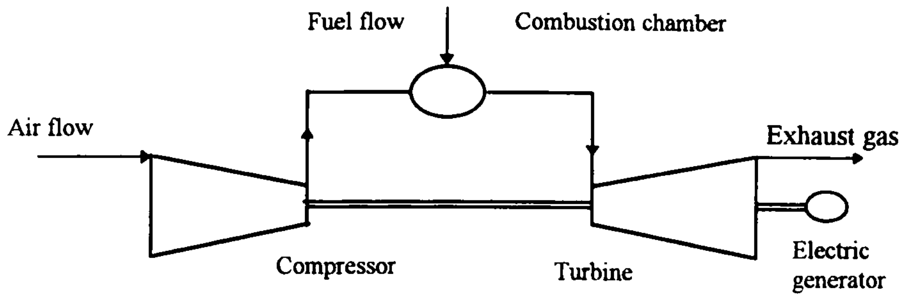


Figure 3.1: Open cycle gas turbine operation.

However, in the closed cycle operation Figure 3.2 the working gas repeatedly circulates in the gas turbine system and additional energy is added to these gases by heat exchange with another hot fluid heated in a boiler or a nuclear reactor. Coolers are necessary in this type of systems to cool down the hot working fluid before entering the compressor.

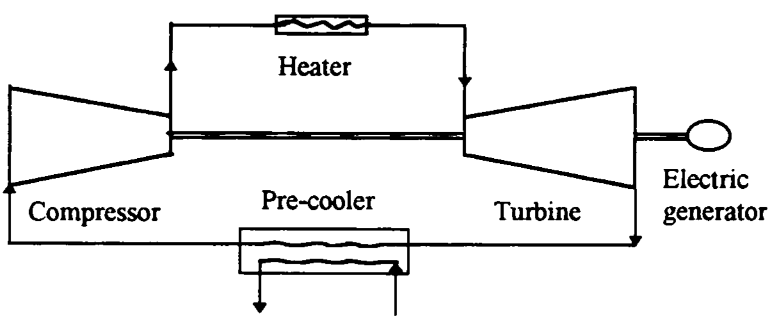


Figure 3.2: Closed cycle gas turbine operation.

Gas turbines can be constructed as single shaft, double shaft, or twin spool turbines. Cohen [48] and Harman [47] gave a detailed review of these types. Here a single unit gas turbine power plant will be briefly described.

3.2.2 Structure

A single unit gas turbine power plant Figure 3.3 is composed of several interconnected subsystems as follows:

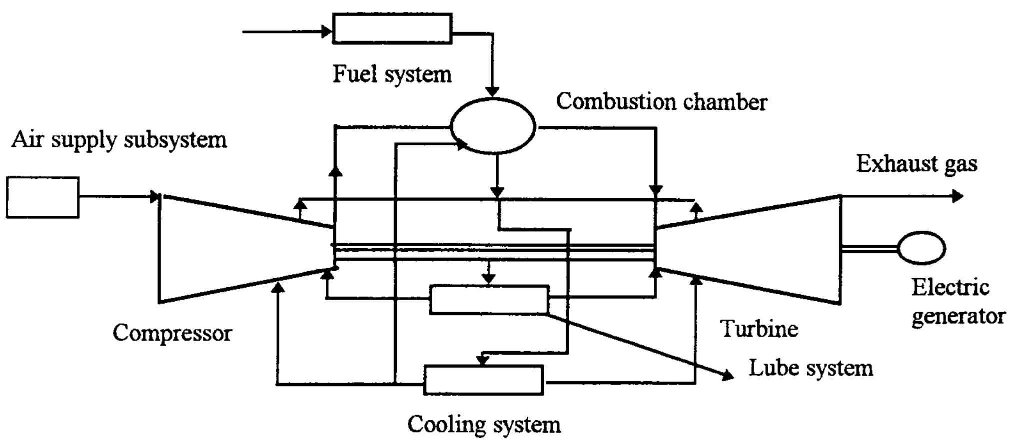


Figure 3.3: Single unit gas turbine power plant subsystems.

3.2.2.1 Fuel System

Fuel is supplied to the gas turbine combustion chamber in a controlled amount to satisfy a predetermined fuel to air ratio for efficient combustion. The fuel is supplied through feeding pumps and controlled valves (position servo, gas valve, pressure ratio valve, dump valve and stop valve) [49] and sprayed through nozzles distributed in the combustion chamber for the best combustion efficiency and regular temperature distribution throughout the chamber.

3.2.2.2 Air Supply System

The air supply to the air compressor is cleaned by passing it through filters and then passed through inlet guide vanes, which are controlled to provide the required amount of air to the compressor.

3.2.2.3 Cooling and Lubrication Systems

Thermal stresses in the turbine system may be developed as a result of high temperature generated by combustion and friction between the mechanical rubbing parts. This causes material fatigue and component failure. To reduce or to prevent such effects, properly designed cooling and lubrication systems are necessary.

The lubrication system (pumps, piping system, oil reservoir and cooler, safety devices, filters, etc.) circulates lubricants to cool the rubbing parts and to cool the lubricants by heat exchange with external air or water. Oil (the lubricant) is filtered before returning back by passing it through filters.

Critical parts (the nozzles and rotor blades) of the turbine must be protected against thermal stresses developed as a result of high working temperatures. This is accomplished by the selection of proper materials to manufacture the critical parts and properly designed cooling system. External parts of the turbine system can be cooled by Coolants which may be air or other fluids. Coolants circulation is achieved using pumps or bleeding some air from the low pressure compressor stage to the cooling passages.

3.2.2.4 Compressor

Air compressors can be reciprocating or rotary types. Reciprocating air compressors can be designed with a single or multi-stage. Rotary air compressors include three types centrifugal compressors, axial flow compressors and positive displacement compressors [47, 50]. Axial flow compressor which is commonly used in gas turbines, is composed of fixed and moving blades. The fixed blades are attached to the compressor outer casing and act as vanes and diffusers. The moving blades are fixed to a rotor and are used to progress the air axially along the compressor. With such a compressor compression ratios of 10:1 or more can be obtained.

3.3.2.5 Combustor

There are three types of combustors, tubular, annular and tuboannular [51]. A tubular combustor consists of small tubular combustion chambers distributed around the rotating shaft connecting the compressor and the turbine. This type is used for centrifugal compressors. The combustor that is commonly used for axial flow turbines is the annular combustor which consists of a single chamber surrounding the rotating shaft. A combination of tubular and annular results in tuboannular combustor.

3.2.2.6 Turbine

Radial flow and axial flow are the basic two types of turbines. The more suitable turbine for high temperatures experienced in gas turbines is the axial flow type of turbines [47]. Hot gases enter the turbine through nozzle guide vanes which direct the gases on the blades that are arranged on the rotating shaft to extract the maximum possible energy of the gases passing over them. Gases enter the axial flow turbine at high temperature, pressure and velocity and leave the turbine at low temperature, pressure and velocity. This causes expansion of the gases in the turbine. Through this expansion energy is being released and delivered to the rotating shaft of the turbine.

3.2.2.7 Electrical Generator

An electrical generator is a means of generating electrical energy. It is composed of the generator, exciter, control and protection systems. It may be an ac (Alternating current) or dc (Direct current) generator. An ac generator can be an induction type or synchronous type. In the second type the output frequency is made equal exactly to the rotational speed of the rotating parts. Most machines used for electrical power generation are synchronous type due to their capability of generating much more power than other machines. The generator mainly consists of two parts [57]:

- A rotor with field winding connected to an excitation system (self or separate excitation) to provide the necessary (dc) current to produce magnetic fields.

- Stator with armature windings arranged in three groups separated by one third of a pole pair. These poles with the rotational speed of the machine determine the output frequency. The rotating magnetic field when cuts the armature windings a three phase voltage is generated which is controlled by the exciter. The power generated is determined by the power delivered by the prime-mover (the turbine) to the generator.

3.2.2.8 Gas Turbine Power Plant Control System

Early gas turbine control systems which include (fuel control, temperature control, sequence of operation control and protection control systems) were mainly hydro-mechanical, electromechanical, pneumatic, and manual with independent protection against over speed, over temperature, fire, loss of flame, loss of lubrication oil pressure and vibration. These control systems are replaced by new advanced systems such as microprocessor based systems. A historical review of gas turbine control system development is given in [52]. Gas turbine control systems, description, classification, operation, main components, and developments can be found in [49, 53, 54, 55, 56].

A gas turbine power plant control system consists of two subsystems, turbine side control subsystem and electrical generator control subsystem. A brief description of these control subsystems is given in the following Sections.

3.2.2.8.1 Turbine Control System

A description of both the old and the modern control systems (Digital Control Systems, DCS) was reported and their performances were compared in [49]. The turbine control system is divided into four individual subsystems. Fuel control system which has three main components: a governor generator with its associated speed circuitry, speed switches, and a fuel governor. These components may be replaced by more advanced components and software. Temperature control system which includes thermocouples, radiation shields, and complex electric-pneumatic system in old systems or electronic components and software in modern systems.

Sequence control systems were based on relays and timers whereas, in modern systems digital switching is more common. Protection control system detects abnormal conditions such as (over speed, over temperature, excessive vibration, fire, etc.,) and provides the necessary condition, signalling, and trip the units under abnormal conditions. Old protection systems were mechanical, electromechanical and electro-pneumatic such systems are substituted with electrical and electronic systems.

3.2.2.8.2 Electrical Generator Control System

Electrical generators are controlled to give the required outputs, output power (active and reactive), rotational frequency, and voltage. Due to the fact that the generator is not independent of the turbine, its output power and frequency are regulated through the turbine regulating power and speed loops respectively. In general two control actions are well known to control electrical generators [57]:

- Control by field (rotor) currents.
- Control by prime-mover shaft torque or power.

Generator frequency is controlled and used to determine the amount of fuel input to the turbine which acts to regulate the prime-mover output torque which in turn regulates the generator output power and frequency. Generator control system includes automatic voltage regulator (adjusts the excitation on reception of a voltage signal from the generator) which is responsible for keeping voltage as specified by the set-point.

3.3 Gas Turbine Models a Brief Review

A simplified model of heavy-duty gas turbines for the purpose of investigation of power system stability and the development of dispatching strategy has been introduced in [53]. The model is limited to simple cycle, single shaft heavy-duty gas turbines (18MW to 106MW), with open inlet guide vanes only and allowable speed range from 95 to 107 percent of the rated speed under ISO conditions (i.e. ambient temperature=15° C or 59° F and atmospheric pressure=101.325 kpa or 14.696 psia).

Another simplified model of single shaft gas turbine was presented by Hussain and Seifi [58]. The model is based on thermodynamic conservation laws, mass balance, energy balance, and mass flow. The obtained model is a simple transfer function which is valid for power system stability analysis under the assumption of operation at fixed load and fixed speed conditions. Models of twin shaft gas turbines with their control systems can be found in [55].

Data and fundamental theory for gas turbine modelling and control can be found in [52]. A model of a single shaft gas turbine suitable for supervisory control applications for combined cycle and combined heat and power plants is described in [59]. The gas turbine model introduced here is based on this model and on the data given in [59]. The model is suitable for the evaluation and implementation of supervisory control strategies. Model assumptions and formulation are reviewed in the following Section.

3.4 Gas turbine Power Plant Model Description and Assumptions

The non-linear model of the gas turbine power plant unit is composed of fuel system submodel, compressor submodel, turbine submodel and electrical generator submodel. These submodels are based on first principles of physics and thermodynamic laws with appropriate simplifications [59]. The data necessary to implement the gas turbine model is given in Appendix A.1. Figure 3.4 shows the single line diagram of the open loop gas turbine model.

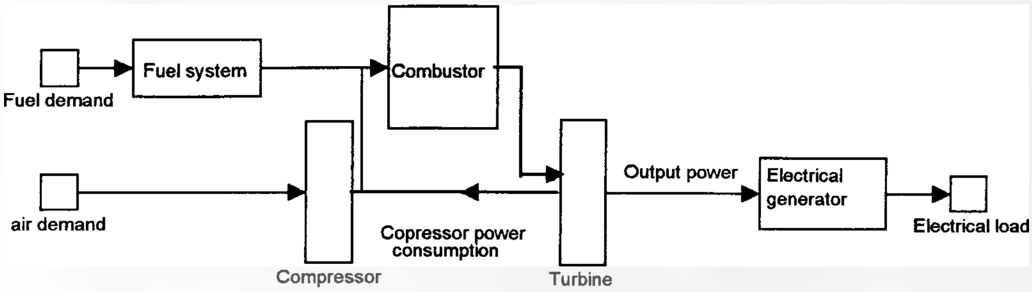


Figure 3.4 Single line diagram of an open-loop gas turbine model.

Each submodel is going to be written as a MATLAB program and to be represented by a SIMULINK building block. Then these blocks will be connected together to constitute the full nonlinear model. PID controllers will be designed to stabilise the nonlinear model around a nominal operating point. Model assumptions are:

- Air and combustion products are treated as perfect gases.
- Specific heats of air and injected steam assumed to be constant for combustion products.
- Flow through nozzles is described by a one dimensional adiabatic uniform polytropic process.
- Energy storage and transport delay in the compressor, turbine and combustion chamber are relatively small, thus steady state equations are applied.
- Inlet kinetic energy of gas flows into the compressor and turbine is negligible.
- Air mass flow through the compressor is controllable via inlet guide vanes.

Nox and CO pollutant formation is modelled as a static function of steam/fuel ratio.

3.4.1 Gas Turbine Model

3.4.1.1 Fuel System

Fuel system is modelled in terms of fuel system actuator dynamics and fuel valve positioner dynamics. A Least Value Gate (LVG) and A Maximum Value Gate (MVG) are modelled using MATLAB functions.

- Fuel system actuator:

$$w_f = \frac{k_{ff}}{\tau_f s + 1} e_1 \tag{3.1}$$

where:

w_f - fuel mass flow [kg/sec].

k_{ff} - fuel system gain constant.

τ_f - fuel system time constant [sec].

e_1 - valve position.

- Valve positioner:

$$e_1 = \frac{a}{bs + c} e_2 \tag{3.2}$$

where:

a,b,c - valve parameters.

e_2 - internal signal.

Signal for valve positioner:

$$e_2 = MF - k_f w_f + F_d \omega e^{-s\tau} \tag{3.3}$$

where:

MF - minimum fuel signal.

k_f - feedback coefficient.

F_d - fuel demand signal.

τ - fuel system pure time delay [sec].

ω - rotation speed of the turbine (from the generator model) [rad/sec].

- Least Value Gate (LVG) is provided to select a minimum of three control signals that is, power control signal, frequency (speed) control signal and exhaust gas temperature control signal. This minimum signal determines the amount of fuel flow to the turbine system.

- A minimum fuel set point is provided to prevent flame loss during loss of the fuel control signal. This is done by selecting the maximum value of either the minimum fuel set-point or the minimum fuel flow control signal using a Maximum Value Gate (MVG).

SIMULINK block diagram of the fuel system is shown in Figure 3.5. Physical constants and data of the model are given in Appendix A.1.

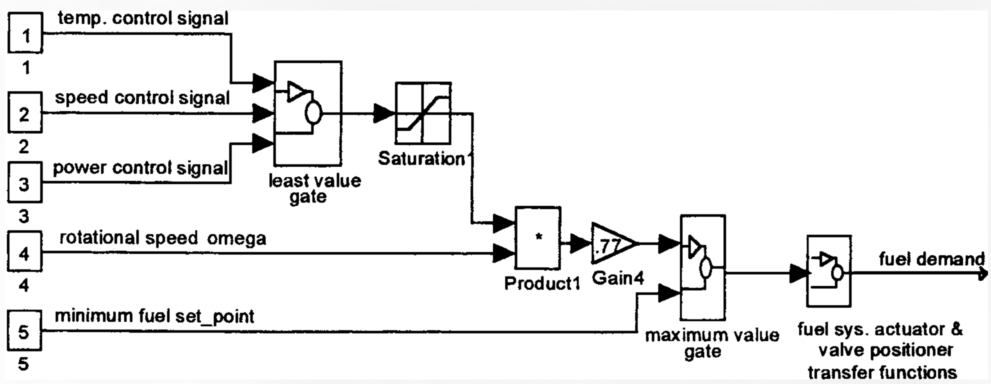


Figure 3.5 SIMULINK model of the gas turbine fuel system.

3.4.1.2 Compressor

The compressor is described by a one dimensional adiabatic uniform polytropic steady state flow nozzle equation.

$$w_a = A_o\left[\left(\frac{2m_a}{\eta_{\infty c}(m_a-1)}\right)\rho_i p_{cin}\left(r_c^{2/ma} - r_c^{(m_a+1)/m_a}\right)\right]^{1/2} \tag{3.4}$$

$$m_a = \frac{\gamma_a}{\gamma_a - \dfrac{(\gamma_a - 1)}{\eta_{\infty c}}} \tag{3.5}$$

where:

w_a - air mass flow into the compressor [kg/sec].

A_o - compressor exit flow area [m^2].

$\eta_{\infty c}$ - compressor polytropic efficiency.

ρ_i - inlet air density [kg/m^3].

p_{cin} - inlet air pressure [Pa].

m_a - polytropic index.

r_c - pressure ratio (outlet/inlet).

$\gamma_a = (c_{pa} / c_{va})$ ratio of specific heats for air (constant).

c_{pa} - specific heat at constant pressure for air [J/(kg °K)].

c_{va} - specific heat at constant volume for air [J/(kg °K)].

To solve for r_c equation (3.4) can be arranged as:

$$r_c^{(\frac{m_a+1}{m_a})} - r_c^{(\frac{2}{m_a})} = \frac{\eta_{\infty c}(m_a-1)}{2m_a}\frac{1}{\rho_i p_{cin}}\left(\frac{w_a}{A_0}\right)^2 \tag{3.6}$$

Outlet Air Pressure Equation

$$p_{cout} = p_{cin} r_c \tag{3.7}$$

Outlet Air Temperature Equation

$$T_{cout} = r_c^{(\frac{\gamma_a - 1}{\gamma_a \eta_{\infty c}})} T_{cin} \tag{3.8}$$

where:

T_{cout} - outlet air temperature [°K].

T_{cin} - inlet air temperature [°K].

Compressor Power Consumption Equation

$$p_c = \frac{w_a \Delta h_I}{\eta_c \eta_{trans}} \tag{3.9}$$

where:

p_c - compressor power consumption [W].

Δh_I - isentropic enthalpy change corresponding to compression from p_{cin} to p_{cout}.

η_c - overall compressor efficiency.

η_{trans} - transmission efficiency from turbine to compressor.

Overall Compressor Efficiency Equation

$$\eta_c = \frac{1-(r_c)^{(\frac{\gamma_a - 1}{\gamma_a})}}{1-(r_c)^{(\frac{\gamma_a - 1}{\gamma_a \eta_{\infty c}})}} \tag{3.10}$$

Perfect Gas Isentropic Enthalpy Change Equation

$$\Delta h_I = c_{pa} T_{cin} (r_c^{R_a / c_{pa}} - 1) \tag{3.11}$$

where:

R_a - air gas constant [J/(kg °K)].

Equation (3.6) is solved using the iterative Newton-Raphson method and the compressor equations are implemented in MATLAB (Appendix A.2). Inputs and outputs of the compressor are shown on the SIMULINK building block in Figure 3.6.

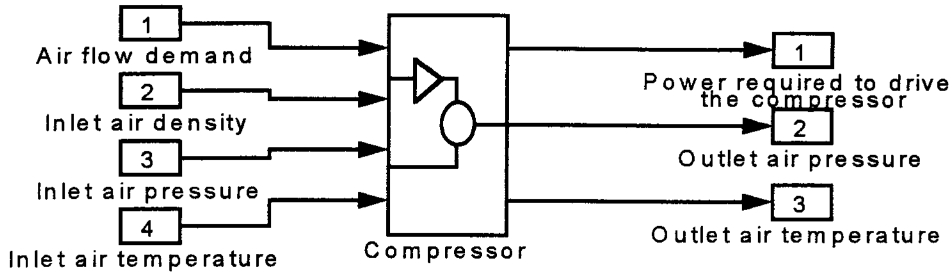


Figure 3.6: SIMULINK model of the compressor

3.4.1.3 Combustion Chamber

Energy and pressure drop equations are used for modelling the combustor.

Exhaust Gas Mass Flow

$$w_G = w_a + w_f + w_{is} \tag{3.12}$$

Combustion Energy Equation

$$w_G c_{pg}(T_{Tin} - 298) + w_f \Delta h_{25} + w_a c_{pa}(298 - T_{cout}) + w_{is} c_{ps}(298 - T_{is}) = 0$$

which is rearranged as follows:

$$T_{Tin} = \left(\frac{1}{w_G c_{pg}}\right)\left[-w_f \Delta h_{25} - w_a c_{pa}(298 - T_{cout}) - w_{is} c_{ps}(298 - T_{is})\right] + 298 \tag{3.13}$$

where:

w_f - fuel mass flow [kg/sec].

w_{is} - injection steam mass flow [kg/sec].

c_{pg} - specific heat of combustion gases [J/(kg °K)].

T_{Tin} - turbine inlet gas temperature [°K].

Δh_{25} - specific enthalpy of reaction at reference temperature of $25°C$ [J/(kg)].

c_{ps} - specific heat of steam [J/(kg °K)].

T_{is} - temperature of injected steam [°K].

$$p_{tin} = p_{cout} - \Delta p \tag{3.14}$$

$$\Delta p = p_{cout}\left[\left(k_1 + k_2\left(\frac{T_{Tin}}{T_{cout}} - 1\right)\right)\frac{R}{2}\left(\frac{w_G}{A_m p_{cout}}\right)^2 T_{cout}\right] \tag{3.15}$$

where:

p_{Tin} - pressure of combustion gases at turbine inlet [Pa].

Δp - combustion chamber pressure loss [Pa].

k_1, k_2 - pressure loss coefficients.

R_{cg} - universal gas constant for combustion gases [J/(kg °K)].

A_m - combustion chamber mean cross-sectional area [m^2].

Pollutant Formation

$$g'_{cnox} = f_{GT2}(\frac{w_{is}}{w_f}) \tag{3.16}$$

$$g'_{cco} = f_{GT3}(\frac{w_{is}}{w_f}) \tag{3.17}$$

where:

g'_{cnox} - mass flow of Nox [ppm].

g'_{cco} - mass flow of CO [ppm].

f_{GT2} - experimental curve: Nox mass flow as a function of steam to fuel mass flow ratio.

f_{GT3} - experimental curve: CO mass flow as a function of steam to fuel mass ratio.

Pollution Formation Measurement Dynamics

$$g_{cnox}(t) = g'_{cnox}(t - \tau_m) \tag{3.18}$$

$$g_{cco}(t) = g'_{cco}(t - \tau_m) \tag{3.19}$$

where:

$g_{cnox}(t)$ - delayed (measured) Nox mass flow.

$g_{cco}(t)$ - delayed (measured) CO mass flow.

τ_m - measurement delay [sec].

Equations describing the combustor and pollutant formation are implemented in MATLAB program (Appendix A.3). SIMULINK building block of the combustor is show in Figure 3.7.

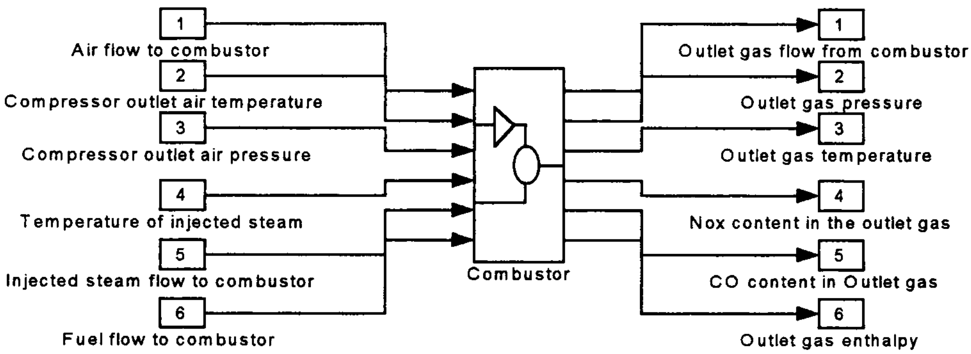


Figure 3.7: SIMULINK model of the combustor.

3.4.1.4 Turbine

Gas mass flow through the turbine:

$$w_G = A_{To}\left[\left(\frac{2\eta_{\infty T} m_{cg}}{m_{cg}-1}\right)\rho_{Tin} p_{Tin}\left[r_T^{(2/m_{cg})} - r_T^{\left(\frac{m_{cg}+1}{m_{cg}}\right)}\right]\right]^{1/2} \tag{3.20}$$

This equation rearranged to solve for r_T (outlet/inlet turbine pressure ratio=$\frac{p_{Tout}}{p_{Tin}}$)

as:

$$r_T^{(\frac{m_{cg}+1}{m_{cg}})} - r_T^{(\frac{2}{m_{cg}})} = \left(\frac{w_G}{A_{To}}\right)^2 \frac{(m_{cg}-1)}{2\eta_{\infty T} m_{cg} \rho_{Tin} p_{Tin}} \tag{3.21}$$

$$m_{cg} = \frac{\gamma_{cg}}{\gamma_{cg} - \eta_{\infty T}(\gamma_{cg}-1)} \tag{3.22}$$

$$\rho_{Tin} = (p_{Tin} / R_{cg} T_{Tin}) \tag{3.23}$$

$$p_{Tout} = r_T p_{Tin} \tag{3.24}$$

where:

ρ_{Tin} - inlet gas density [kg/m^3].

m_{cg} - combustion gases polytropic index.

A_{To} - turbine exit flow area [m^2].

$\eta_{\infty T}$ - turbine polytropic efficiency.

w_G - inlet gas flow from combustor [kg/sec].

$\gamma_{cg} = \left(\frac{c_{pg}}{c_{vg}}\right)$ specific heat ratio for combustion gases (3.25)

Temperature-Pressure Relationship

$$T_{Tout} = T_{Tin} r_T^{\frac{\eta_{\infty T}(\gamma_{cg}-1)}{\gamma_{cg}}} \tag{3.26}$$

where:

T_{Tout} - gas temperature at exit of turbine [°K]

Overall Turbine Efficiency

$$\eta_T = \frac{1-(r_T)^{(\eta_{\infty T}\frac{\gamma_{cg}-1}{\gamma_{cg}})}}{1-(r_T)^{(\frac{\gamma_{cg}-1}{\gamma_{cg}})}} \tag{3.27}$$

η_T - overall turbine efficiency

Power Delivery

$$p_T = \eta_T w_G \Delta h_I \tag{3.28}$$

$$p_{mech} = p_T - p_c \tag{3.29}$$

with:

$$\Delta h_I = c_{pg} T_{Tin} (r_T^{R_{cg}/c_{pg}} - 1) \tag{3.30}$$

$$h_{Tout} = h_{Tin} + \eta_T \Delta h_I \tag{3.31}$$

where:

p_T - mechanical power delivered by turbine [Pa]

p_c - power required to drive the compressor [W]

p_{mech} - net available mechanical power [W]

Δh_I - isentropic enthalpy change for a gas expansion from p_{Tin} to p_{Tout} [J/kg].

h_{Tout} - exhaust gas enthalpy [J/kg].

Equation (3.21) is solved for the pressure ratio r using the iterative Secant method. Turbine equations are written as a MATLAB program (Appendix A.4) and represented by a SIMULINK building block as shown in Figure 3.8. The SIMULINK building block for the interconnected compressor, combustor and turbine system is shown in Figure 3.9.

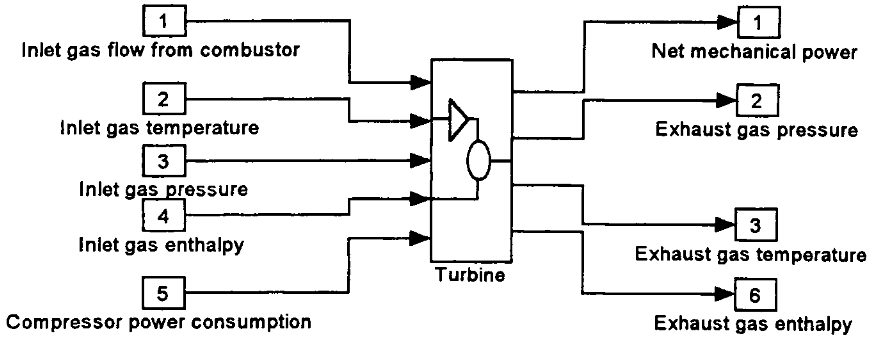


Figure 3.8: SIMULINK model of the turbine.

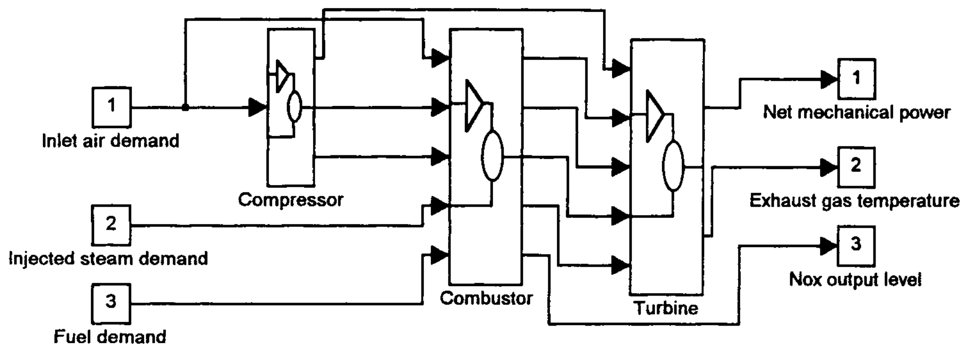


Figure 3.9: SIMULINK model of the open-loop Gas turbine.

3.4.2 Electrical Generator Model

The electrical generator model is represented by a swing equation with a stochastic noise generator to represent the real electrical load fluctuations. The electrical generator SIMULINK building block model based on the generator swing differential equation is shown in Figure 3.10. The swing equation is given in terms of power. The model assumptions are:

- rotor speed does not vary greatly from synchronous speed.
- Windage and friction losses due to rotation are ignored.

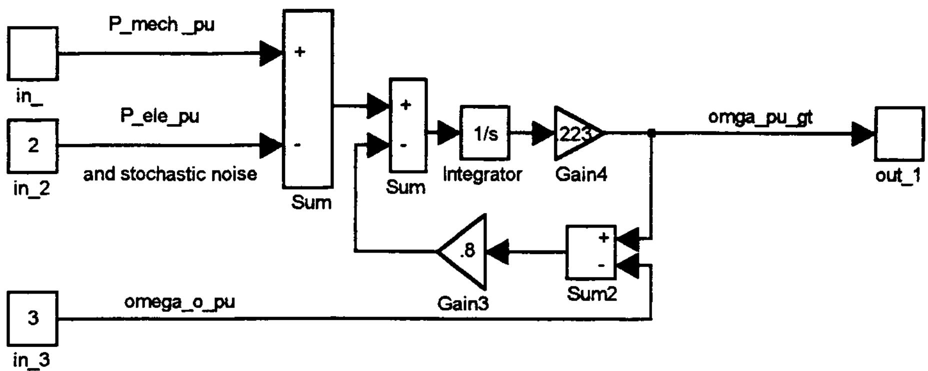


Figure 3.10: Electrical generator block diagram.

$$\frac{d\omega}{dt} = \frac{\omega o}{2H_{eq}}\left[p_{mech} - p_{el} - D_{eq}(\omega - \omega o)\right] \tag{3.32}$$

where:

ω - frequency of the generator [rad/sec].

H_{eq} - inertia constant for machine.

p_{mech} - mechanical shaft power [W].

p_{el} - real electrical load power [W].

D_{eq} - electrical damping coefficient.

ωo - synchronous frequency [rad/sec].

3.4.3 Turbine and Generator Controller Model

This controller model is mainly intended to control the mechanical side of the gas turbine. However, the generator output power and frequency are controlled as a result of controlling the turbine side. The controller model can be regarded as a four input three output system where, the controller inputs are the turbine outputs and the controller outputs are the inputs to the turbine system. The controller model block diagram is shown in Figure 3.11. This model consists of the following components:

- Local power regulating loop which is the most important loop that specifies the amount of fuel flow to the turbine system in normal operation. A proportional integral (PI) control action is applied to this loop.
- Local frequency and exhaust gas temperature regulating loops are provided to achieve safe operation by ensuring that the speed and temperature limits are not exceeded during load changes. PID control action and PI control action are provided to regulate frequency (speed) and temperature respectively.
- *Nox* pollutant content in the exhaust gas is regulated through a separate loop using PI control action by controlling the amount of the injected steam in the combustion chamber.
- Exhaust gas temperature regulating loop, with PI control action, specifies the amount of air flow to the compressor system by controlling its inlet guide vanes.

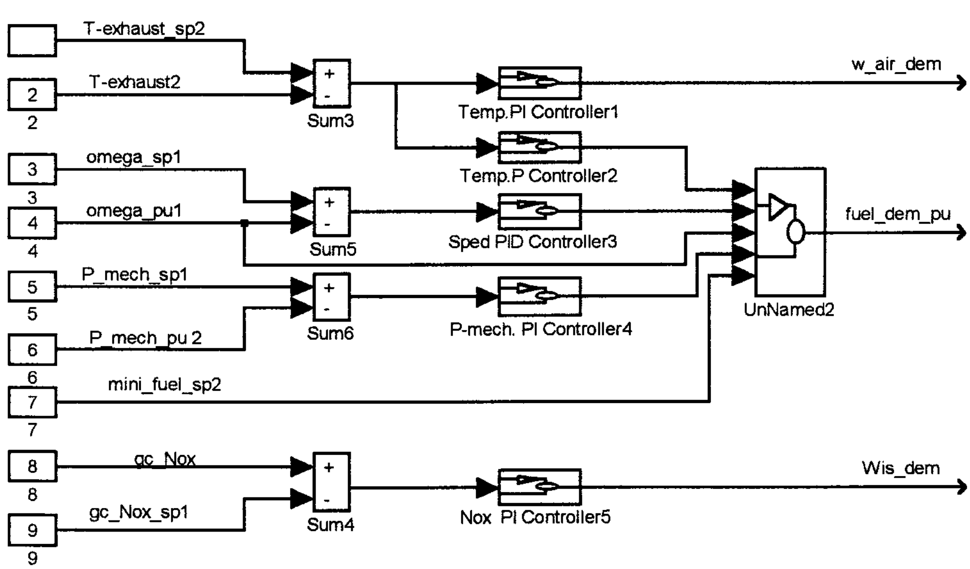


Figure 3.11: Gas turnine and generator controllers model.

3.5 Model Implementation

The described power plant model is implemented using MATLAB routines and SIMULINK building blocks. The submodels that is, compressor model, combustor model and turbine model, which are described by equations (3.4) to (3.31), are written as MATLAB subroutines (m.files). These subroutines are implemented as SIMULINK blocks using S-functions to allow communication with other submodels. A SIMULINK model can be run from the SIMULINK menu as well as the MATLAB command line. SIMULINK provides different simulation algorithms (integration methods to solve model differential equations) such as, Linsim algorithm, Runge-Kutta third and fifth order algorithms, Gear's predictor-corrector algorithm, etc. Each method is designed for a particular system for instance, Gear's method is designed for nonlinear stiff systems (systems with both fast and slow dynamics). This method of integration is found to be the best method to simulate the gas turbine power plant model of interest because of its nonlinearities and stiffness. Integration parameters i.e., minimum step size, maximum step size and relative error tolerance of integration are set to 0.1, 1, 0.1, respectively. All simulations are run for 200 seconds to investigate the model responses both at transient period and at steady state. The necessary data to implement the model which include inputs, parameters and steady state outputs can be found in appendix A.1. The full controlled nonlinear model is shown Figure 3.12.

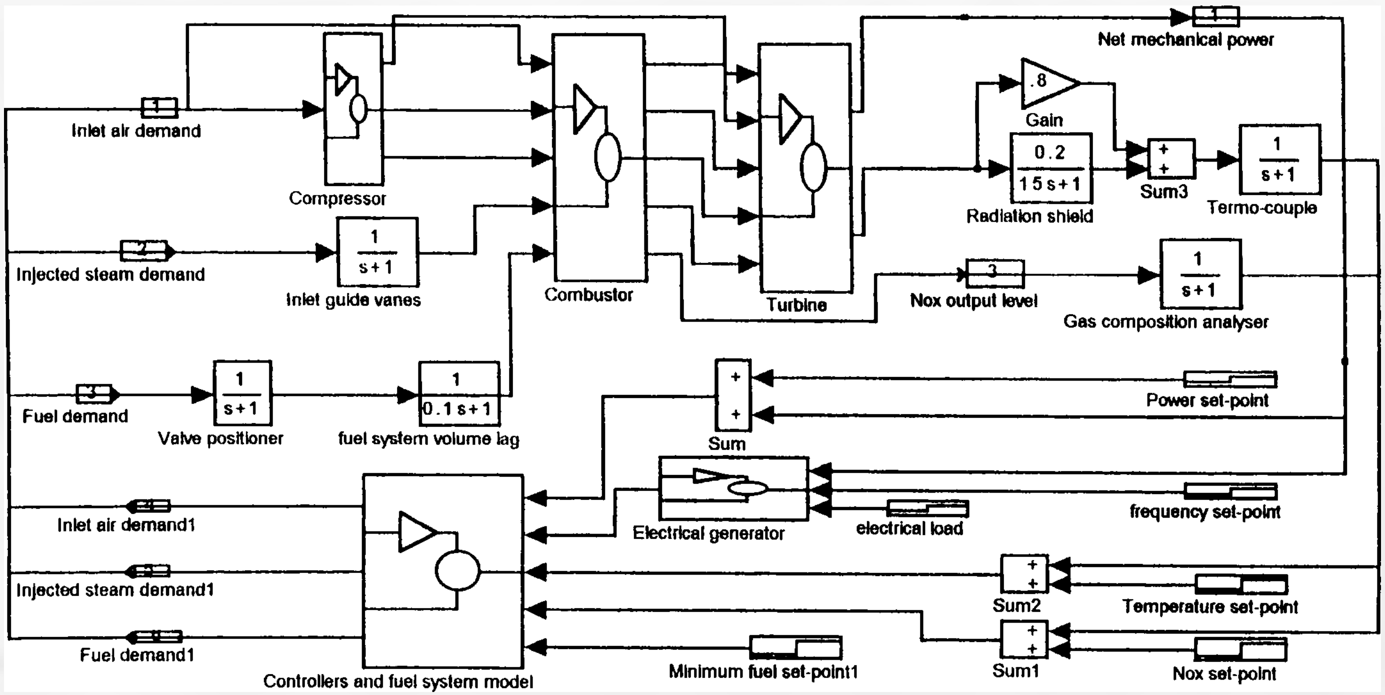


Figure 3.12 Gas turbine power plant nonlinear SIMULINK model.

3.6 Controller Tuning

PID controllers are widely used in process industry. Many tuning procedures for PID controllers are available in the literature. Some tuning procedures are based on time domain measurements such as process reaction method which is based on open-loop step response of the process, sustained oscillation method that is based on closed-loop system step response to get sustained oscillations of the process outputs, and quarter amplitude-decay method in which the amplitude of the second peak is made equal to 0.25 the amplitude of the first peak. In the last two methods ultimate gain and ultimate period are used to determine the controller parameters. These methods can be found in many standard control books such as, Franklin *et al* [60]. Other methods are based on frequency response measurements such as sustained oscillation method in which gain margin and cut-off frequency are used to determine the controller parameters Aström [61].

The procedure followed for controllers tuning in this work is carried out by trial and error based on the manual procedure described in Klafter *et al* [62]. Firstly, the loop of interest is closed and a step input is applied to this loop. Secondly the proportional gain k_p of the controller is increased, with the integral gain k_i and the derivative gain k_d are set to zero, until the system step response becomes critically damped or slightly underdamped. Thirdly the integral gain is changed to minimise or to remove steady state offset. Finally the derivative gain is introduced to provide response damping and to improve the stability of the system. Figure 3.13 shows the gas turbine local controllers loops. The controllers are tuned directly on the nonlinear model as follows:

First the mechanical power regulating loop is closed and the proportional gain of the PI controller is increased, keeping the integral gain equal to zero, a value of k_p=.8 is obtained at which the output became slightly underdamped. Then the integral gain k_i is increased to 0.1 to reduce the offsets as shown in Figure 3.14 (a, b, c). It can be noted that while tuning is made to the power control loop not only the mechanical output power controlled but also the other outputs i.e. exhaust temperature and *Nox* level are already at their steady state values.

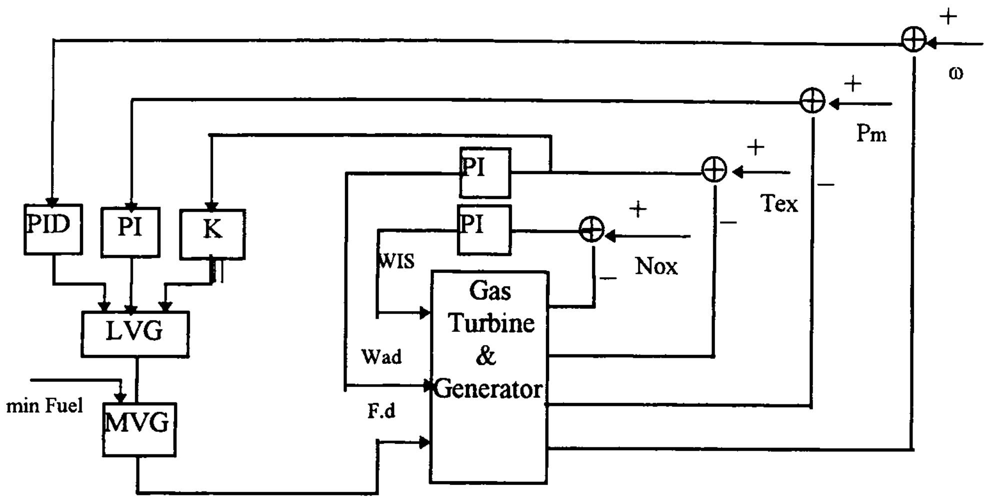


Figure 3.13: Gas turbine local controllers loops.

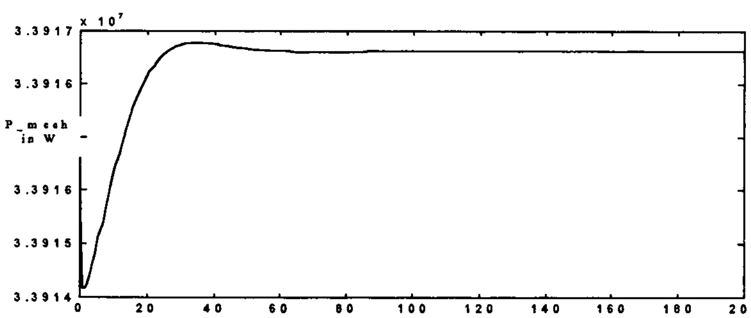


Figure 3.14(a): Mechanical output power at the nominal operating point

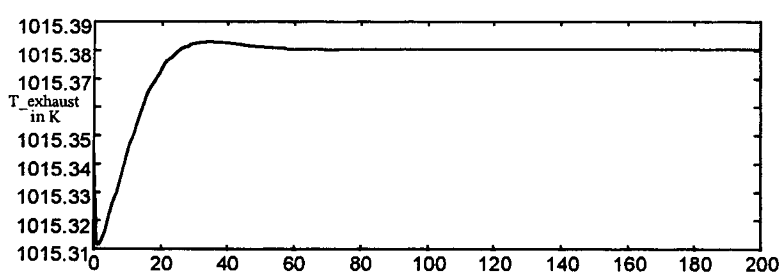


Figure 3.14(b): Exhaust gas temperature around the operating point.

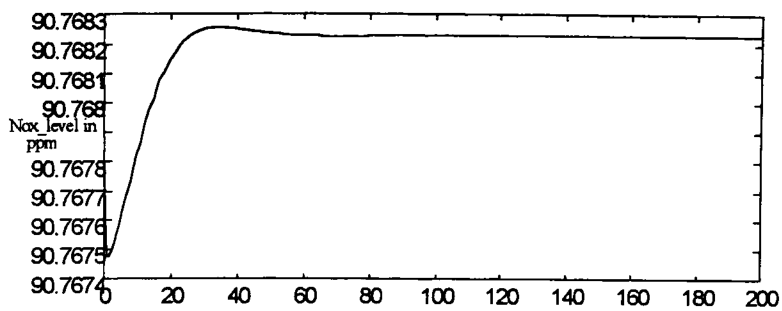


Figure 3.14(c): N_{ox} level content in the exhaust gas

The second loop is the temperature regulating loop. Using PI controller and repeating the same steps as per the mechanical power regulating loop the following controller parameters are obtained k_p=0.2 and k_i=0.4.

Nox regulating loop is controlled with a PI controller and tuned in the same manner as the previous loops to get k_p=0.0009 and k_i =0.008.

The power regulating loop is disconnected and the speed regulating loop is then inserted. PID controller is used for speed regulation. Tuning of this loop is done as before and the following values of the controller parameters are obtained k_p=0.8, k_i =0.2 and k_d=0.01.

Finally the least value gate is inserted to select a minimum value of three control signals namely, temperature control signal, speed control signal, and power control signal. As a result of interaction between the regulating loops fine tuning was carried out and the new settings for the controllers are given in Table 3.1. The output responses to these settings are shown in Figure 3.15 (a, b, c).

Controller parameters / Control loops	k_p	k_i	k_d
P_mechanical	0.5	0.2	0
Speed	0.9	0.4	0.15
Temperature	0.2	0.2	0
Nox	0.000009	0.001	0

Table 3.1 Controller parameters of different control loops.

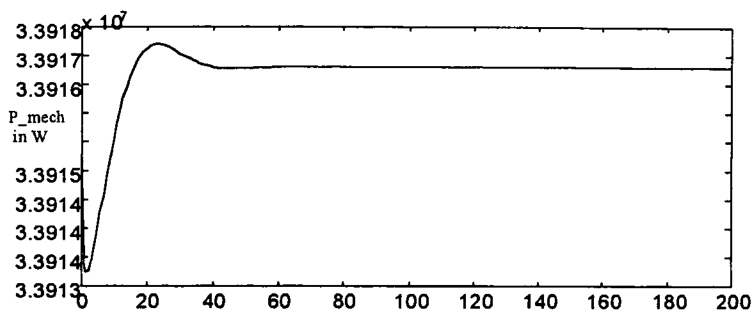


Figure 3.15(a): Mechanical output power tuned at the operating point.

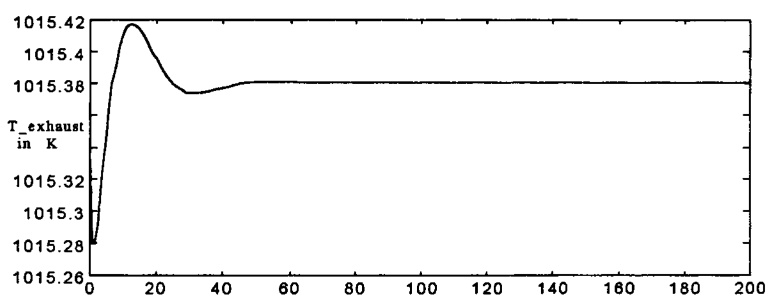


Figure 3.15(b): Exhaust gas temperature tuned at the operating point.

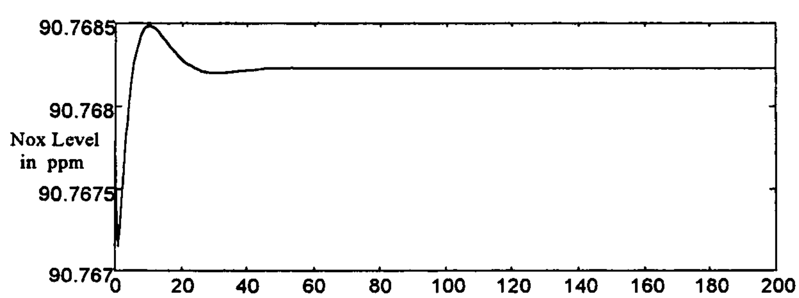


Figure 3.15(c): ***Nox*** **level in the exhaust gases tuned at the operating point.**

3.7 Simulation of the Nonlinear Model

3.7.1 Step and Ramp Reference Inputs

The nonlinear model simulation is made to check the ability of the controllers to achieve tracking of changes in the set-point. Exhaust gas temperature step-input of 10 % of the steady state value is applied to the model. Nox and P_mech reference inputs are kept unchanged. The responses of the controlled outputs, output power, output temperature and Nox_level, are shown in the first column of Figure 3.16 respectively and the manipulated inputs, air demand, fuel demand and injected steam demand, are shown in the second column of Figure 3.16 respectively. To achieve the required increase in the output temperature, with the lowest amount of fuel, while keeping the output power and the Nox_level unchanged, an increase in the fuel demand and a decrease in the air demand are needed. This can be seen easily from the second column of Figure 3.16. For any increase in the fuel demand there is an increase in the pollutant Nox. To keep Nox level unchanged, the injected steam demand must be increased. This can be seen from the last plot in the second column of Figure 3.16.

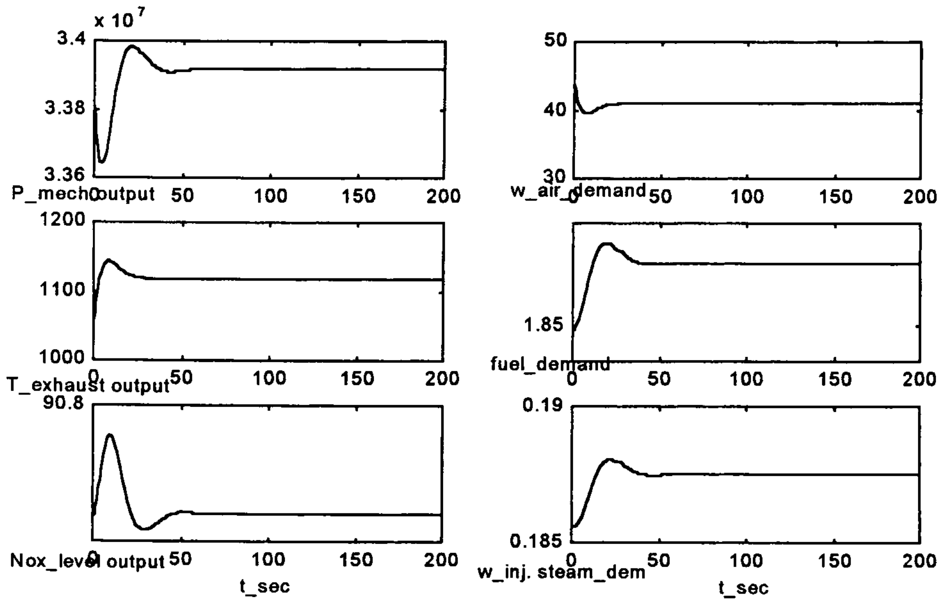


Figure 3.16: Controlled outputs and manipulated inputs for 10% step in T_exhaust input.

The output responses and the manipulated inputs for a 10% ramp in the power set-point input for 200 sec are shown in Figure 3.17. It can be noted that there is an increase in the values of the manipulated inputs. This can be explained as follows:

1. For an increase in the output power a corresponding increase in the fuel demand is required. This can be seen from the second plot in the second column of Figure 3.17.

2. The increase in the amount of combusted fuel leads to an increase in the temperature of the exhausted gases. To compensate for this, and to keep the temperature of the exhaust gases unchanged, the amount of air flow to the combustion chamber should be increased this can be seen from the first plot in the second column of Figure 3.17.

3. As a result of increased fuel and air demand the level of pollutants in the exhaust gases will increase and to compensate for this the amount of injected steam should be increased. This increase in the injected steam demand can be seen from the third plot in the second column of Figure 3.17.

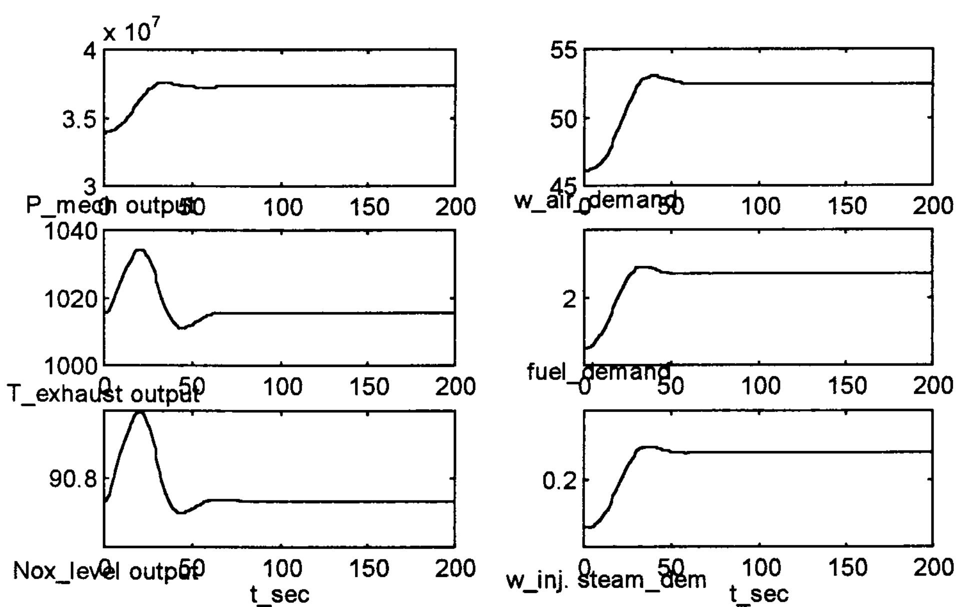


Figure 3.17: Controlled outputs and manipulated inputs for 10% ramp in P_mech. Set-point.

3.7.2 Disturbance Rejection:

The ability of the control design to reject disturbances is tested by applying two types of disturbances namely, step disturbances (process disturbances) at the input of the power turbine and band limited white noise (output disturbances) at the outputs. Column one of Figure 3.18 shows the power turbine outputs and the injected step disturbances. The step disturbance injected at the fuel demand input at 250 sec is rejected quickly. The step disturbance injected at the air flow demand input at 350 sec is also rejected quickly but caused a significant interaction at the output temperature response which is also rejected quickly. The rejection of the step disturbance for the steam demand at 150 sec is slow compared with the other loops. The second column of Figure 3.18 shows the output responses for step (process disturbance) and white noise (output disturbance) disturbances. The fluctuation in the output mechanical power is due to the white noise which represents the fluctuations in the electrical load. The third plot in the second column of Figure 3.18 shows slow tracking of the reference input as well as the disturbance. This is of less importance because it is usually desired to keep Nox level at its minimum value. Thus it can be seen that good disturbance rejection is achieved by the control design and thus good performance is obtained.

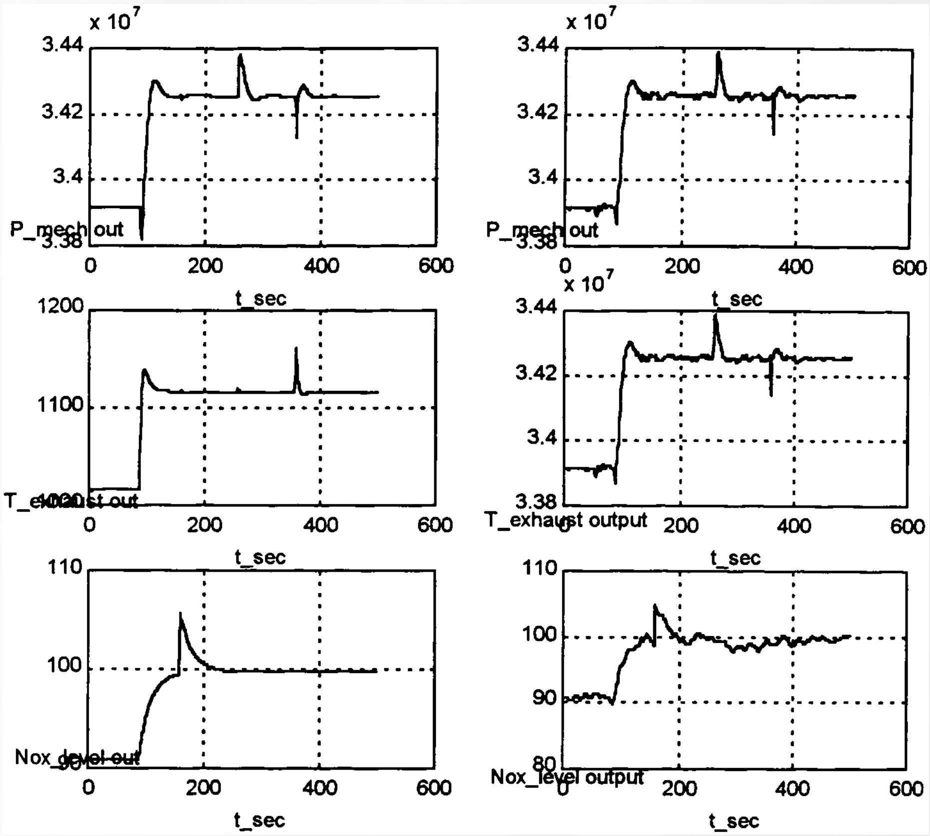


Figure 3.18: Disturbance rejection.

3.8 The Linear Model

Most physical systems have nonlinearities. These can be due to the presence of, dead band, saturation, relays, etc., or due the presence of square roots, exponentials, product, orders higher than one, etc., in the process or in the formulation of the process model. These nonlinearities can be inherent in the model of the system, or added for improving the system performance. Nonlinear systems are difficult to solve and most of the developed theory and analysis deals with linear systems. Thus, nonlinear systems are usually approximated with equivalent linear systems. The linearized model must capture the main system dynamics. The complexity of the linear model depends on its intended application. Linearization is the process of approximating a set of nonlinear differential equations by a linear set.

There exists in the literature different techniques to obtain linearized dynamic models. Taylor series expansion is one of the basic techniques used to get linearized models around an operating point by truncation of high order terms of the series. MATLAB offers different methods to extract the linear model. The following functions are provided in SIMULINK to find a linearized model [63]:

- LINMOD function obtains the state space linear model of a system of ordinary differential equations described in S-function (system function). S-function may be a MATLAB m.file or SIMULINK blocks. Different options are available for instance linear models for systems with state variables and inputs set to zero can be obtained, also options for linearization about an operating point.

- LINMOD2 this function extracts a linear model from systems of ordinary differential equation using an advanced method to reduce truncation error. It provides the linearized model in state space with the same options as LINMOD.

- DLINMOD function obtains the discrete state space linearized model of systems of ordinary differential equations with options as that of LINMOD command.

The gas turbine model is linearized around the operating point (P_{mec}=3.391e^7 W, T_{tout}=1015.37 °K and $Nox = 90.768$ ppm) using the SIMULINK command (LINMOD). The linearized model obtained had 11 states which was reduced to ten states using (MINREAL) command which extracts the minimum realisable linear model. Comparison of the nonlinear and the linear model step responses is made with a 10% step in the exhaust temperature set-point and the results are shown in Figure 3.19 (a, b, c). The nonlinear response is the dashed line and the linear response is the continuous line. Both models reached steady state at the same time but with differences in the transient responses. From Figure 3.19 (b) it can be seen that very small errors in the overshoot exists which are neglected. Figures 3.19 (a) and 3.19 (c) show the presence of small differences between the linear and the nonlinear response.

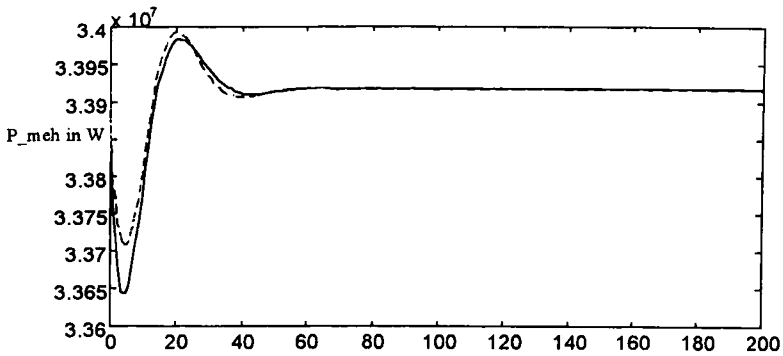


Figure 3.19(a): Mechanical output power for linear and nonlinear models.

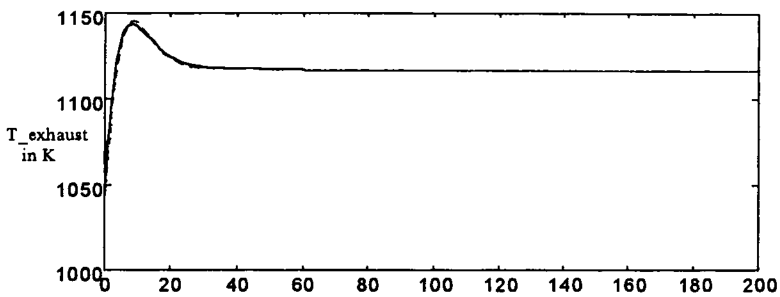


Figure 3.19(b): T_exhaust output for linear and nonlinear models.

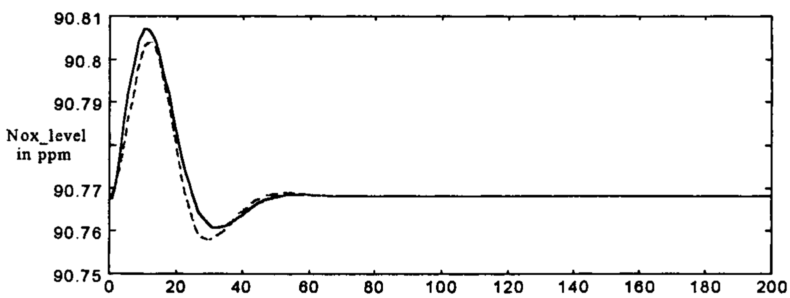


Figure 3.19(c): *Nox* output level for linear and nonlinear models.

3.9 Model reduction

MATLAB control tool box provides the necessary commands to perform model reduction by the elimination of the unwanted states. The states that can be eliminated are those considered as infinitely fast. These states can be eliminated in selected combinations to reduce the error in the reduced model responses.

Using the MATLAB model reduction command (MODRED) the linear model of the gas turbine power plant is reduced to the following orders, 8, 7, and 6 respectively. The responses of the reduced models are compared with the responses of the full order model (10 states) as shown in Figure 3.20 (a, b, c). The lowest possible approximation of the full order model as can been seen from step responses is the sixth order model where the reduced model can still capture the model dynamics with acceptable errors. The full order and the reduced order models matrices are given in Appendix 3.B.

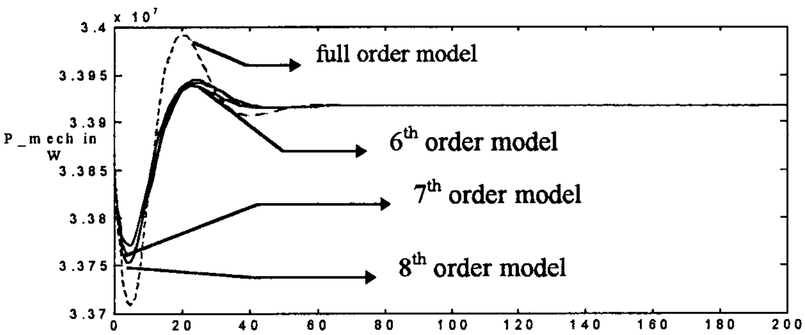


Figure 3.20(a): Comparison of responses of the reduced linear models with the full order model.

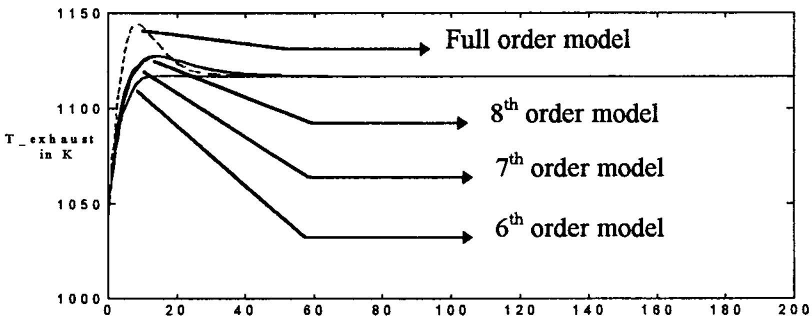


Figure 3.20(b): Comparison of responses of the reduced linear models with the full order model.

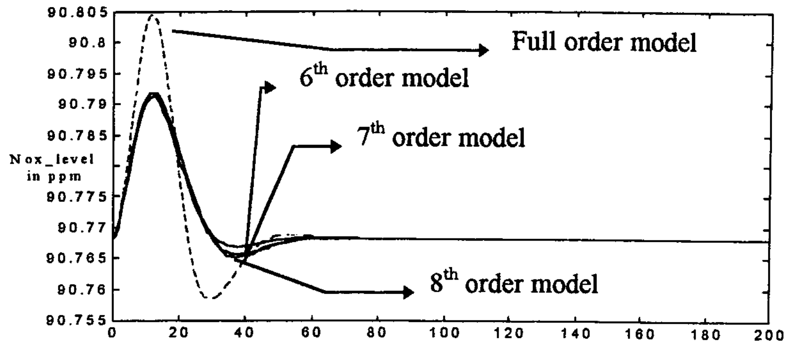


Figure 3.20(c): Comparison of responses of the reduced linear models with the full order model.

3.10 Conclusions

A gas turbine power plant model was presented and implemented in SIMULINK. Local PID controllers were designed to stabilise the model around a nominal operating point. It has been found that a simple tuning method (manual method) was effectively sufficient to tune the controllers. Simulation of the nonlinear and the linear model showed that the control system has achieved good reference input tracking and disturbance rejection. Linearization was made and a linear model was extracted using SIMULINK facilities. Responses of both the nonlinear and the linear models were compared and simulation showed that the linear model obtained is a good approximation of the nonlinear model for the purpose of control design. The linear full order model was reduced and an acceptable sixth order model was obtained.

Chapter four

GPC Design for a Gas Turbine Power Plant

4.1 Introduction

This chapter is concerned with the design of a second layer of controllers for power turbines steady state operation. Start-up and shut-down operations will be considered in the next Chapter. A multivariable GPC controller is applied as a second layer add-on controller to the PID controlled nonlinear model of the gas turbine to optimally change the set-points of the regulator loops such that a quadratic cost function is minimised. This arrangement can be considered as a part of the supervisory control system where the local controllers set-points are adjusted according to the improved plant operating conditions.

Computer programs periodically review the plant operating conditions to find new improved plant operating conditions. These new operating conditions are used along with the plant operating and economic models for the optimisation of a certain objective function [64]. The work in this chapter is restricted only to the design of the second layer controller assuming that the inputs to this controller are the optimal operating conditions. A GPC supervisory control system that uses predictive control strategy will be designed to generate local controllers set points for the gas turbine described in the previous Chapter. The multivariable state space

GPC algorithm presented in Chapter two is used for the controller design. A comparison with conventional supervisory control system which uses a suitable ramping mechanism to generate local controllers set-points will be made wherever possible.

Detailed analysis and design will be devoted only to the predictive control strategy. GPC closed-loop formulation developed in Chapter two is applied here to the supervisory control system to investigate the performance robustness and the stability robustness through frequency domain indicators and Singular value Bode plots. The user defined GPC tuning parameters are identified to globally tune the controller to ensure good performance robustness and stability robustness as well as the desired nominal performance.

The capabilities of the second layer controller to handle constraints will be considered. In fact, most real systems are subject to constraints that include constraints imposed as a result of operating conditions and those imposed due to economic factors. Thus a case study including operating constraints, constraints on exhaust temperature, constraints on output mechanical power, constraints on the control signals etc., will be presented. On one hand the effects of physical constraints such as, control increment and control signal constraints on the performance of the control system are studied. On the other hand user defined constraints such as, constraints on overshoot and on the process outputs are investigated to improve the control performance and robustness.

4.2 Conventional and GPC Supervisory Control Systems

The conventional supervisory control system for a single unit gas turbine power plant is shown in Figure 4.1. In conventional supervisory control, set-points are generated through a suitable ramping mechanism which changes the set-point slowly to avoid excessive overshoots and any undesired dynamic behaviour in the system. A fast response for fast changes in the set-points is desirable. However, as a result of the fast set-point changes undesirable overshoots usually happen in the outputs which in turn may cause unsafe operating conditions. This is usually the case with the conventional supervisory control if fast ramping signals are used.

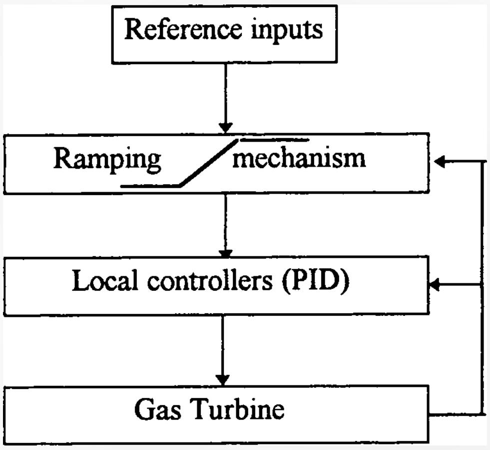


Figure 4.1: Conventional supervisory control structure.

The drawbacks of the conventional supervisory control can be avoided by replacing the ramping mechanism with a predictive controller which has the capability of providing faster and smoother responses to fast set-point changes with controlled overshoots. Also constraints on manipulated input and output variables can be handled with GPC which can not be handled using the PID control technique. A schematic diagram of GPC supervisory control for a single unit power plant is demonstrated in Figure 4.2.

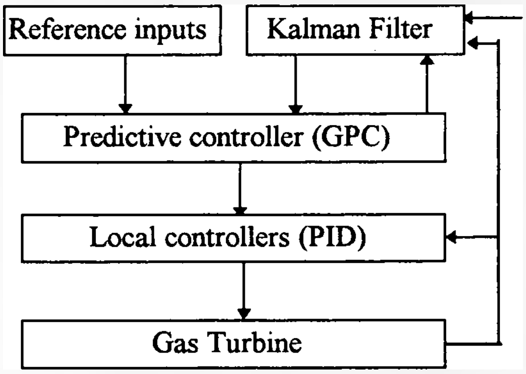


Figure 4.2: The GPC supervisory control structure

4.3 GPC Supervisory Control System Design

Before embarking on the design of the multivariable GPC supervisor a description of the process and the predictive controller is reviewed.

4.3.1 Process and Control System Description

Gas turbine power plant operation comprises three regimes: start-up, shut-down and steady state operations as described in the previous Chapter. The aim of the local controllers is to stabilise the plant at an operating point. The gas turbine control system considered in the previous Chapter is intended for normal operation, steady state operation, and is composed of local PID regulators. The gas turbine block diagram is shown in Figure 4.3. The control inputs and the turbine outputs are as follows:

- Fuel flow to provide the desired output power and frequency.
- Injected water flow to reduce emission of Nox.
- Control of inlet guide vanes to provide the desired output temperature.

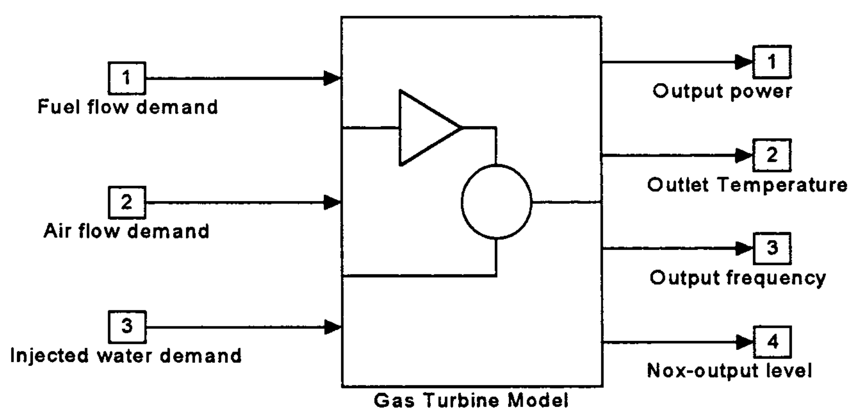


Figure 4.3: Power gas turbine model.

The local control scheme consists of four regulating loops, namely, power regulator, frequency regulator, exhaust gas temperature and Nox regulator. Fuel flow is adjusted to get the correct output power and frequency. The fuel flow rate is selected as the minimum output of the first three mentioned regulators. The regulator loops are closed and fine tuned as described in the previous Chapter.

The second layer of control is introduced as an add-on to the regulating loops to optimally manoeuvre the gas turbine from one operating point to a new one. The predictive controller inputs are the power turbine controlled outputs, the mechanical output power, the exhaust gases temperature and the Nox pollutant level. The differences between these controlled outputs and the desired set-point values are minimised over the prediction horizon by optimising a quadratic objective function. This results in optimised set-points i.e., GPC controller outputs, which are the inputs to the local controllers. Kalman filter is used for state estimation. The SIMULINK model of GPC supervisory control system for power gas turbine is shown in Figure 4.4.

4.3.2 The Augmented Linear Model

Since the closed-loop linear model of the gas turbine power plant was derived based on the control signal vector u and since the GPC control law equation (2.14) and the Kalman filter state estimator equation (2.20) are derived based on the control signal increment thus, the linear discrete time state space model based on u will be modified to include the control signal vector u as additional states [20]:

$$x_{k+1} = Ax_k + Bu_k + Ew_{wk} \tag{4.1}$$

$$y_k = Cx_k + Du_k + v_k \tag{4.2}$$

$$\Delta u_k = u_{k+1} - u_k \tag{4.3}$$

Rearranging this equation gives,

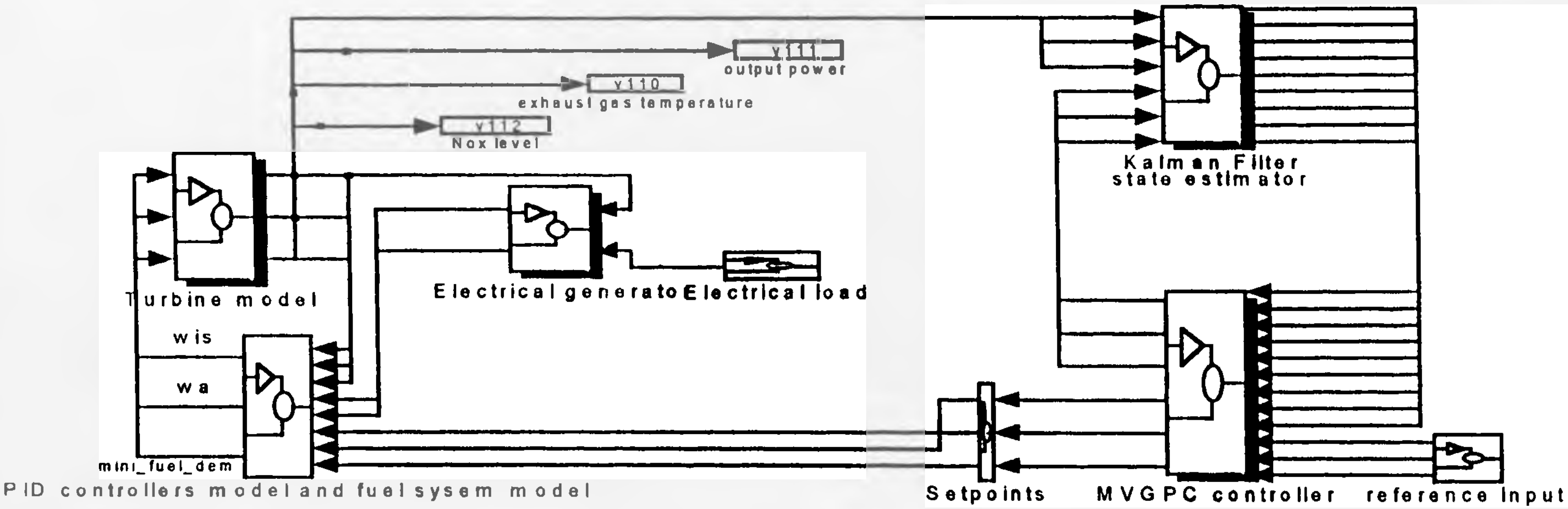


Figure 4.4: GPC supervisory control system for power turbine.

$$u_{k+1} = \Delta u_k + u_k$$

Equations (4.1), (4.2) and (4.3) can be combined to give

$$x_{k+1} = Ax_k + Bu_k + Ew_{wk}$$

$$u_{k+1} = I\Delta u_k + Iu_k$$

$$y_k = Cx_k + Du_k + v_k$$

which can be written as follows:

$$x_{k+1} = \widetilde{A}x_k + \widetilde{B}\Delta u_k + \widetilde{E}w_{wk} \tag{4.4}$$

$$y_k = \widetilde{C}x_k + \widetilde{D}\Delta u_k + v_k \tag{4.5}$$

where,

$$x_{k+1} = \begin{bmatrix} x_{k+1} \\ u_{k+1} \end{bmatrix}, \quad x_k = \begin{bmatrix} x_k \\ u_k \end{bmatrix}$$

$$\widetilde{A} = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix}, \quad \widetilde{B} = \begin{bmatrix} 0 \\ I \end{bmatrix} \tag{4.6}$$

$$\widetilde{C} = \begin{bmatrix} C & D \end{bmatrix}, \quad \widetilde{D} = [0], \quad \widetilde{E} = \begin{bmatrix} E \\ 0 \end{bmatrix}$$

This augmented state space linear model will be used for designing both GPC controller and the Kalman filter state estimator. For producing the desired state estimates the process equations must satisfy the conditions of observability. Since it was the gas turbine power plant linear model is both controllable and observable this implies that the augmented model is also controllable and observable.

made. In our design the aim is to find the best sets of GPC tuning parameters (N1, N2, Nu) and weighting matrices Γ and Λ to achieve a stable controller, a reduction of interaction between different channels, set-point optimisation, good disturbance rejection and improvements to the stability and performance robustness. Since the control design objectives are satisfied through the controller tuning parameters and since it is very difficult to achieve all the required objectives at the same time, a comprehensive study of the effects of various tuning parameters on the behaviour of the controller is needed. For this reason two studies will be carried out. The first one is the performance study and the second one is the robustness study. Through these studies a compromise is made to obtain the final control design.

4.5 Supervisory Controller Performance Study

This study includes step response analysis, set-point decoupling and disturbance rejection. The aim of this study is to investigate the performance of the control design over a wide range of tuning parameters such that a compromise among different design objectives can be made.

4.5.1 Step Response Analysis

The values of the tuning parameter and weighting elements of Table 4.1 are used as a guide for tuning the controller on the nonlinear model and to perform stability and performance robustness study. After many tuning trials and tests including nonlinear step response test, performance robustness tests, stability robustness tests and constraints handling, a new set of tuning parameters and weighting elements, Table 4.2, is selected. The tuning parameters and weighting matrices elements used in the following analysis are varied over a wide range, around the selected set to examine the effects of each parameter on the dynamic behaviour of GPC supervisory control system and to see the way the performance can be improved. The aim of this investigation is to find the best set of tuning parameters that satisfy the design objectives for final design stage. The basic nominal performance test is the step response analysis. Nominal performance indicators are, rise time τ_r, settling time ts, percentage overshoot %o.s and time delay td. These indices will be measured each time a tuning or weighting parameter is changed.

Different possible combinations of the tuning parameters to achieve the desired performance can be found. In the following analysis the tuning parameters and weighting matrices elements are fixed except for the one under investigation which is allowed to vary in a specified range.

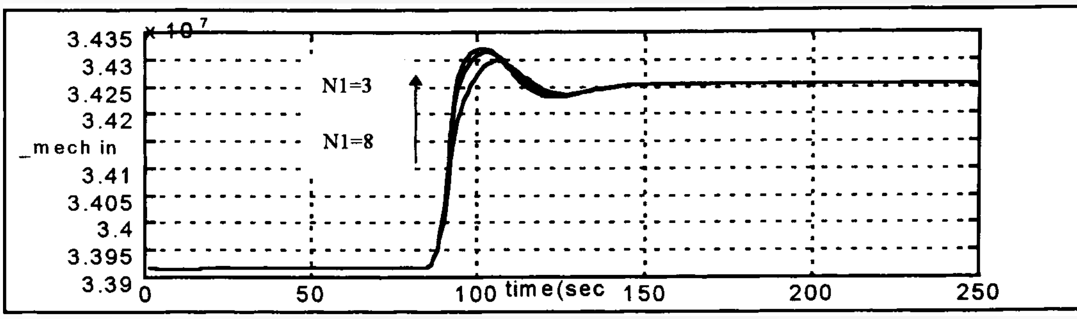
GPC tuning parameters	Weighting on control increments (λ)	Weighting on tracking error (γ)
$N1$=5	λ_1=0.00000001	γ_1= 0.0000000007
$N2$=19	λ_2= 22	γ_2=0.55
Nu=11	λ_3=0.3	γ_3=5

Table 4.2 GPC tuning and weighting parameters.

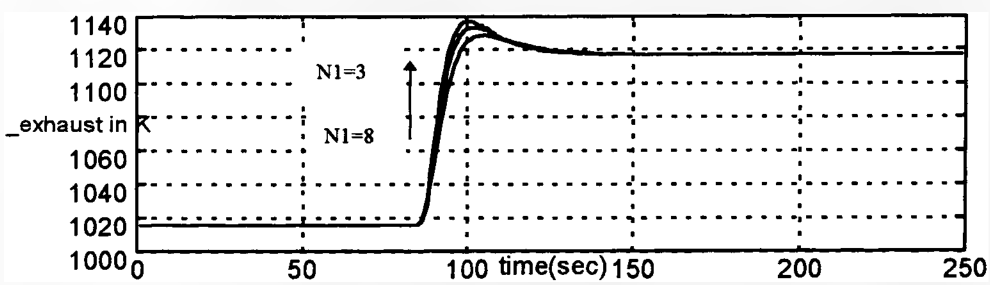
4.5.1.1 Effects of N1

The graphs that are shown in Figure 4.8 (a, b, c) illustrate the effects of changing the minimum output horizon in the range from 3 to 8 on the dynamic performance of GPC supervisory control system. The %overshoot in the output power is 18.18% for N1=3. This %overshoot can be decreased to 12.2% when N1 is increased to 8. Settling time and time delay remain unchanged at 51.78 sec and 7.4 sec, respectively while the rise time is increased from 5.9 sec to 8.87 sec for the same increase in N1.

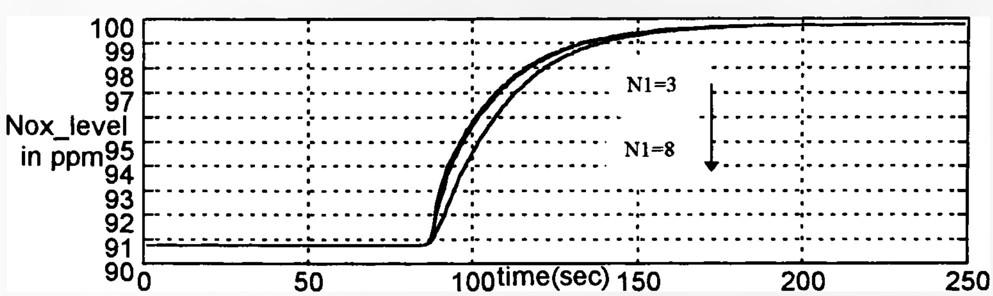
In the output exhaust gas temperature an %overshoot of 20% is obtained for N1=3 which can be reduced to 11% for N1=8. Approximately the same settling time of 31.36 sec and time delay of 5.91 sec are obtained for the same increase in N1. Rise time is increased from 4.14 sec for N1=3 to 6.21 sec for N1=8. The Nox output level becomes more overdamped as N1 increases with longer settling time (62.42 sec), rise time (44.37 sec) and time delay (17.75 sec) are obtained for N1=8.



(a)



(b)

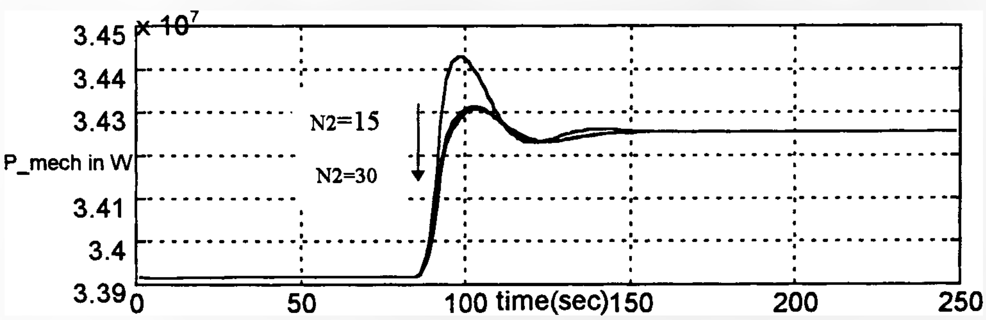


(c)

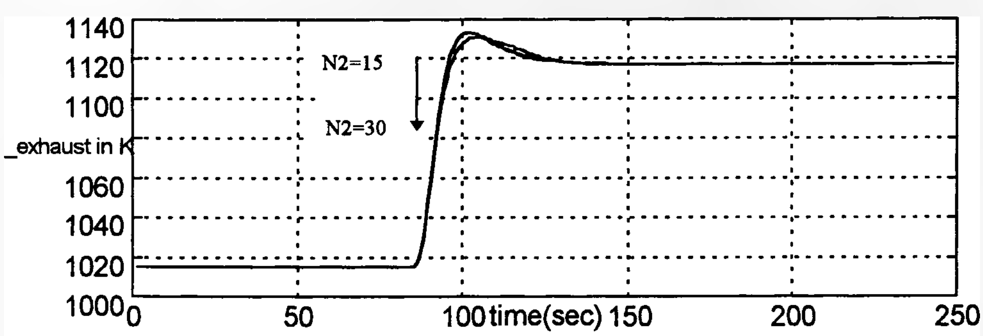
Figure 4.8(a, b, c): Effects of changing N1 on the performance of GPC supervisory control.

4.5.1.2 Effects of N2

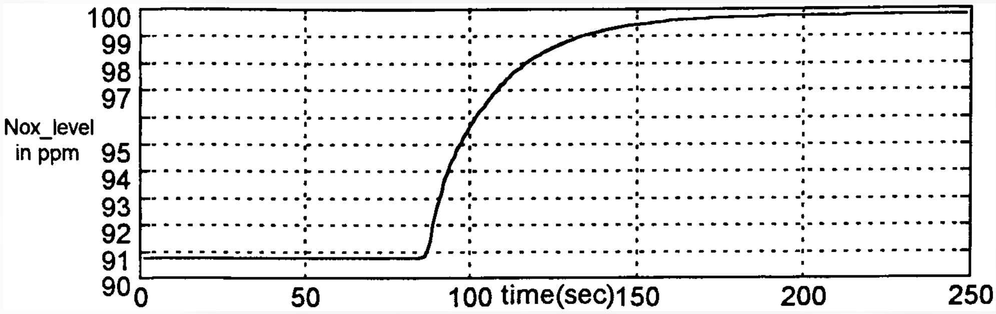
The dynamic behaviour of the GPC supervisory control as a result of increasing the maximum output horizon from 15 to 30 is shown in Figure 4.9 (a, b, c). The main observation is the dramatic overshoot reduction in the output power from 50.3% to 15.1% when N2 goes from 15 to 30. Settling and rise times are slightly affected for the same increase in N2. The improvement in the output exhaust gas temperature response that can be noticed as N2 increases from 15 to 30 is the %overshoot reduction from 15.92% to 12.4%. Nox output level response remained unchanged.



(a)



(b)

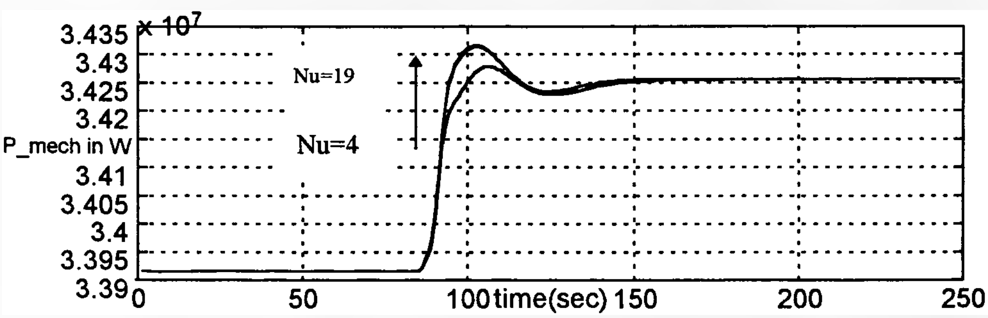


(c)

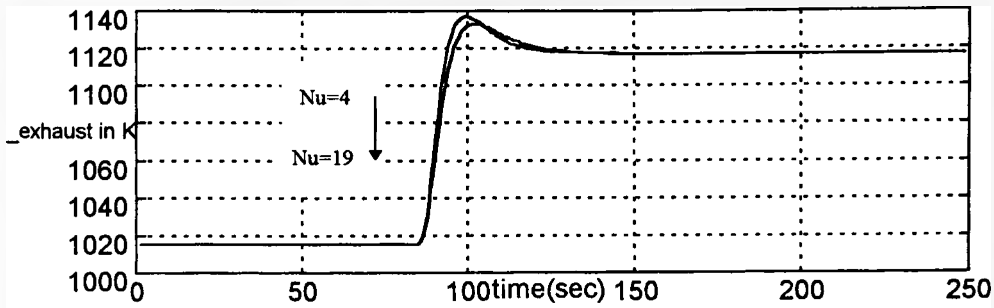
Figure 4.9(a, b, c) Effects of changing N2 on the performance of GPC supervisory control.

4.5.1.3 Effects of Nu

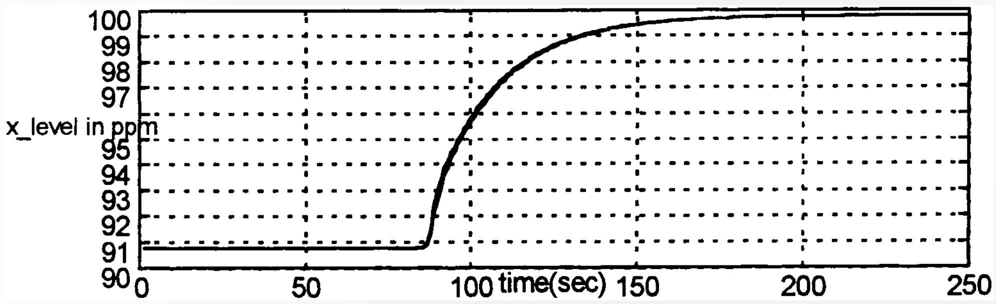
The control horizon Nu has opposite effects on the output power and the output exhaust temperature. On one hand, an increase in Nu from 4 to 19 causes an overshoot increase from a approximately 5 % to 16.96 % in the output power response, a decrease in both the settling time from 53.5 sec to 47.94 sec and the rise time from 8.83 sec to 4.41 sec and time delay remains unchanged for both cases. On the other hand, for the same increase in the control horizon, a drop in the %overshoot from 21.62 % to 16.2 % in the output exhaust temperature is obtained, and an increase in both the settling time from 27.97 sec to 32.39 sec and the rise time from 5.89 sec to 7.36 sec. Nox output level approximately remains unchanged change. This is illustrated in Figure 4.10 (a, b, c).



(a)



(b)



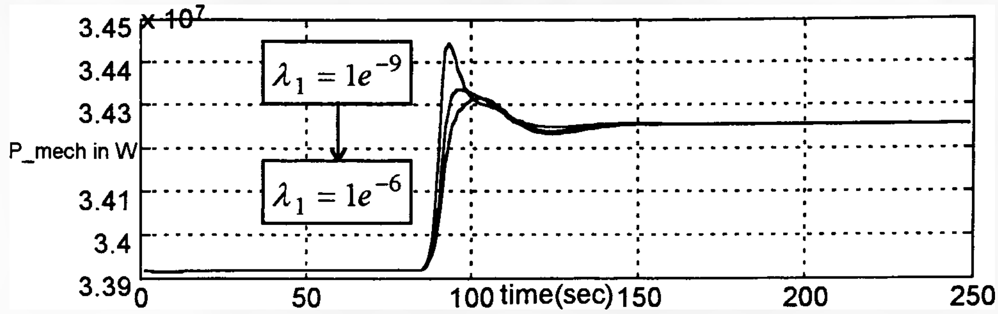
(c)

Figure 4.10(a, b, c): Effects of changing Nu on the performance of GPC supervisory control.

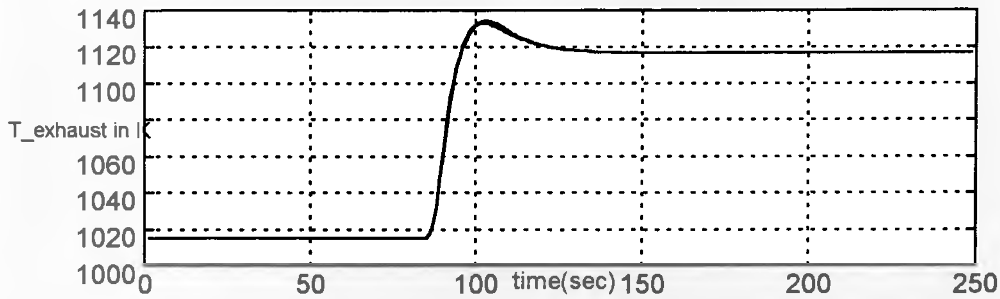
4.5.1.4 Effects of Weighting Elements of Λ and γ on the Performance

4.5.1.4.1 Effects of the Weighting on Power Control Increment (λ_1)

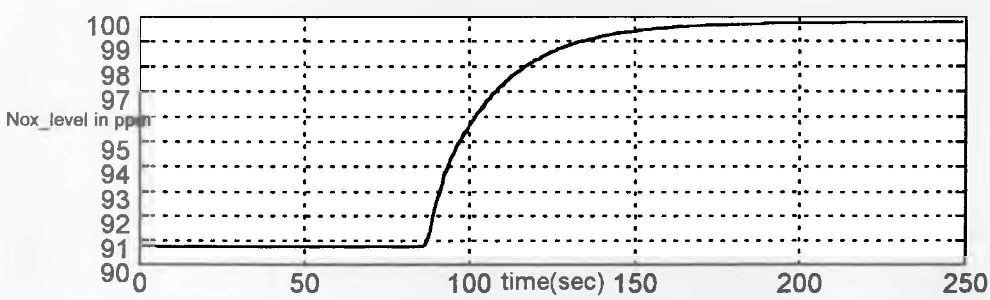
When the weighting element λ_1 on the mechanical power control signal increment Δu_p is increased from $1e^{-9}$ to $1e^{-6}$, a %overshoot decrease from 54.08 % to 17.92 % in the output power, an increase in the settling time from 27.35 sec to 46.76 sec and an increase in the rise time from 3.23 sec to 5.88 sec are obtained. A shift in the time of the overshoot peak can be seen from Figure 4.11 (a). The output exhaust temperature and Nox output level remains unaffected as shown in Figure 4.11 (b) and Figure 4.11 (c).



(a)



(b)



(c)

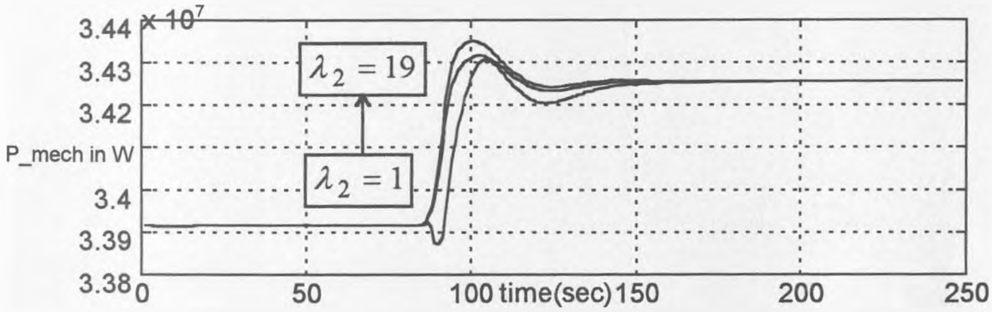
Figure 4.11(a, b, c) Effects of changing λ_1 on the performance of GPC supervisory control.

4.5.1.4.2 Effects of the Weighting on the Exhaust Temperature Control Increment (λ_2)

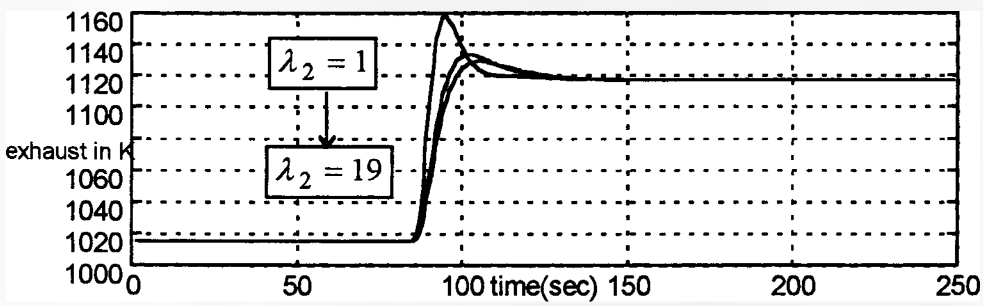
The importance of the weighting element λ_2 on Δu_t is to eliminate the 12.69% undershoot measured with respect to the new operating point in the output power as λ_2 is increased from 1 to 19. This will, however cause an increase in the

%overshoot from 15.71% to 28.82%. However, an improvement in the speed of response can be easily observed from Figure 4.12 (a) that is, the time delay is reduced from 5.88 sec to 3.82 sec, the rise time is slightly decreased and the settling time is greatly shortened (54.44 sec to 28.82 sec).

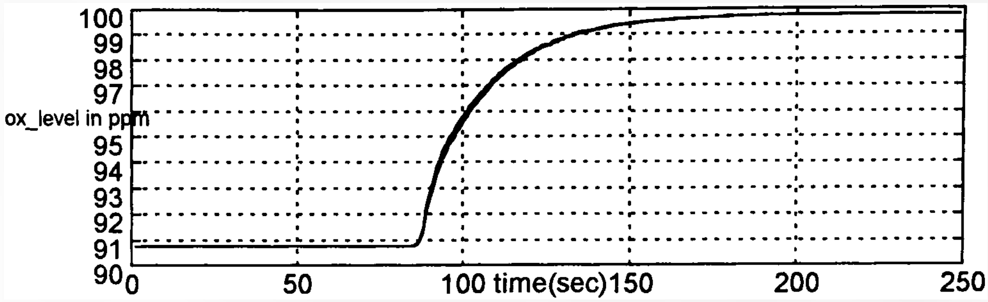
The effect of λ_2 on the output exhaust temperature is just the opposite where, at $\lambda_2=1$ there is a large overshoot peak of 39.65% which is decreased to 12% for $\lambda_2=19$. Corresponding to this there is, an increase in the settling time from 20.58 sec to 32.35 sec, an increase in the rise time from 4.1 sec to 8.82 sec and an increase in time delay 2.94 sec to 5.88 sec. These can be seen from Figure 4.12(b).No significant effects can be seen on Nox output level as shown in Figure 4.12(c).



(a)



(b)

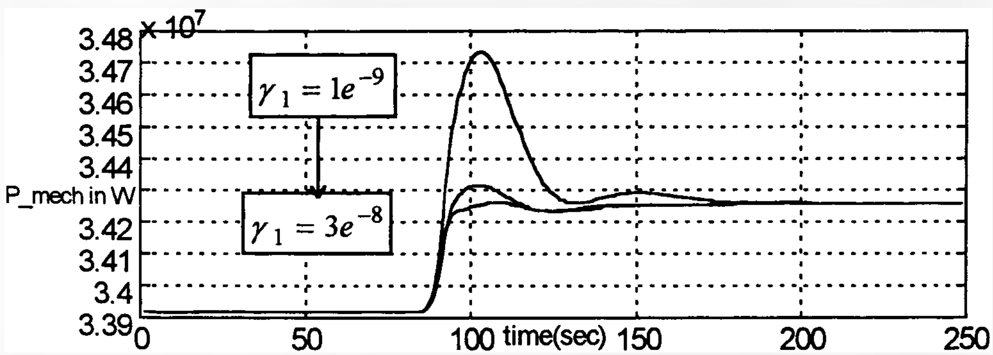


(c)

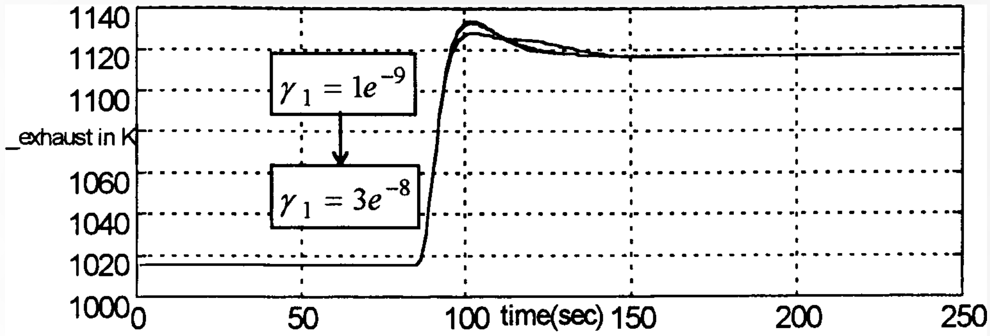
Figure 4.12(a, b, c) Effects of changing λ_2 on the performance of GPC supervisory control.

4.5.1.4.3 Effects of the Weighting on the Mechanical Output Power (γ_1)

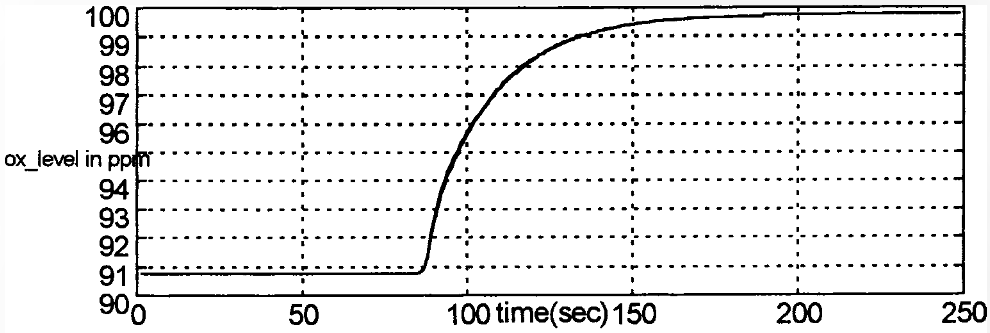
The range over which γ_1 is allowed to change is $1e^{-9}$ to $3e^{-8}$. The overshoot in the output power response is disappeared when γ_1 is made equal to $3e^{-8}$ as shown in Figure 4.13 (a). However, a drastic increase in the %overshoot is occurred for $\gamma_1=1e^{-9}$. It can be seen from Figure 4.13 (b) that there is a decrease in the overshoot in the output exhaust temperature as γ_1 is decreased to 10^{-9}. Nox output level remains unaffected.



(a)



(b)



(c)

Figure 4.13(a, b, c): Effects of changing γ_1 on the performance of GPC supervisory control.

4.5.1.4.4 Effects of the Weighting on the Exhaust Temperature Output (γ_2)

The %overshoots in output power and in the output exhaust temperature are decreased from 34.92% to 18.38% and 26.41% to 16.98% respectively when γ_2 is decreased from 2 to 0.2. Nearly no changes in the settling times of both the output power and the output exhaust temperature can be observed. However, observable changes in the rise times of both the output power and the output exhaust temperature can be seen from Figure 4.14 (a) and Figure 4.14 (b). No change in the output Nox level has happened.

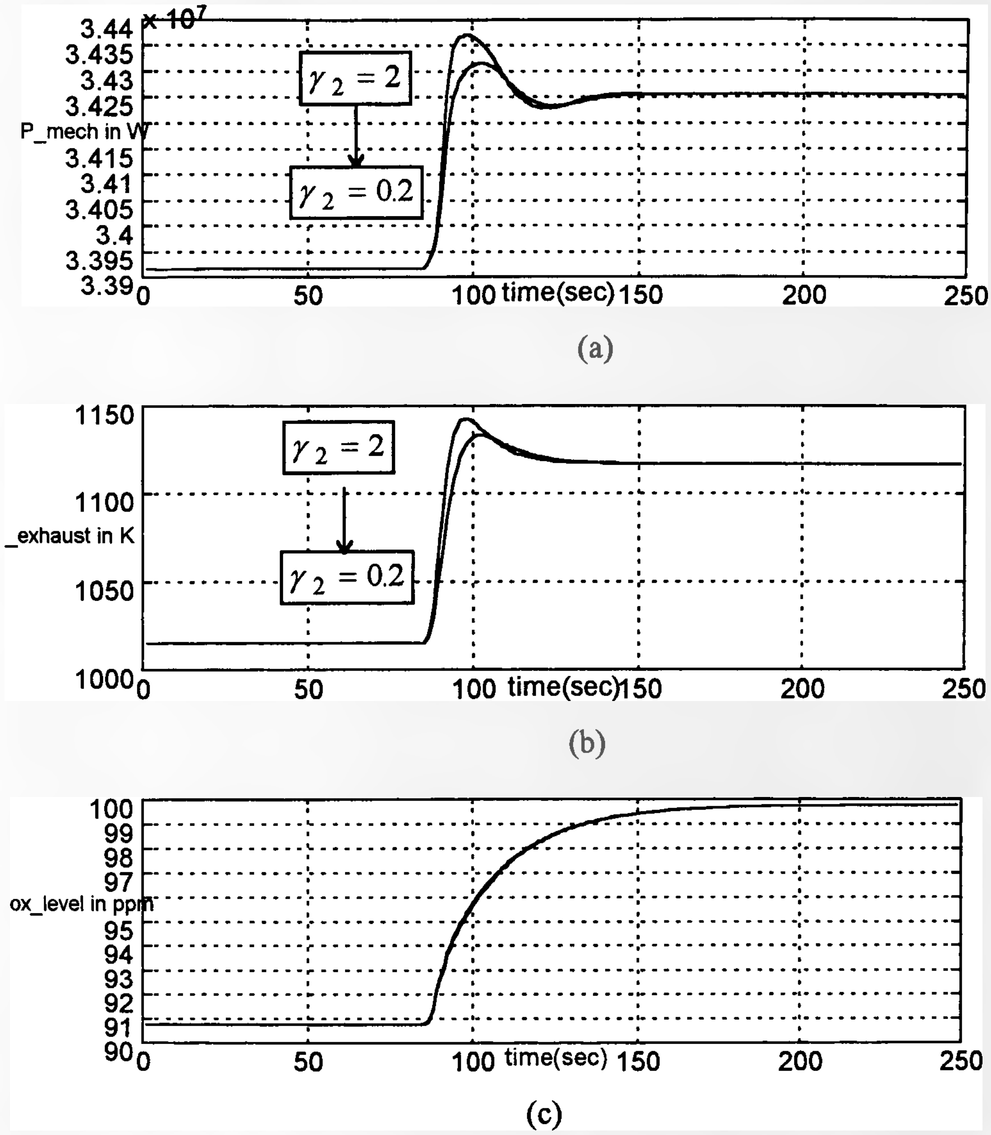


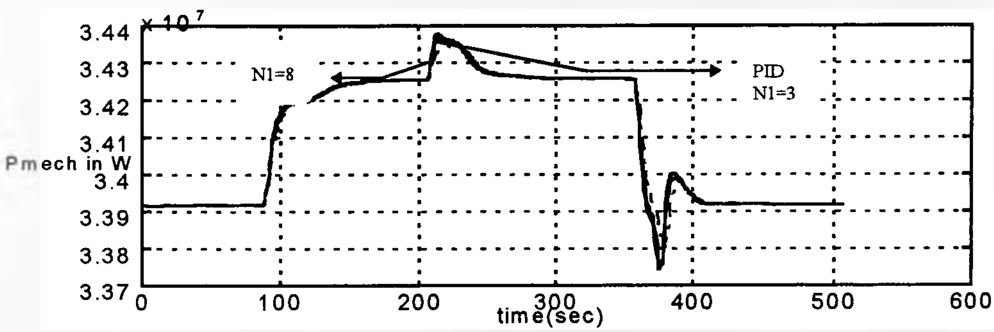
Figure 4.14(a, b, c): Effects of changing γ_2 on the performance of GPC supervisory control.

4.5.2 Set-point Decoupling

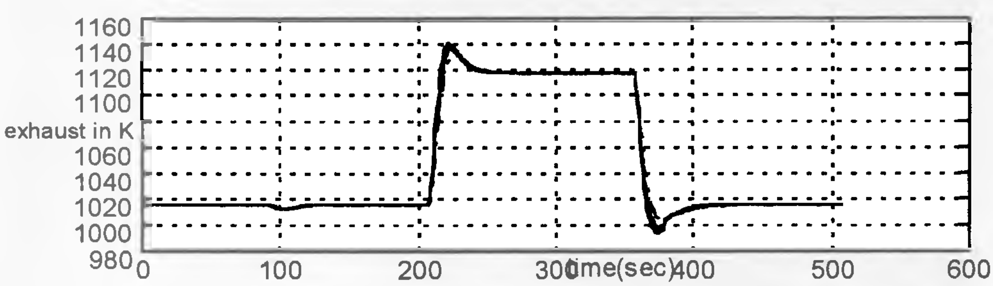
It is known that GPC can guarantee reference signal decoupling due to the feedforward part in its control law. Here the improvements of reference signals decoupling that can be achieved through various tuning and weighting element of GPC controller will be investigated and the results will be compared with that of

PID controllers. The tuning parameters and weighting elements are varied around the values given in Table 4.2. This is to investigate the effects of each parameter on set-points decoupling in comparison with PID controller and to make a compromise at the final stage of the controller design.

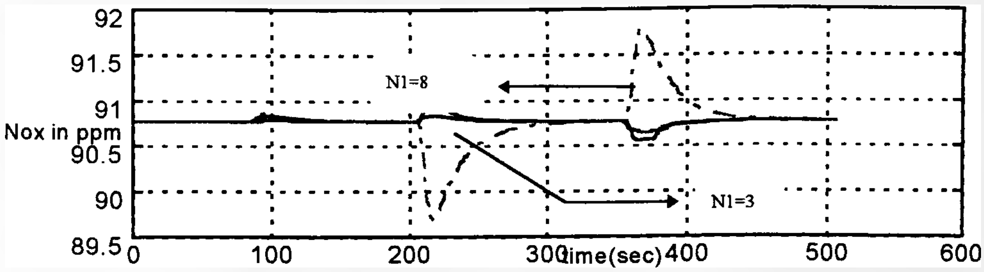
The minimum output horizon N1=8 caused an improvement in set-points decoupling where the %overshoot due to exhaust temperature switching and the %overshoot in the falling edge of the output power are decreased to 27.41% and 30.64% respectively. Set-points interaction in the Nox output level is remarkably increased when N1=8 and slightly decreased when N1=3. This can be seen in Figure 4.15 (a, b, c).



(a)



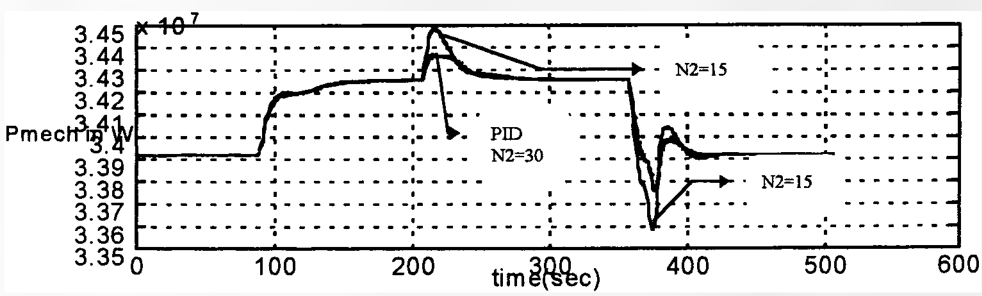
(b)



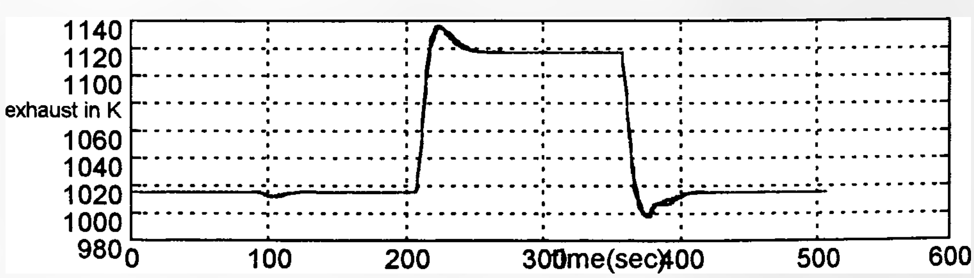
(c)

Figure 4.15(a, b, c): Set-point decoupling when N1=3 and 8 compared with PID.

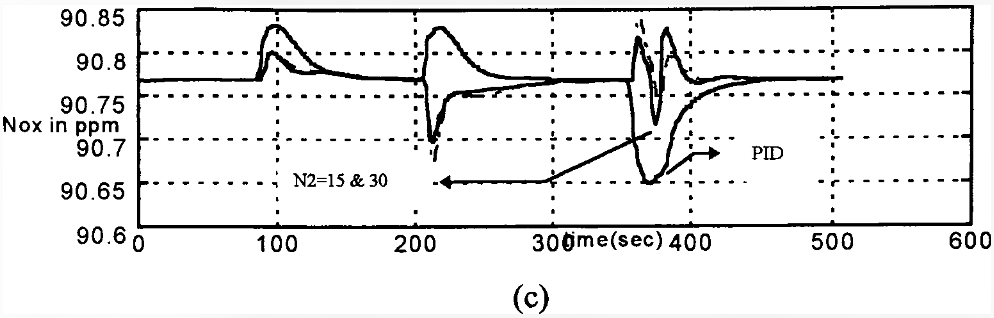
The maximum output horizon N2 is assigned two values 15 and 30. Figure 4.16 (a, b, c) shows that when N2=15 there is 68% overshoot due the set-points interaction in the output power compared with only 33.35% for PID and Nu=30. In the falling edge of output power response both PID and GPC with N2=30 approximately give the same %overshoot while N2=15 give a dramatic increase in the percentage overshoot (97.8%). No effects on the interaction in the output exhaust temperature can be seen. Both Nu=15 and 30 give the same effects, on the Nox output level, of improved decoupling.



(a)



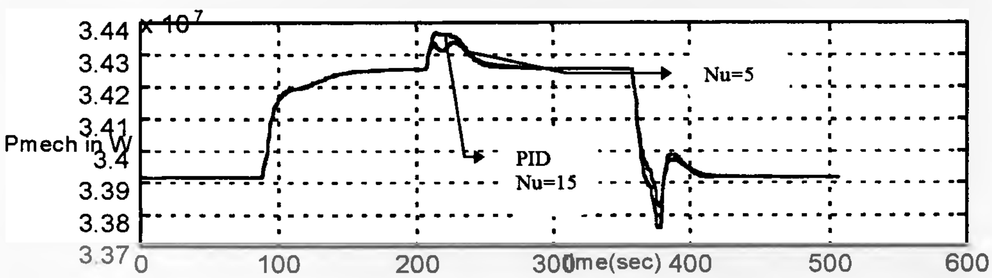
(b)



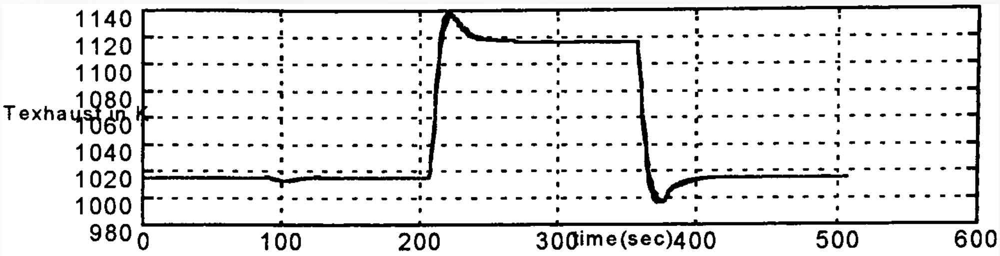
(c)

Figure 4.16(a, b, c): Set-point decoupling when N2=15 and 30 compared with PID.

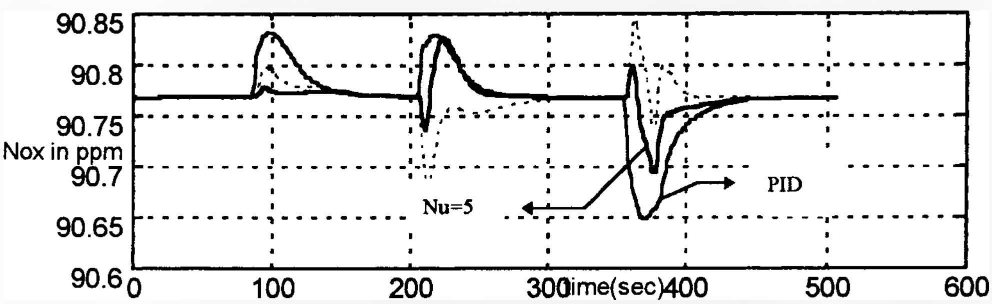
The mechanical power and the exhaust temperature set-points are switched at different times and the Nox level is kept constant at its minimum value. The control horizon Nu is increased from 5 to 15 and the GPC step responses at these values as well that of PID are compared in Figure 4.17 (a, b, c). PID and GPC output power responses coincide for Nu=15 and better set-point decoupling is obtained in the output power when Nu=5 and in terms of %overshoot a decrease from 33.35% (PID) to 23.48% due to exhaust temperature set-point switching and from 46.97% (PID) to 26.84% in the falling edge of the output power response. In the Nox output level the effects of Nu on set-points decoupling are more clear where the overshoots are better attenuated for Nu=5 compared with PID. This is illustrated in Figure 4.17(c).



(a)



(b)

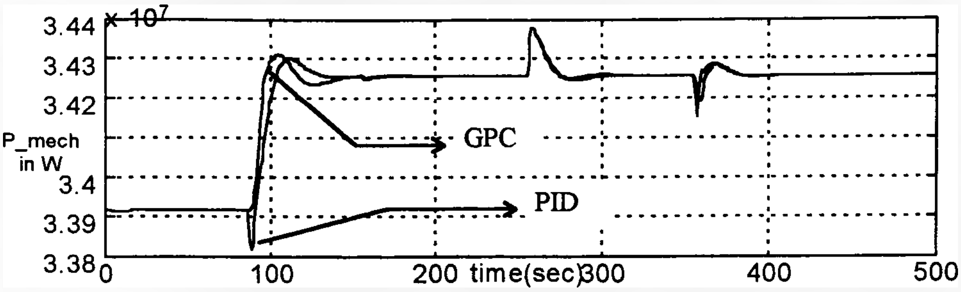


(c)

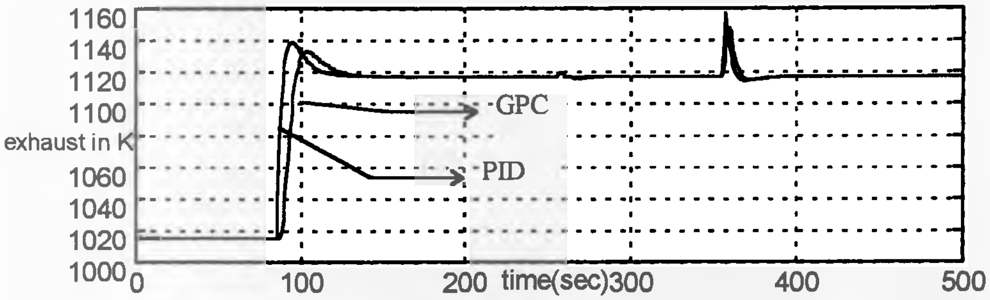
Figure 4.17(a, b, c): Set-point decoupling when Nu=5 and 15.

4.5.3 Disturbance Rejection

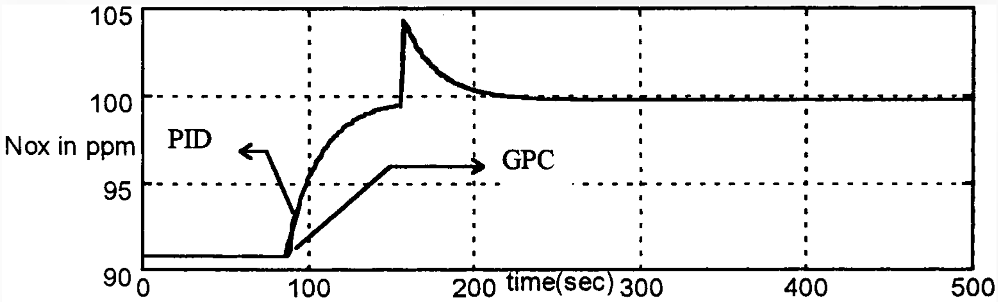
Step disturbances are injected at the turbine inputs, injected steam demand, air demand and fuel demand at 150 sec, 260 sec and 360 sec respectively as can be seen in Figure 4.18 (a, b, c). Total disturbance rejection is achieved with no offsets using the local controllers. GPC controller resulted in an overshoot attenuation as can be seen from Figure 4.18 (b) in the output exhaust temperature and the interaction caused in the output power at 260 sec. The results shown here are for the tuning parameters of Table 4.2. However, simulations using other sets of parameters give no improvement over these results. This may be due to good tuning of PID controller which is a desired feature for supervisory control design.



(a)



(b)



(c)

Figure 4.18(a, b, c): Disturbance rejection.

4.6 Performance and Stability Robustness Study

Maximum and/or minimum singular values of sensitivity functions [40, 41, 43] (sensitivity, complementary sensitivity and control sensitivity) and unstructured additive and multiplicative perturbations are used for the assessment of the control systems performance and robustness of the design. Stability robustness may be quantified in terms of the maximum modelling errors that will not destabilise a stable closed-loop system. The method developed in Chapter two for the multivariable state space GPC will be used to derive the sensitivity functions of the closed-loop GPC supervisory control system. The augmented linear model derived in Section 4.3.2 will be used in the formulation instead of the general state space model of equation (2.14). The linear model of GPC supervisory system used to derive the sensitivity functions is shown in Figure 4.19.

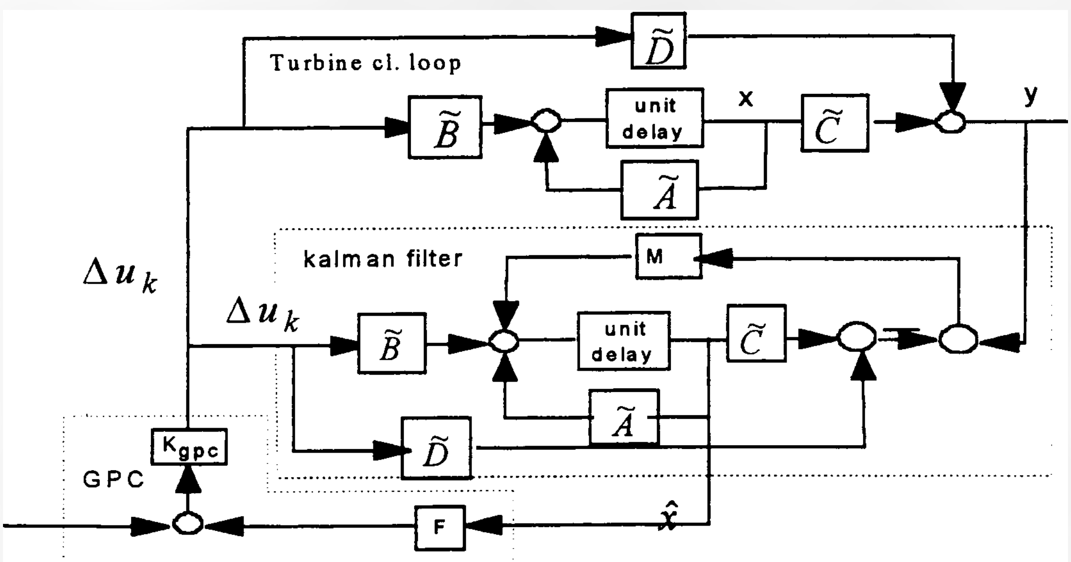


Figure 4.19: GPC Supervisory closed loop system

The same system matrices and transfer function matrices as those derived in chapter two are obtained except for the use of the augmented model. Thus, a reference will be made to Chapter two whenever it is necessary. Detailed analysis to quantify the effects of the controller tuning parameters on the sensitivity functions and on the uncertainties is made to understand how to improve the performance and the stability robustness.

4.6.1 Performance Robustness

4.6.1.1 Sensitivity

Sensitivity is an indication of disturbance transmission from the disturbance input d to the plant outputs y. The maximum singular values of equation (2.28) can be written as:

$$\bar{\sigma}(T_{yd}) = \bar{\sigma}([I + G_c K_{eu} K_{fy}]^{-1}) \tag{4.10}$$

It must be low at low frequencies and zero dB at high frequencies. Disturbance amplification occurs if the sensitivity goes higher than zero dB. This usually occurs at crossover frequencies. The singular values of the sensitivity function of GPC controller is shown in Figure 4.20. The improvements that can be achieved on the sensitivity by assigning different values to the tuning parameters N1, N2 and Nu are investigated and the best combination of these tuning parameters is selected. The effects of each individual tuning parameter on the sensitivity are examined. The first tuning parameter that is examined is the minimum output horizon. Increasing N1 from 3 to 8 causes the amplitude of the peak in the sensitivity function, 3.3 dB, to decrease. This implies a reduction in disturbance amplification and thus performance improvement.

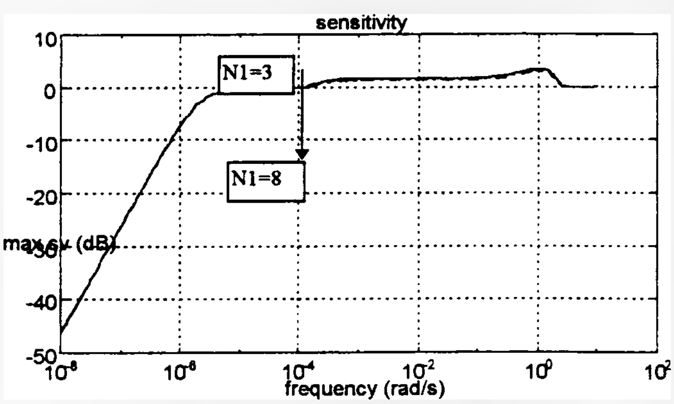


Figure 4.20: Effects of N1 on sensitivity.

The second tuning parameter that affects the sensitivity is the maximum output horizon N2. It is clear from Figure 4.21 that this parameter has a significant influence on the sensitivity shaping. Decreasing N2 from 30 to 10 results in, narrowing the band of frequencies over which disturbance amplification occurs, i.e. reduction of the magnitude of disturbance amplification in this band from 1.88 dB to less than 0.5 dB, except for the peak. A shift of the peak from approximately 2 rad/sec to 0.77 rad/sec occurs with an increase in the peak magnitude from 3.3 dB to 5.1 dB.

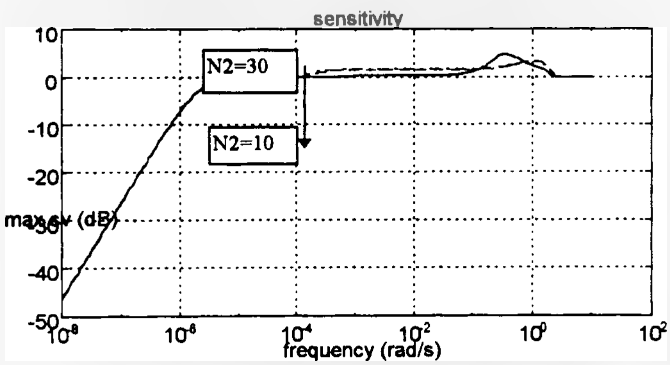


Figure 4.21: Effects of N2 on sensitivity.

An increase in the control horizon Nu from 4 to 19 helps to attenuate low frequency disturbances at the peak from 3.3 dB to 3.1 dB and at the other frequencies from 2 dB to 1.5 dB, and to reduce the band of frequencies over which disturbance amplification occurs. This can be seen in Figure 4.22.

The main conclusion is that N2 has a major influence in shaping the sensitivity function.

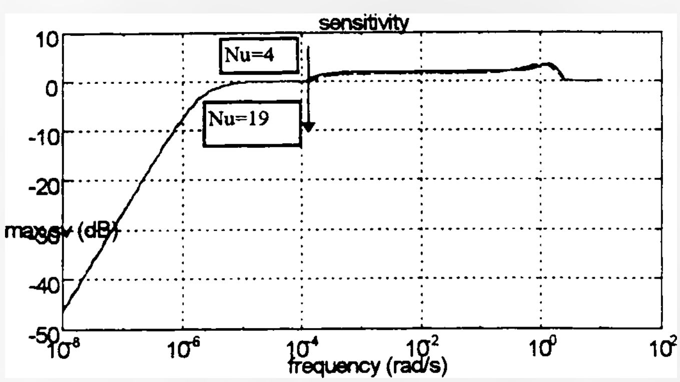


Figure 4.22: The effects of Nu on sensitivity.

4.6.1.2 Complementary Sensitivity

The measure of noise transmission from noise input n to the outputs y in Figure 2.10 is determined using the maximum singular value of the complementary sensitivity. Equation (2.29) can be expressed in terms of the maximum singular values as:

$$\bar{\sigma}(T_{yn}) = \bar{\sigma}([I + G_c K_{eu} K_{fy}]^{-1} G_c K_{eu} K_{fy}) \tag{4.11}$$

The complementary sensitivity function of GPC supervisory closed loop system is shaped using GPC tuning parameters. The effects of the tuning parameters N1, N2 and Nu on the complementary sensitivity function are examined as shown in Figures 4.23, 4.24 and 4.25 respectively. It can be seen from Figure 4.23 that high frequency noise is attenuated at the peak from 2 dB to 1 dB and from approximately 0 dB to -0.7 dB at other frequencies for an increase in N1 from 3 to 8.

N2 causes high frequency noise amplification as it is increased from 10 to 30 at the peak from 1.67 dB to 4.45 dB and a decrease from -0.4 dB to -0.8 dB at other frequencies as can be noticed from Figure 4.24.

Nu is increased from 4 to 19 and it is observed that better high frequency noise attenuation is obtained when Nu=19. This can be seen from Figure 4.25.

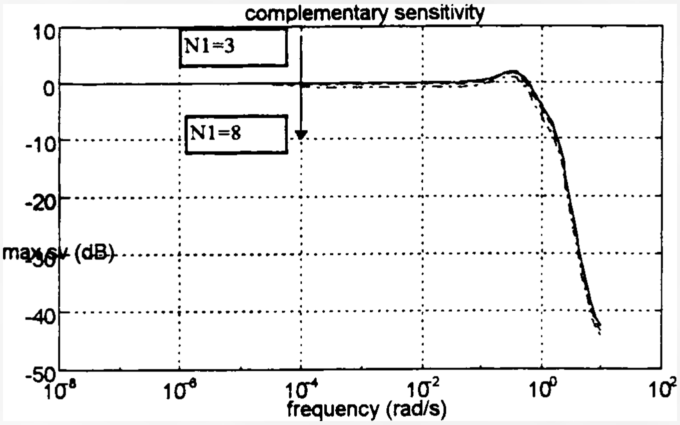


Figure 4.23: Effects of N1 on the complementary sensitivity.

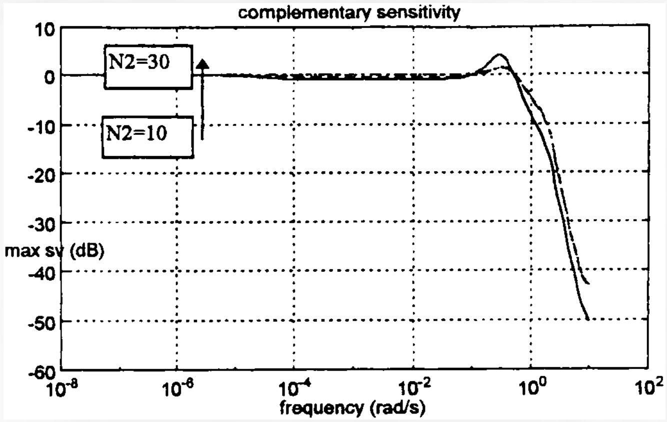


Figure 4.24: Effects of N2 on the complementary sensitivity.

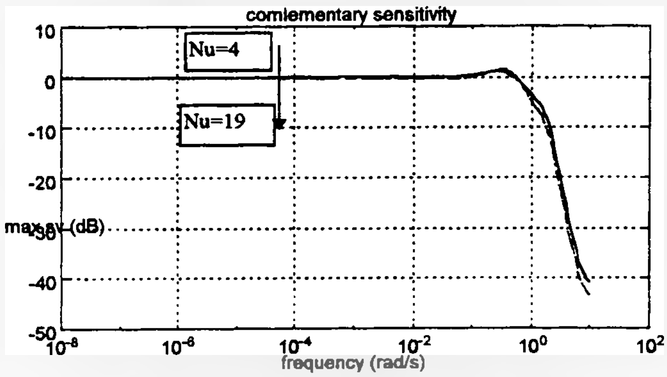


Figure 4.25: Effects of N2 on the complementary sensitivity.

4.6.1.3 Control Sensitivity

It is a measure of the set-point tracking and control activity. Its maximum singular values can be found using equation (2.29) as follows:

$$\overline{\sigma}(T_{\Delta uw}) = \overline{\sigma}([I + K_{eu}K_{fy}G_c]^{-1}K_{eu}) \tag{4.12}$$

The control sensitivity as a function of the tuning parameters N1, N2, and Nu is shown in Figures 4.26, 4.27 and 4.28 respectively. An increase in the minimum output horizon N1 increases the control activity at low frequencies but decreases this activity at high frequencies. The maximum output horizon N2 decreases the control activity at high frequencies as it increases from 10 to 30.

The control costing horizon Nu increases the control activity over the whole range of frequencies as it is increased from 4 to 19.

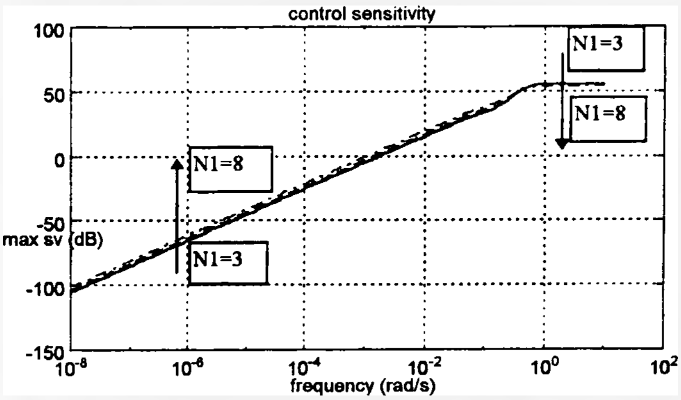


Figure 4.26: Effects of N1 on the control sensitivity.

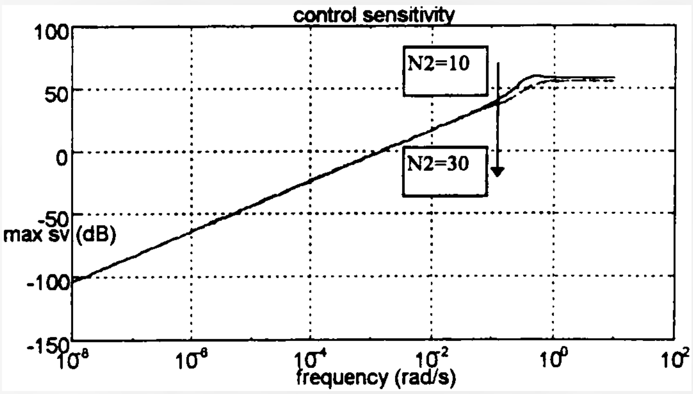


Figure 4.27: Effects of N2 on the control sensitivity.

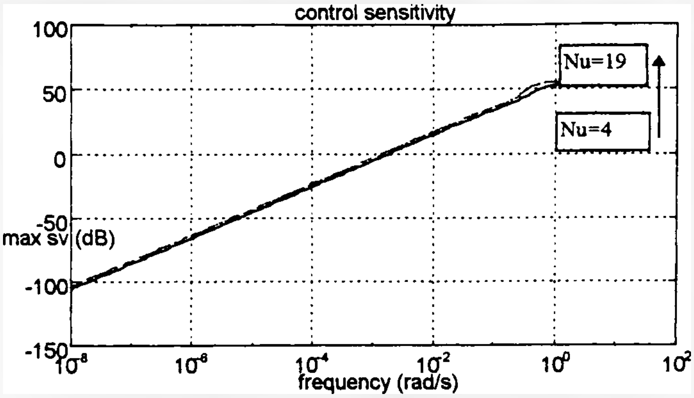


Figure 4.28: Effects of Nu on the control sensitivity.

4.6.2 Stability Robustness

4.6.2.1 Additive Perturbation

A measure of systems closed loop stability is the maximum singular values of the additive perturbation ΔGc that changes the nominal output y_n to $y = y_n + \Delta G_c \Delta u$. The transfer function matrix seen by this perturbation is given by equation 2.36.

Applying small gain theorem which states that the closed loop system is stable for the perturbation ΔGc only if:

$\|\Delta G_c\|\|M_p\| \leq 1$ for all frequencies ω

Thus, the maximum singular value of the perturbation ΔGc can be written as:

$$\bar{\sigma}(\Delta G_c) \leq \frac{1}{\bar{\sigma}([I + K_{ue}K_{fy}G_c]^{-1}K_{ue}K_{fy})} \tag{4.13}$$

This represents a sufficient test for stability robustness with an additive perturbation giving indication of the tolerable additive modelling error for the plant G_c at each frequency range.

The effects of changing the maximum output horizon N2 from 10 to 30 on the permissible additive perturbation limits are demonstrated in Figure 4.29. It can be seen that the additive perturbation limit remains unchanged for frequencies $\leq$ 0.8 rad/s. This limit is decreased as N2 increases.

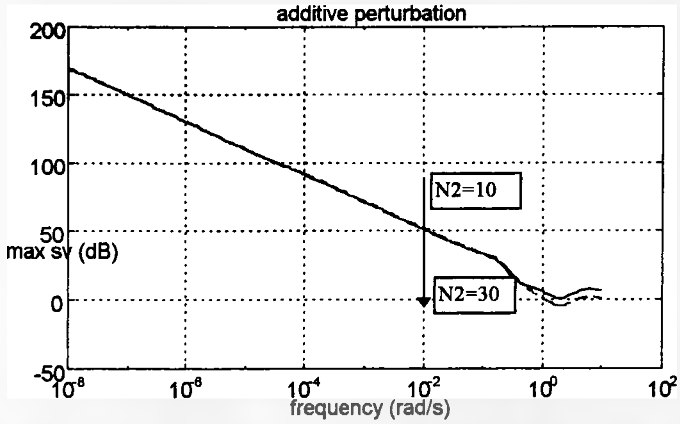


Figure 4.29: Effects of N2 on additive perturbation.

4.6.2.2 Multiplicative Perturbation

The equivalent transfer function matrices seen by the input and output multiplicative perturbations are given by equations (2.38) and (2.39) respectively.From small gain theorem the conditions for stability under input and output multiplicative perturbations are:

$\|\Delta G_{ci}\|\|M_i\| \leq 1$ for all frequencies ω (For input multiplicative perturbations)

$\|\Delta G_{co}\|\|M_o\| \leq 1$ for all frequencies ω (For output multiplicative perturbations)

Using the maximum singular values test for the multiplicative perturbations the following robust stability bounds are obtained;

$$\overline{\sigma}(\Delta G_{ci}) \leq \frac{1}{\overline{\sigma}([I + K_{ue}K_{fy}G_c]^{-1}K_{ue}K_{fy}G_c)} \tag{4.14}$$

$$\overline{\sigma}(\Delta G_{co}) \leq \frac{1}{\overline{\sigma}([I + G_cK_{ue}K_{fy}]^{-1}G_cK_{ue}K_{fy})} \tag{4.15}$$

It can be seen from equation (4.15) and equation (4.11) that minimising the output multiplicative perturbation is corresponding to maximising the complementary sensitivity function.

The permissible input multiplicative perturbation limit as function of N2 is shown in Figure 4.30. At low frequencies, the limit remains constant. However, at high frequencies the limit is affected by changing N2 as can be seen in Figure 4.30.

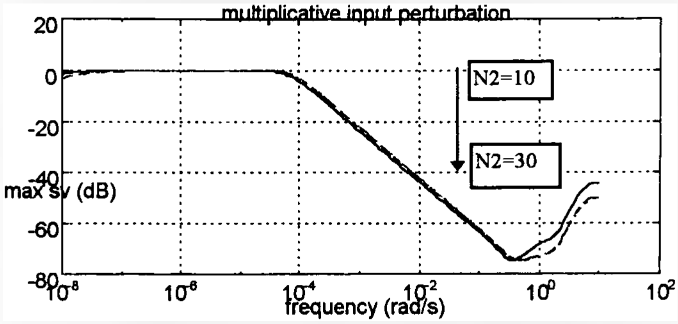


Figure 4.30: Effects of N2 on input multiplicative perturbation.

The output multiplicative perturbation limit as a function of N2 is just the inverse of the complementary sensitivity as shown in Figure 4.31.

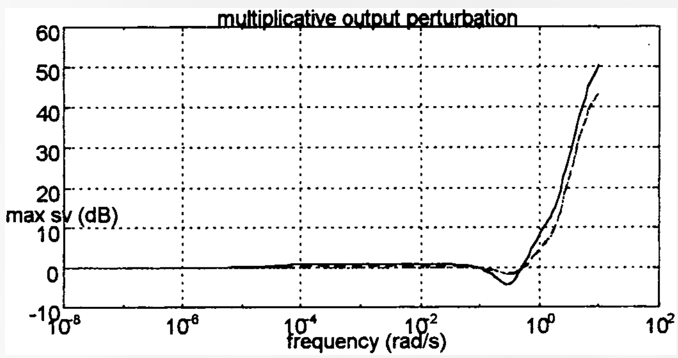


Figure 4.31: Effects of N2 on output multiplicative perturbation.

4.7 The Effects of Filter Parameters

Kalman filter weighting matrices Q and R (diagonal matrices) play an important role in loop shaping. They have direct and significant influence on the sensitivity and the complementary sensitivity (that is, low and high frequency disturbance rejection). This can be observed easily from Figure 4.32 and Figure 4.33.

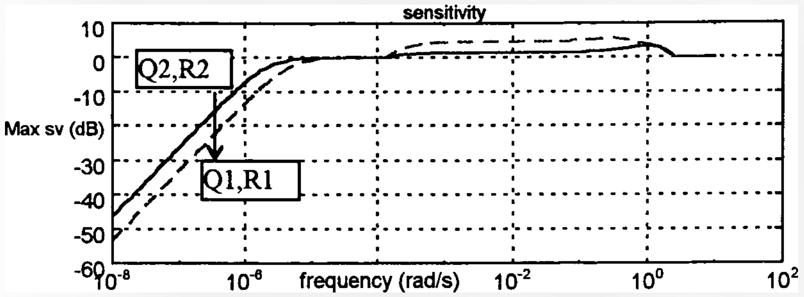


Figure 4.32: Effects of filter weighting parameters on sensitivity

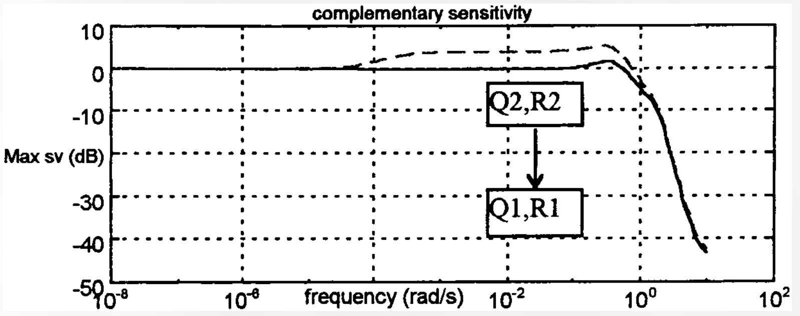


Figure 4.33 Effects of Filter parameters on complementary sensitivity

Unlike the sensitivity and the complementary sensitivity, Kalman filter weighting matrices have negligible effects on the control sensitivity. This can be seen in Figure 4.34. Additive perturbation limits seem also slightly affected by Kalman filter weighting matrices as shown in Figure 4.35.

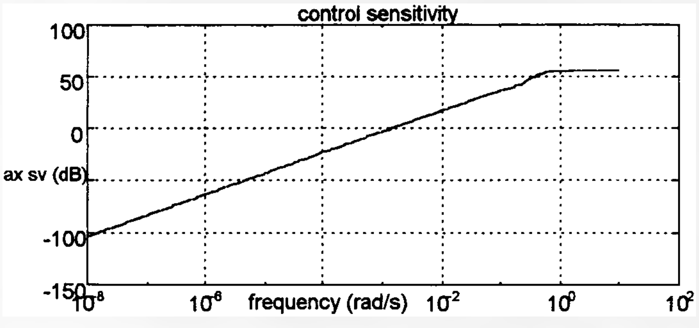


Figure 4.34: Effects of Kalman filter parameter on control sensitivity.

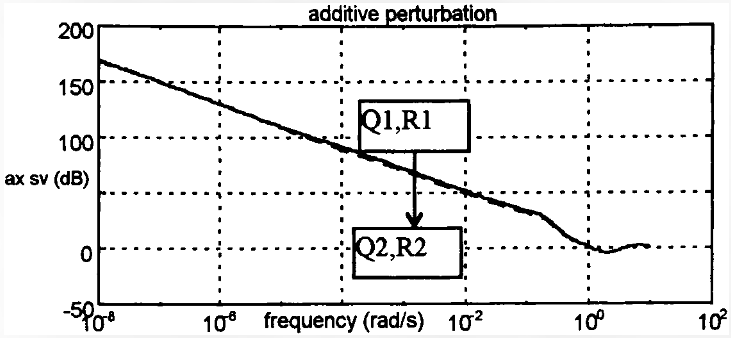


Figure 4.35: Effect of Kalman filter parameters on additive perturbation.

The main conclusions of performance and stability robustness studies can be summarised as follows:

- The previous analysis revealed that the controller is stable over a wide range of tuning parameters and weighting elements. This provides a flexible design to achieve the desired performance.

- Interaction between different channels can be reduced by proper selection of the controller tuning parameters.

- Low frequency disturbance rejection can be improved in many ways by choosing suitable values for the controller tuning parameters.

- High frequency disturbances rejection is also a function of the controller tuning parameters, i.e. the effects of the high frequency disturbances can be suppressed by proper selection of the controller tuning parameters.

- The control activity depends on the controller tuning parameters. The control effort can be affected over the frequency range of interest by varying the tuning parameters.

- The desired performance robustness can be achieved by several combinations of the controller tuning parameters.

- Kalman Filter parameters play important roles in shaping the sensitivity functions mainly the sensitivity and the complementary sensitivity. The filter parameters provide additional degrees of freedom for the control design.

- Stability robustness of the control design is also affected by the controller various tuning parameters. The limits of the uncertainties that determine the stability robustness can be improved by the controller tuning parameters.

4.8 Final Control Design

The two studies presented in the previous Sections are summarised here to make a compromise among the design objectives and to select a set of controller parameters that satisfy these objectives. Table 4.3 gives a summary of the effects of the tuning parameters (N1, N2, Nu) on different performance specifications and Table 4.4 summarises the results of performance and stability robustness study. The control objectives are summarised here.

- Minimum possible overshoots in the outputs and in the optimised set-points.
- Minimum possible rise time and settling time.
- Small time delays.
- Good rejection of low and high frequency disturbances.
- Good tracking properties.
- Improved stability and performance robustness.

The other objectives are to allow, a flexibility in constraints handling and a smooth transition through several operating points (start up and shut down of the gas turbine). These will be discussed in the following Sections and in the next Chapter. It is possible through investigating the results shown in Tables 4.3 and 4.4 to draw conclusions on how some of the objectives can be met. After the investigation of the results in Table 4.3 and Table 4.4 a compromise among all the stated design objectives is made and new control design parameters are obtained as shown in Table 4.5. It can be seen from this table that intermediate values of controller parameters are selected. From Table 4.3 the following notes can be concluded:

	N1 [3 → 8]						N2 [15 → 30]						Nu [4 → 19]					
	Percentage overshoot	t_r	t_s	t_d	t_p	Set-point decoupling	Percentage overshoot	t_r	t_s	t_d	t_p	Set-point decoupling	Percentage overshoot	t_r	t_s	t_d	t_p	Set-point decoupling
Output power	↓	↑	≅	≅	↑	↑	↓	↑	≅	≅	↑	↓	↑	↓	↓	≅	↓	↑
Output temperature	↓	↑	≅	≅	↑	≅	↓	≅	≅	≅	↑	≅	↓	↑	↑	≅	↑	≅
Nox output level	≅	↑	↑	↑	≅	↑	≅	≅	≅	≅	≅	↓	≅	≅	≅	≅	≅	↑

tp: Time of overshoot peak. →: Increased to. ↑: Increasing. ↓: Decreasing. ≅: slightly affected.

Table 4.3: Effects of the controller tuning parameters on the performance of the control design

Performance and stability robustness measures	Frequency (rad/sec)	N1 [3 → 8]	N2 [10 → 30]	Nu [4 → 19]
Sensitivity (Low frq. Disturbance amplification)	High (at the peak)	↑	↓	↓
	Medium	↑	↑	↓
	Low	N	N	N
Complementary sensitivity (High frq. Disturbance amplification)	High (at the peak)	↓	↓	↓
	at Higher frequencies)	↓	↑	↑
	Medium	↓	↑	↓
	Low	N	N	N
Control sensitivity (Control activity and tracking)	High	↓	↓	↑
	Medium	↑	≅	↑
	Low	↑	≅	↑
Additive uncertainty limit (Stability robustness measure)	High	×	↓	×
	Medium	×	↓≅	×
	Low	×	↓≅	×
Input multiplicative uncertainty limit (stability robustness measure)	High	×	↓	×
	Medium	×	↑	×
	Low	×	≅	×
Output multiplicative uncertainty limit (Stability robustness measure)	High (at the peak)	×	↑	×
	Higher frequencies	×	↓	×
	Medium	×	↑	×
	Low	×	N	×

→: Increased. ↑: Increasing ↓: Decreasing ≅: Slightly affected ×: Not used N: No change ↓≅: :Slightly decreased

Table 4.4: Effects of the controller tuning parameters on performance and stability robustness of the control design.

GPC tuning parameters	Weighting on the control increments (λ)	Weighting on the prediction error (γ)
$N1$=5	λ_1=0.00000007	γ_1= 0.0000000007
$N2$=19	λ_2= 19	γ_2=0.25
Nu=11	λ_3=0.3	γ_3= 5

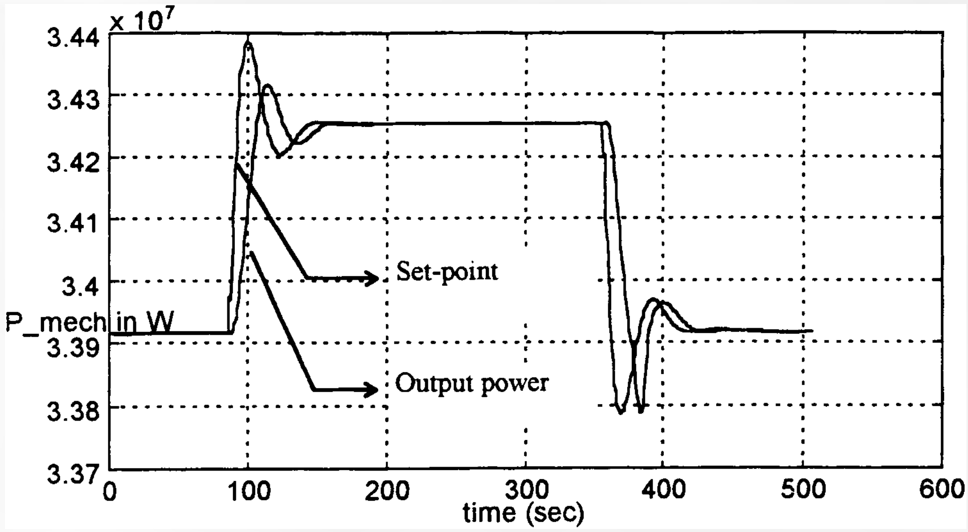
Table 4.5 GPC Final design tuning and weighting parameters.

1. Percentage overshoots in both the output power and the outlet temperature can be decreased by either increasing N1 or N2. This indicates a decreased in the speed of response (rise time increased). An increase in Nu increases the output power speed of response. However for the same increase in Nu there is a decrease in the speed of response in the outlet temperature.

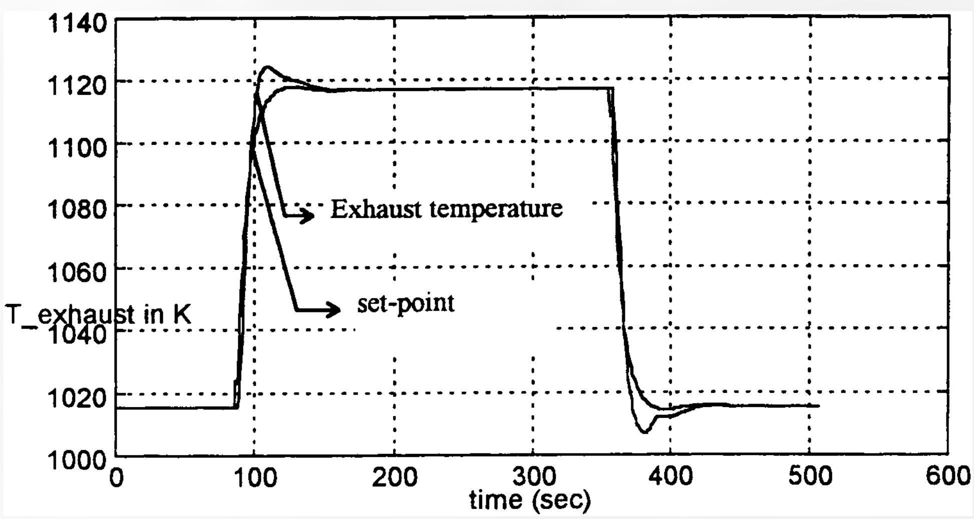
2 Increasing N1 and Nu improves set-point decoupling (decreases interaction) but an increase in N2 increases the interaction which is not desirable.

4.8.1 Simulation Results for Final Design

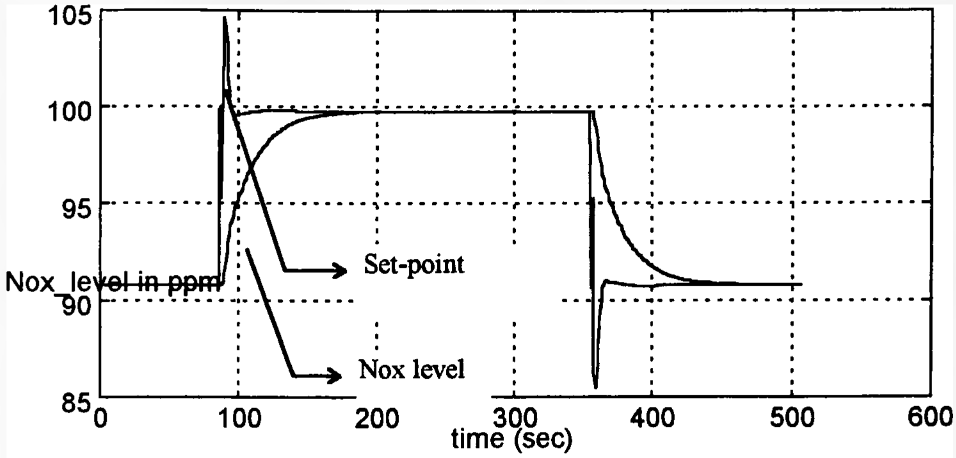
The responses of the gas turbine power plant supervisory system to step inputs are shown in Figure 4.36 (a, b, c). These Figures also show the lower level PID controllers optimised set-points. These plots are used to determine the time domain specifications of the control design. A summary of these specifications is given in Table 4.6. The frequency response of the final control design is shown in Figure 3.37. Frequency domain specifications are determined from this figure and are summarised in Table 4.7.


x 10^7
3.44
3.43
3.42
3.41
3.4
3.39
3.38
3.37
P_mech in W
Set-point
Output power
0
100
200
300
400
500
600
time (sec)

(a)


1140
1120
1100
1080
1060
1040
1020
1000
T_exhaust in K
Exhaust temperature
set-point
0
100
200
300
400
500
600
time (sec)

(b)



(c)

Figure 4.36(a, b, c): Gas turbine power plant outputs versus local controllers set-points.

Inputs and outputs	Control design performance (time in sec)				
	%Overshoot	tr	ts	td	tp
Output power	18.33	12.54	60.83	13.3	28.5
Optimised power set-point	38.67	7.41	51.33	3.8	15
Output exhaust temperature	7.7	9.54	34.35	7.63	40.07
Optimised exh. temp. set-point	1.78	13.35	21	6.49	22.88
Nox output level	×	39.69	60.30	15.26	×
Optimised Nox level set-point	53.57	×	7.63	×	3.81

Table 4.6 Time domain specifications of the control design.

	Frequency range	Robustness
Sensitivity magnitude	(at the peak) 0.5 rad/sec	4.86 dB
	Medium 0.1 to 0.0002 rad/sec	1.26 dB
Complementary sensitivity magnitude	(at the peak) 0.4 rad/sec	3.15 dB
	Medium 0.1 to 0.0002 rad/sec	-0.81 dB
Bandwidth	.7 rad/sec	×

Table 4.7: Frequency domain specifications of the control design.

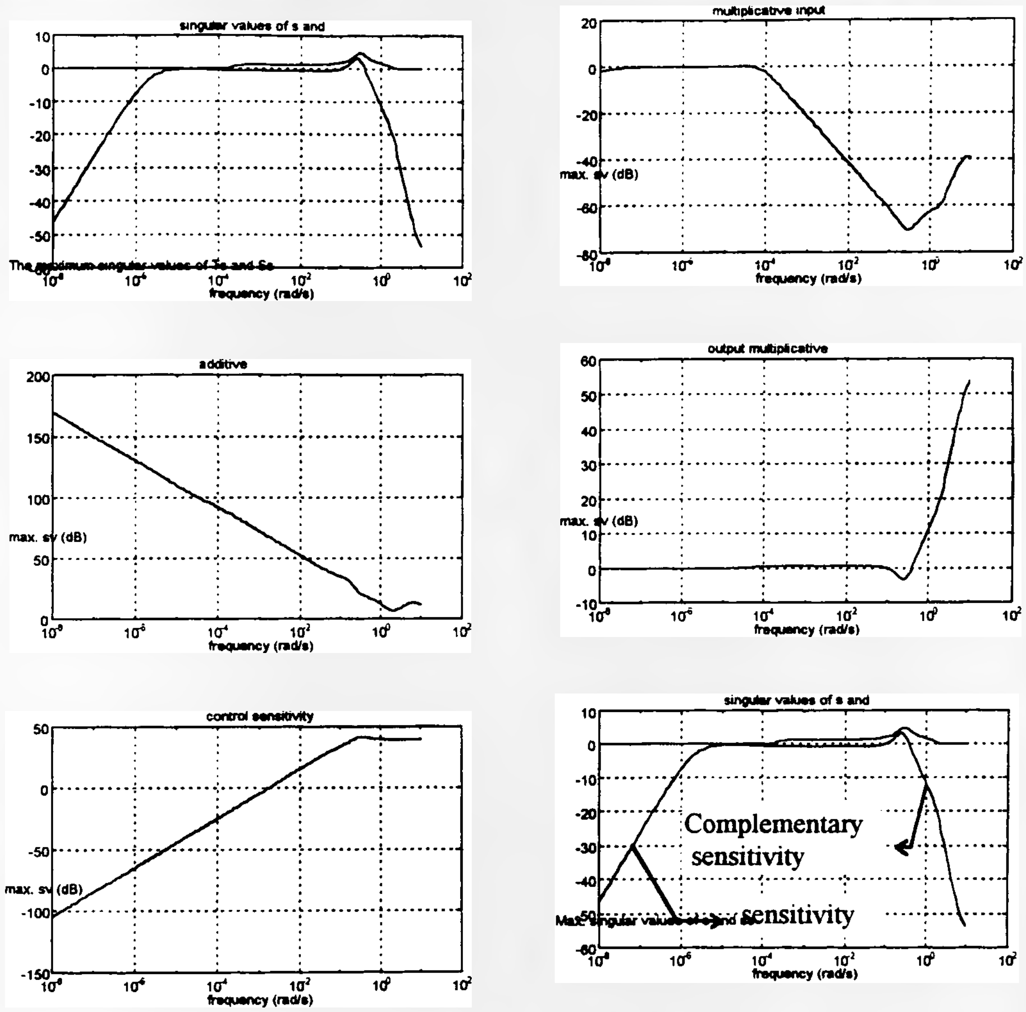


Figure 4.37: Frequency response of the supervisory control system.

4.8.2 GPC versus PID

A set of simulation results comparing the performance of the PID supervisory control system and the GPC supervisory control system is shown in Figure 4.38 (a, b, c). It can be seen from Figure 4.38(a) that GPC supervisory control offers a faster response to the changes in the mechanical power set point with a very slight increase in the % overshoot and the elimination of the nonminimum phase behaviour (the negative going response). The response to the change in the exhaust temperature set point in the case of GPC has a 30 % overshoot reduction over that of PID but with a significant increase in both the rise time and the time delay. Nox level changes to a 10 % increase in the set point remains unchanged except for a slight dead time introduced in case of the GPC supervisory control.

It can be concluded from this comparison that GPC has improved the performance of the control system mainly by the elimination of the nonminimum phase responses and by the significant reduction in the overshoots.

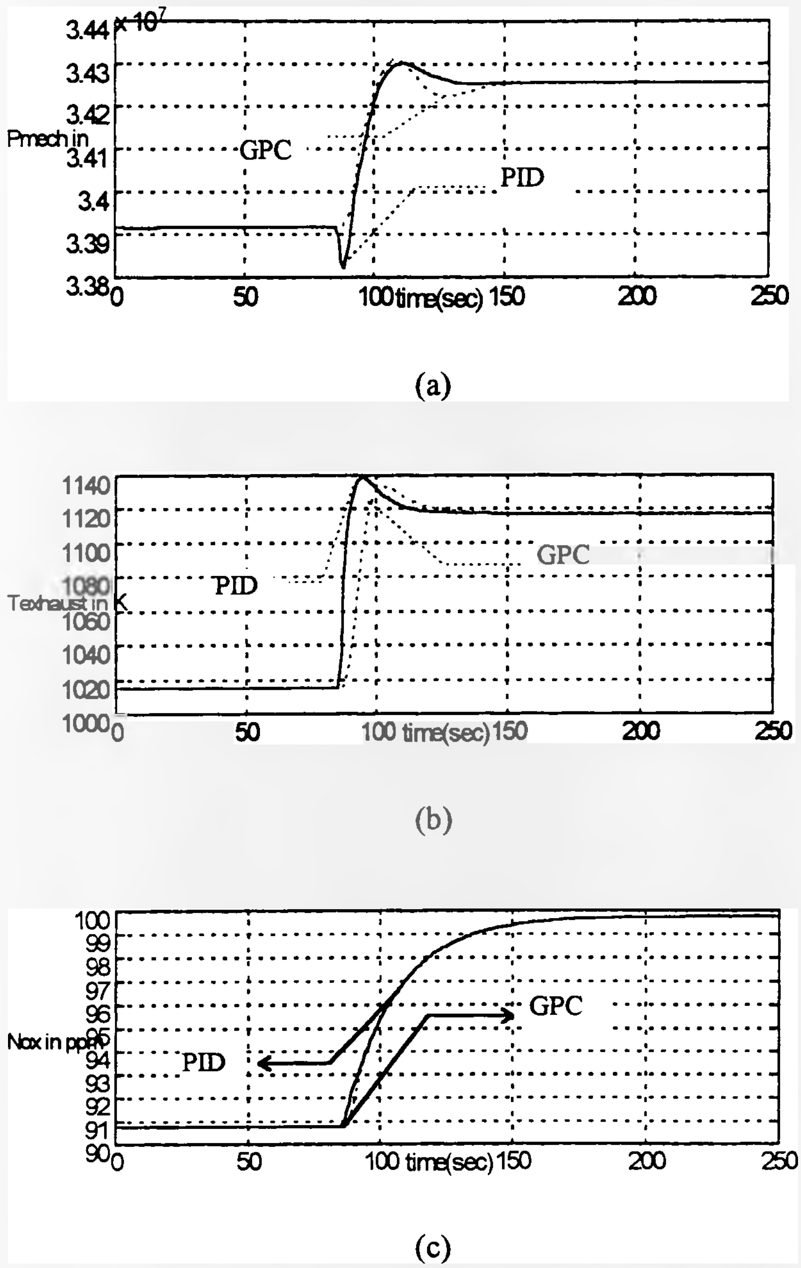


Figure 4. 38(a, b, c): GPC supervisory versus PID supervisory control step responses.

4.9 Performance Robustness and Stability Robustness Analysis

Figure 4.36 and Table 4.6 show the performance results of the control design. It can be seen that the output power has the highest % overshoot of 18.33 on the rising edge of the step response and a %overshoot of 38.67 on the falling edge of the step response. In addition, this output has the longest settling time of 60.83 sec. Nox output level has the longest rise time of 39.69 sec and the longest time delay of 15.36 sec. The performance of the control design is satisfactory. Good tracking of the reference signals is obtained with zero steady state off-set errors. Good performance in disturbance rejection is obtained for all the loops as already discussed in Section 4.6.3. The performance of the closed-loop system is improved as shown in the performance study and in the comparison with PID control. The results of stability robustness are shown in Figure 4.37 and Table 4.7. The peak of 3.15 dB in complementary sensitivity function, which quantifies the tracking properties of the closed loop system, is an indication of overshoots in the process outputs. This can be seen from figure 4.36. The peak of 4.86 dB in the sensitivity function is an indication of disturbance amplification. Control effort is poor at low frequencies. However as the frequency increases the control activity also increases until it saturates at approximately 0.5 rad/esc. The control design has the lowest robustness at high frequencies and high robustness at low frequencies for the additive perturbations. The controller has a high robustness at low frequencies and the lowest robustness at 0.4 rad/sec against the multiplicative input perturbation. Robustness against output multiplicative perturbations is the best at low frequencies and decreases for frequencies more than 0.4 rad/sec.

4.10 Constrained GPC Supervisory Control Design

Constraints can be used for improving performance of closed-loop systems [9, 88, 53, 89]. The presence of constraints in a process result in a control action that should be physically constrained. This may result in an inadequate closed loop performance if the constraints are neglected during controller design. The constrained GPC algorithm developed in Section 2.8 will be used to design the supervisory control system. SIMULINK model of the constrained GPC supervisory control is shown in Figure 4.39. Four types of constraints will be considered namely, constraints on control increments (actuator slew rate), constraints on control magnitude (actuator saturation), constraints on the output (for safety reasons) and overshoot constraints. Performance of the closed loop system is investigated at different input rates and at different reference input arrangements. The performance of the constrained control design is going to be compared with the unconstrained control design. A case where more than one constraint applied at the same time will also be considered. The MATLAB program to implement the constrained GPC supervisory control design is given in appendix B.3.

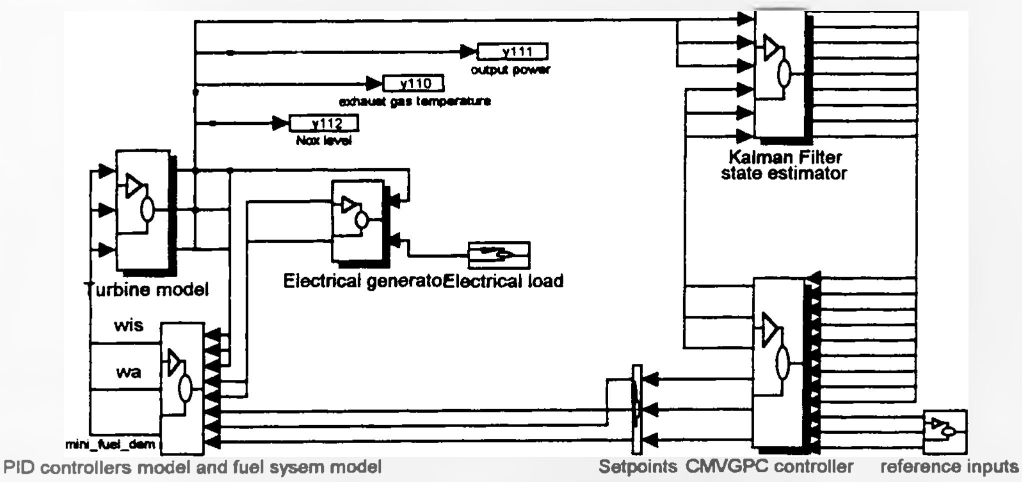
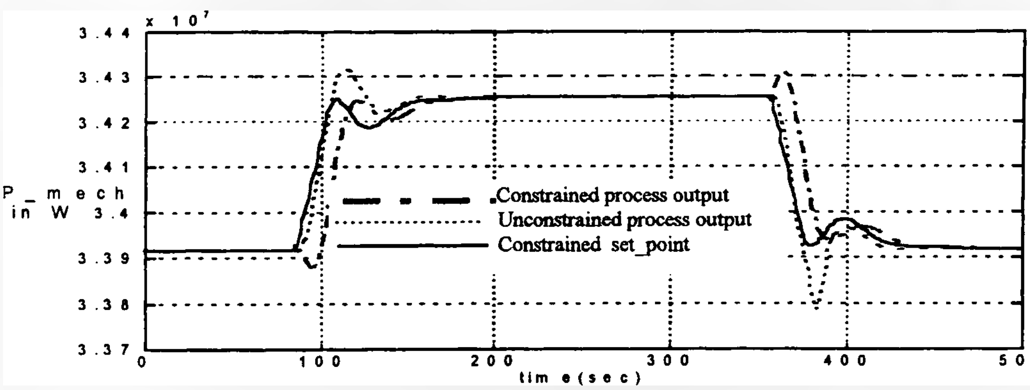


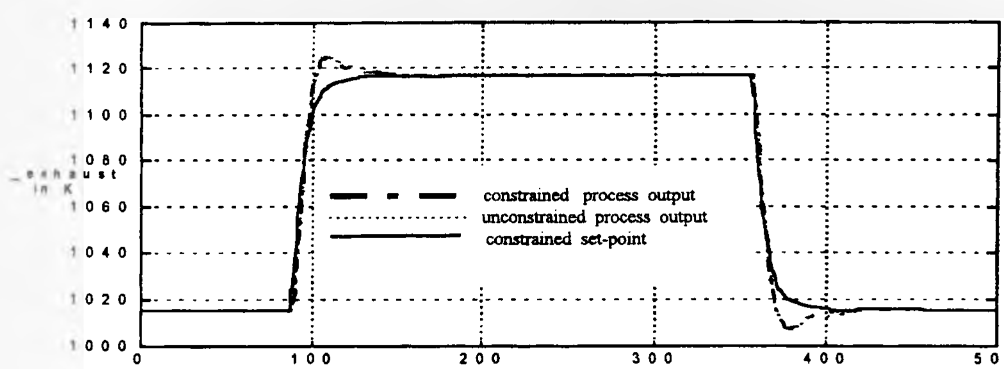
Figure 4.39: Constrained GPC supervisory control system.

4.10.1 Rate Constraints on Power Control Increment

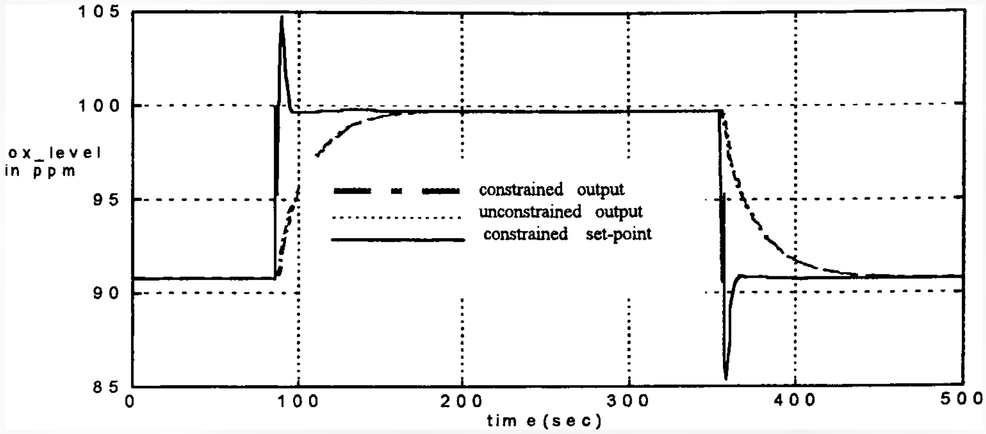
The gas turbine outputs of the unconstrained GPC supervisor as well as the optimised set-points are compared with the outputs of the constrained GPC supervisory system to evaluate the performance of the constrained control system under one input rate constraint only (i.e. rate constraint on power set-point 3.4e5/30 MW). The set-points (GPC outputs) are switched on at the same time and are increased by 10% from their nominal values. The power set-point (continuous line) under the rate constraint, 3.4e5/30, is plotted in Figure 4.40(a) together with the constrained (dashed dotted line) and the unconstrained (dotted line) GPC outputs. The lower $N_{l\Delta u}$ and the upper $N_{u\Delta u}$ rate constraint horizons for future control increments are set to 1 and Nu. It can be seen from this Figure that the rate constraint imposed on the power set-point results in the elimination of overshoot in both the rising and falling edges of the output power step response. The nonminimum phase response and the time delay are however increased. If the nonminimum phase behaviour is undesirable another constraint can be added to reduce or to eliminate this behaviour. From Figure 4.40(b) and 4.40(c) it is obvious that the applied rate constraint on the power set-point has very slight effects on the exhaust temperature and Nox level step responses.



(a)



(b)

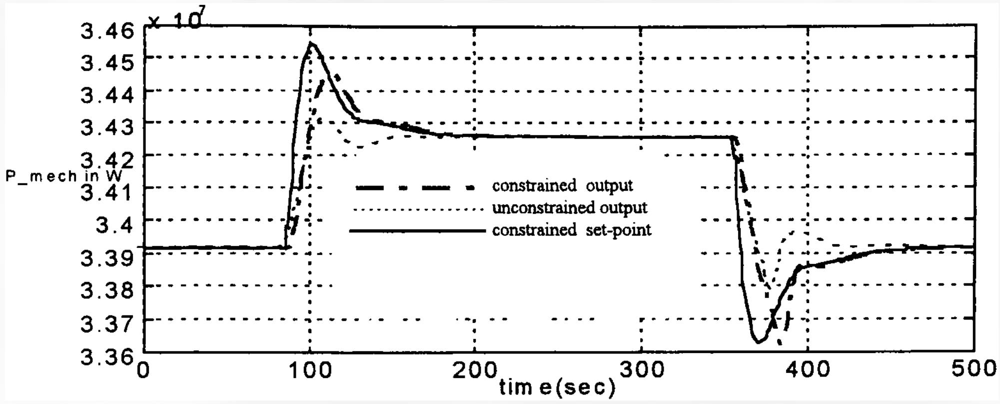


(c)

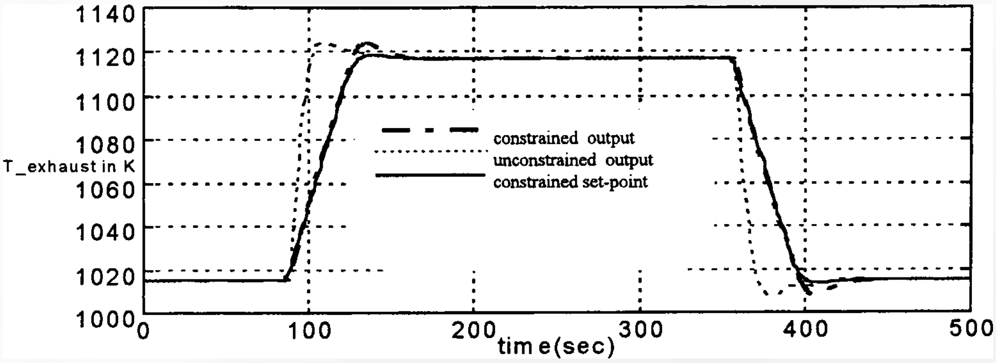
Figure 4.40(a, b, c): Gas turbine step response under rate constrained power input.

4.10.2 Rate Constraints on Temperature Control Increment

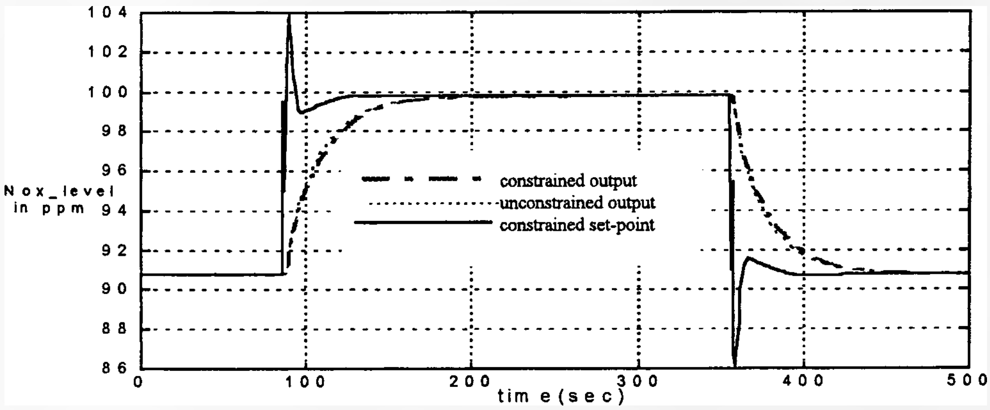
The optimised set-points are allowed to change at the same time and are increased by 10% from their nominal values. The lower $N_{l\Delta u}$ and the upper $N_{u\Delta u}$ rate constraint horizons for future control increments as set to 1 and Nu. A rate constraint of 101.5/30 K is applied to the exhaust gas temperature set-point while keeping the power and Nox level set-point unconstrained. This rate constraint has affected the performance of the closed loop system as can be seen in Figure 4.41 (a, b, c). The overshoots in the power set-point and in the output power, both in the rising and the falling edges of response, are dramatically increased. They are exceeded 70% and 54.4 % respectively. This represents a deterioration in the performance. The exhaust output temperature follows smoothly the set-point and without any time delay but with a longer rise time with respect to the unconstrained case. The overshoot in the Nox level set-point is reduced in response to the rate constraint on exhaust temperature set-point.



(a)



(b)



(c)

Figure 4.41(a, b, c): Gas turbine step responses under rate constrained temperature input

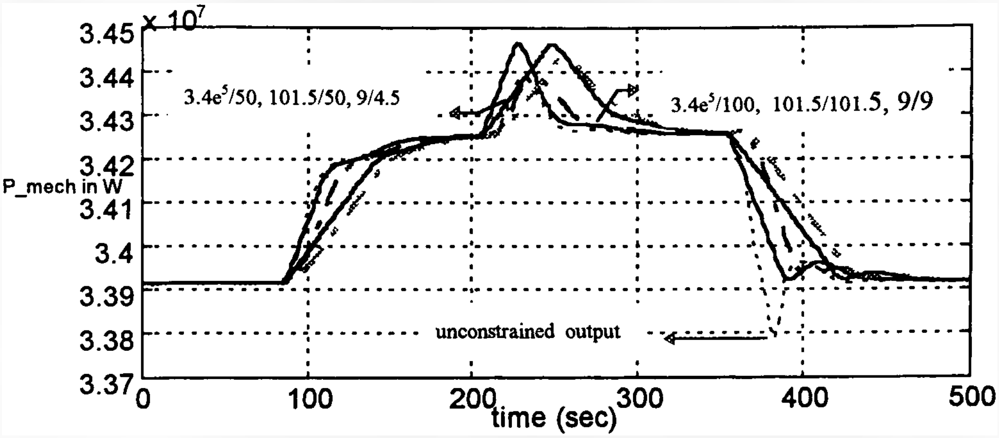
4.10.3 Rate Constraints on Power Temperature and *Nox* Control Increments

The power reference and the output temperature reference signals are switched on at different times and the Nox level is kept at its minimum. These are applied to investigate the effects of the rate constraints on the interaction between the set-points.

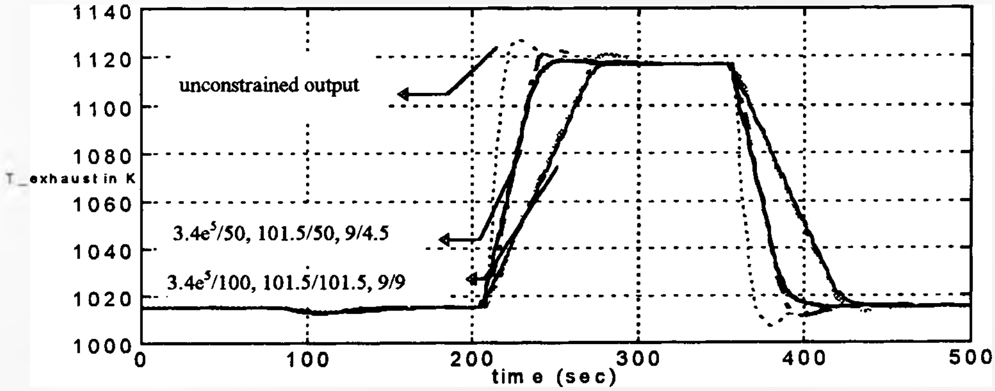
The results of applying two different sets of rate constraints, on power control increment, exhaust temperature control increment, and Nox level control increment are presented to demonstrate the effects of tightening the rate constraints on the performance of the closed-loop system. The responses under the rate constraints ($3.4e^5/50$, 101.5/50, 9/4.5) and ($3.4e^5/100$, 101.5/101.5, 9/9) are compared with the responses of the unconstrained case and are shown in Figure 4.42 (a, b, c). Control increment constrained signals are plotted in Figure 4.43 (a, b, c). Different effects on the performance are observed.

The first rate constraint causes, the elimination of the overshoot that occurs in the falling side of both the power set-point and the output power, no change in the overshoot in the output power due to set-points interaction, and slow response in the output power compared with the output of unconstrained case. However both have reached steady state approximately at the same time.

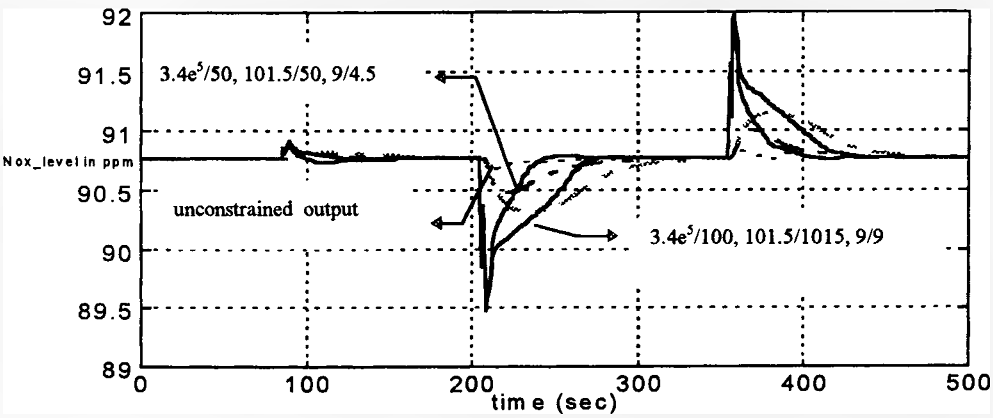
The second constraint results in slower responses compared with the first constraint and an increase in the output power overshoot. These effects can be noticed easily from Figure 4.42 (a). The effects of both sets of constraints on the exhaust temperature set-point and output temperature are an increase in rise time in both sides of the response, (slow responses) there is also a reduction in the overshoots in both sides of the response. It is clear from Figure 4.42 (c) that set-point interactions are increased for tighter constraints. It can be concluded that imposing the rate constraints on the set points improves the performance. However, tightening the constraints may result in undesirable effects such as, slow responses and increased overshoots in some outputs.



(a)

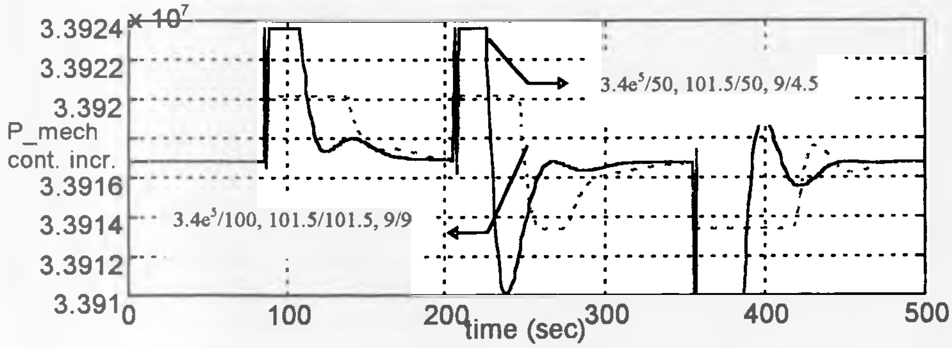


(b)

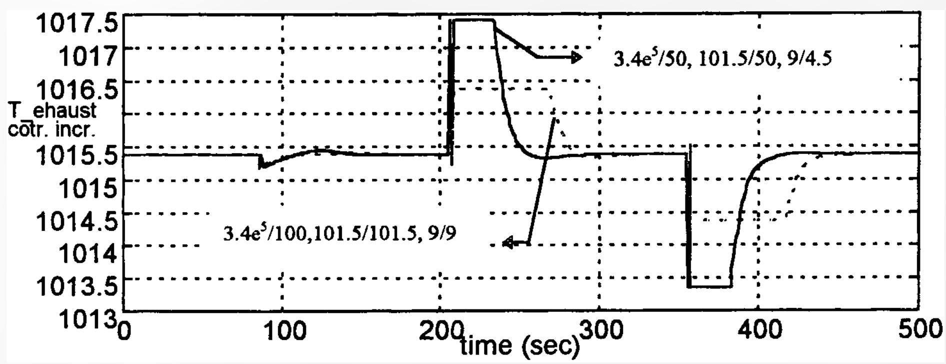


(c)

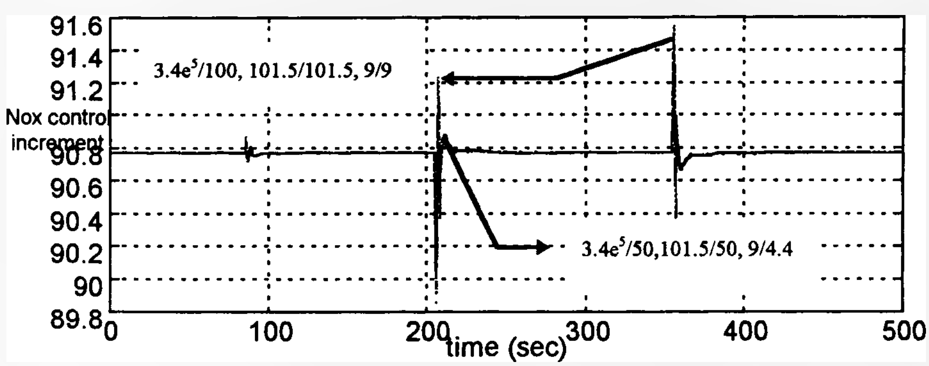
Figure 4.42(a, b, c) Gas turbine outputs under rate constraints on power, temperature and Nox control increments.



(a)



(b)

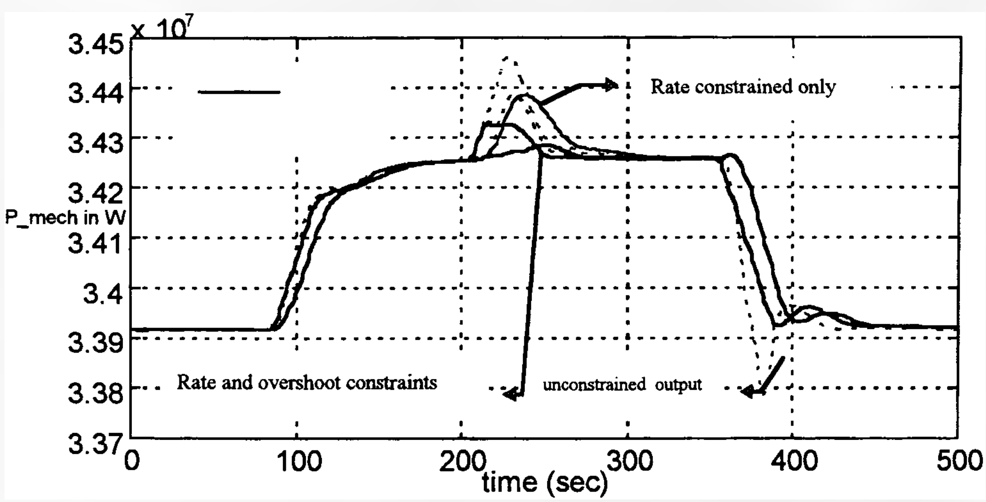


(c)

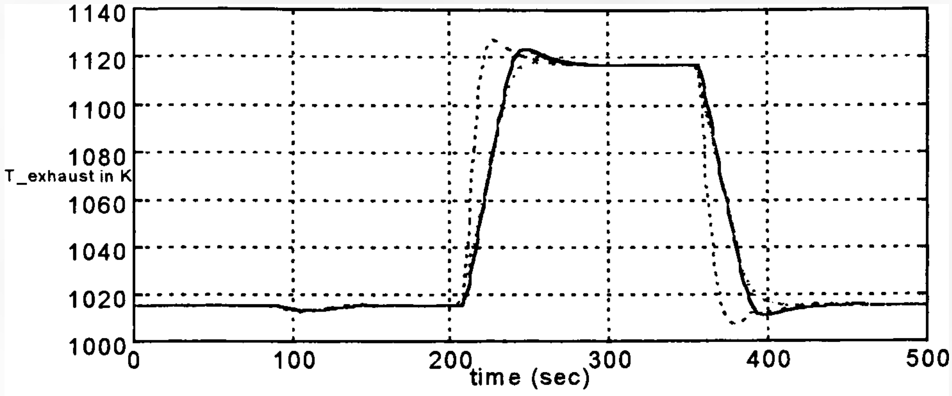
Figure 4.43(a, b, c): Constraints on control increment.

4.10.4 Rate and Overshoot Constraints

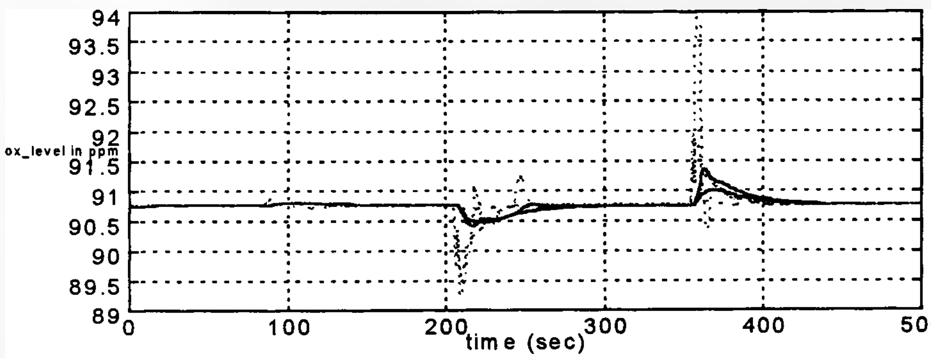
The undesired overshoots in the power set-point and in the output power that resulted from the rate constraints can be reduced or even eliminated by introducing overshoot constraints with lower and upper constraint horizons, N_{lo} and N_{uo}, as new tuning knobs. These new tuning parameters can be used to monitor the amount of the desired overshoots in the system without changing the controller tuning parameters. The idea of overshoot constraint is related to setting an upper or lower limit on the future set-point movements. The step responses of the gas turbine supervisory system under a rate constraint set of ($3.4e^5/50$, 101.5/50, 9/4.5), an overshoot constraint set of (3.4e5, 102, 9) and overshoot tuning parameters set of (5, 19) are depicted in Figure 4.44 (a, b, c). It can be seen from Figure 4.44 (a) that overshoots in the power set-point and the output power are reduced significantly. This set of overshoot constraints have a very slight effect on the output temperature overshoot because the set-point is nearly at its minimum overshoot. The overshoots due to set-points interactions in the Nox output level are affected accordingly by the imposed constraints.



(a)



(b)

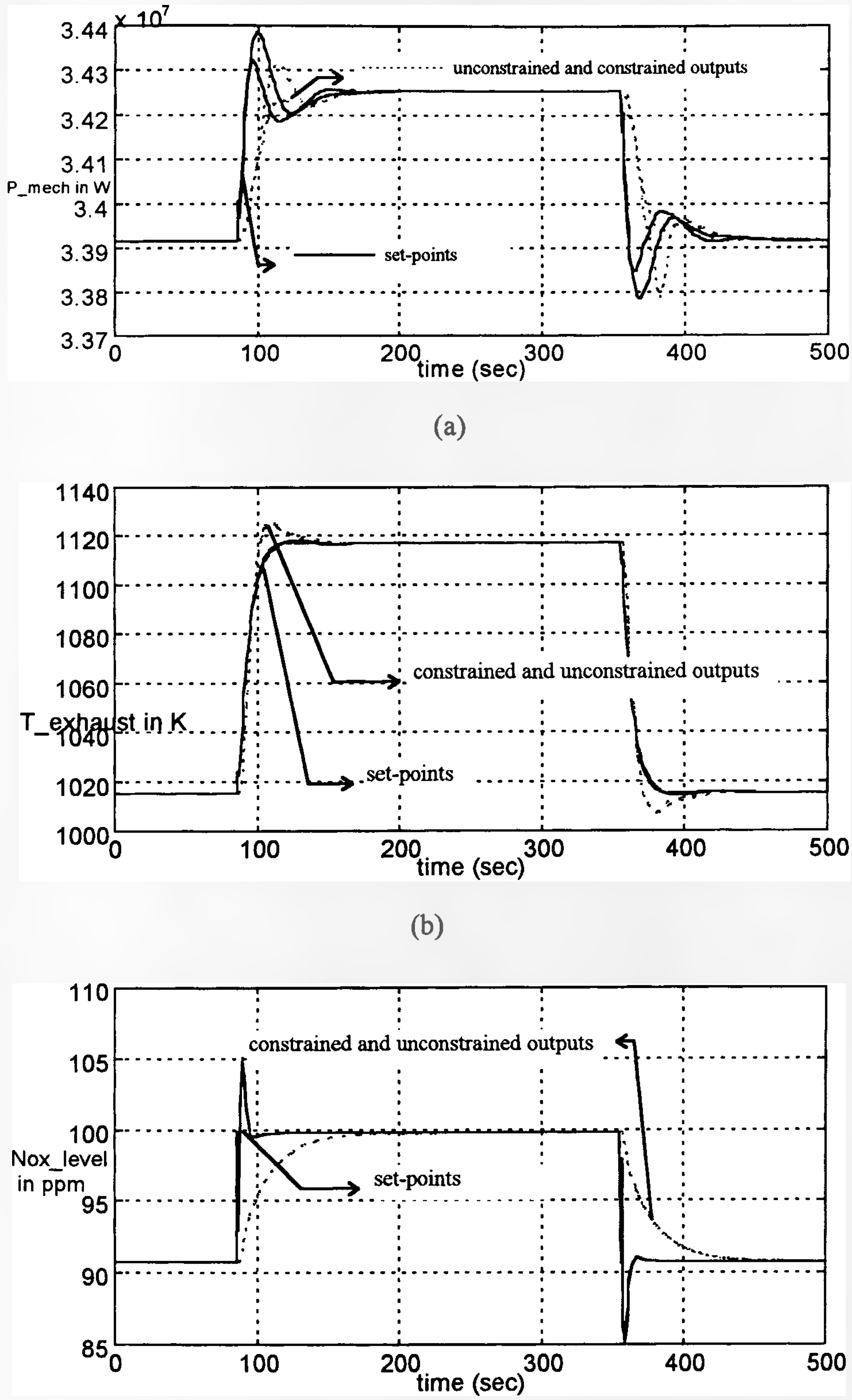


(c)

Figure 4.44(a, b, c): Rate and overshoot constraints.

4.10.5 The Output Constraints

Constraints on the process outputs usually are imposed for safety reasons or when the outputs are allowed to vary within lower and upper limits. Two sets of output constraints namely, maximum output constraint of (3.397e5, 102, 9.3) and minimum output constraint of (-3000, -3, -1) are introduced. The constrained as well as the unconstrained set-points and process outputs are compared in Figure 4.45 (a, b, c). The maximum output constraints applied have values just above the desired outputs to avoid output clipping. It can be seen that output constraints have the same effects as the overshoot constraints on the performance of the control design.

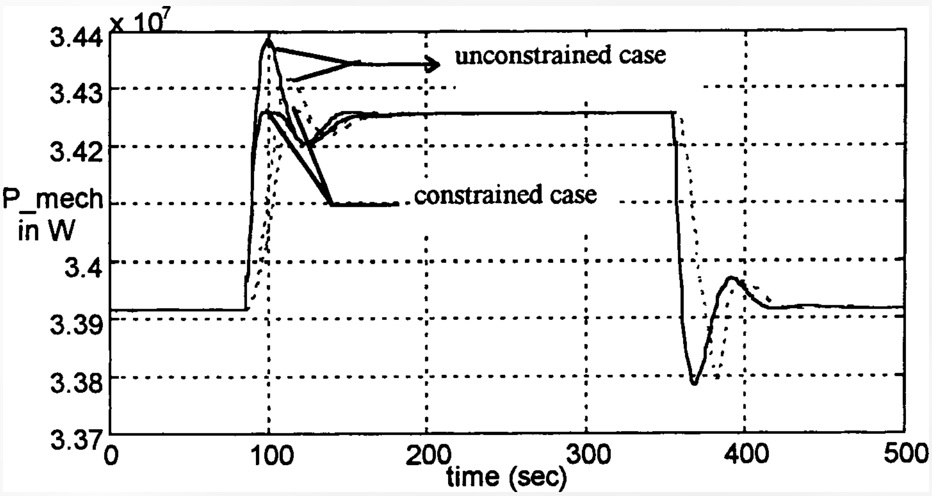


(c)

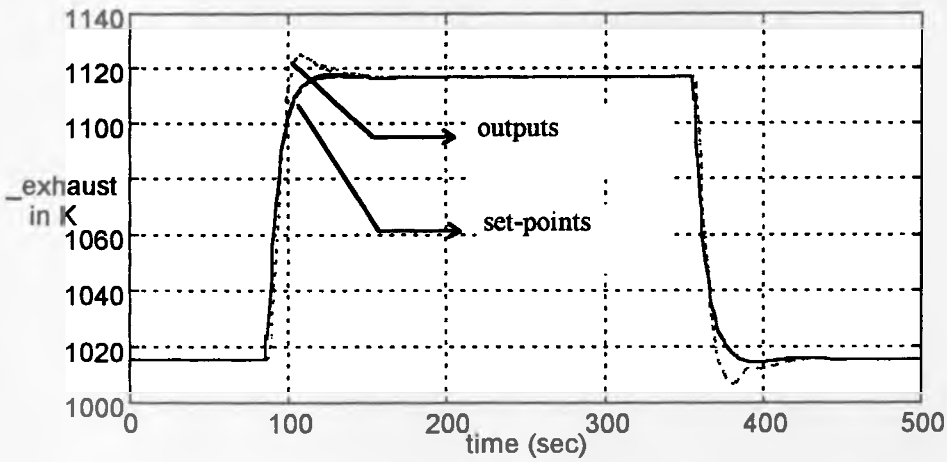
Figure 4.45(a, b, c): Output constraints effects.

4.10.6 Constraints on Control Input Amplitude

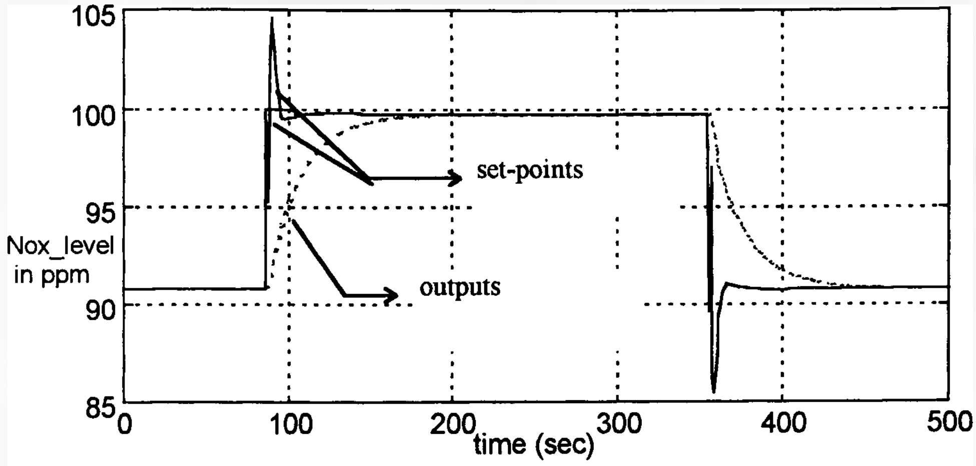
Amplitude constraint on control signals leads to overshoots clipping in the set-points. This in turn leads to overshoot reduction in the output responses of the system. This improves the performance of the control system. In this experiment amplitude constraints on, the power control signal, the exhaust temperature control signal and the Nox level control signal (3.396e5, 102, 9.2) are applied. It is clear from Figure 4.46 (a, b, c) that the optimised set-points overshoots are clipped and the overshoots in the other outputs are reduced.



(a)



(b)



(c)

Figure 4.46: Constraints on control signals

4.11 Conclusions

A predictive supervisory control system was designed for a power plant gas turbine using a multivariable GPC algorithm. The closed-loop analysis developed in Chapter two for this algorithm was applied to quantify the performance and stability robustness of the design. Two studies were carried out for performance and stability robustness assessment. Through these studies it was possible to make a compromise among the design objectives to design a good controller. It was shown how the performance and the stability robustness of the control design can be improved using tuning parameters. The performance of PID and GPC controllers were compared and the results showed that the performance of the closed-loop system can be improved using GPC.

The constrained GPC algorithm of Chapter two was used to design a constrained predictive supervisory system. Several constraints were used to see how the performance of control design can be improved. Simulation results showed that improvement to the performance can be made by the introduction of suitable constraints.

Adaptive Predictive Controller Design

5.1 Introduction

The goal of this chapter is to design a predictive controller in an adaptive manner (gain scheduling) to optimise the start-up and shut-down operations of the power turbine as well as to achieve smooth transition from one operating point to another. As a consequence several linear models at different operating points need to be identified. To achieve this, one of the recently developed subspace state space identification algorithms will be applied off-line to derive a set of linear models from the nonlinear model presented in Chapter three. The algorithm is going to be presented and implemented. Design aspects of identification experiments will be discussed and an m-sequence Pseudo Random Binary Signal (PRBS) will be designed to excite the dynamic modes of the power turbine.

A constrained adaptive predictive controller will then be designed to handle the actuator and process constraints and simulation results will be presented and discussed. Emphasis is placed on state-space model rather than on the input-output transfer functions. This Chapter is organised as follows:

Part one: Power turbine identification.

Part two: Gain scheduling GPC design.

Part one: Power Turbine Identification

A dynamic system is a system where the current output value depends not only on the current external inputs but also on the past inputs and outputs. The mathematical equations relating different variables of a dynamic system establish a model. There are essentially two main approaches to obtain a dynamic model of a system. The first approach applies the laws of physics, that describe the dynamic behaviour of the system, to build the dynamic model of the system as has been presented in Chapter three. The second approach is system identification where the dynamic model of a system is obtained from the experimental data.

An identification problem can be a black-box problem in which no information about the identified system is available or a partial identification problem where knowledge of some characteristics of the system is available. In most practical situations some knowledge about the system to be identified is available. Thus, most practical identification problems are partial [65]. In theory given a set of exact input and output measurements a corresponding exact model can be determined. However, in real life input and output data usually are corrupted by measurement noise and random disturbances. Moreover usually there are inaccuracies in the model equations due to the approximations and the assumptions made about the model. This leads to a statistical problem. The quality of the identification algorithm is then measured by its capability to extract a good model from noisy data.

The motivations of system identification are the need to design better control systems, simulation, prediction, fault detection, operator training, state and parameter estimation and system analysis [65]. System identification is applied to almost all physical, economical, biological and industrial systems. The models obtained by system identification have the following properties [67, 72].

1 They have limited validity.

2 They give little physical insights.

3 They are relatively easy to construct.

Many text books embodying extensive system identification theory are available Ljung [69], Soderstrom and Stoica [67], Graupe [71] and Eykhoff [68] for nonsubspace methods, Overschee and de Moor for subspace methods [72, 83] in time domain and Tomas McKelvey [80] for subspace methods in frequency domain.

5.2 Subspace State Space Identification

Recently developed Subspace State Space methods [73, 77, 79, 85] for dynamic system identification have become well-known and attractive. The approach makes use of the modern robust algorithms of numerical linear algebra such as RQ-factorization, LQ-decomposition, Singular Value Decomposition (SVD) and angles between subspaces. Projections (orthogonal or oblique) provide a geometric framework which allows to determine the states of the model. Unlike subspace based identification methods, prediction error and instrumental variable methods which are classified as classical methods in [73, 83, 85] are based on least squares theory.

Subspace state space identification methods on the other hand make use of modern Algebra and its numerical algorithms to extract directly the linear model from input-output data. Time domain subspace identification algorithms as well as frequency domain counterparts are readily developed. Several works have been published in the field of subspace system identification. Algorithms to identify purely deterministic processes and purely stochastic processes are given in [77] and [74] respectively. The combined deterministic and stochastic systems identification problem was solved in several works for instance, Overschee and De Moor [73] have derived two Numerical Subspace State Space Identification (N4SID), exact and approximate, algorithms that determine the deterministic and the stochastic system at the same time. In these algorithms first a state sequence is found from the projection of the input-output data. Then the state space matrices are determined from the state sequences by solving a least squares problem.

5.3 Subspace versus Classical Model Identification

Overview of subspace-based system identification methods in comparison with the classical methods can be found in different papers [73, 76, 81];

- The main difference between the subspace identification approach and the classical identification approach is depicted in Figure 5.1 [73]. It can be seen that the subspace methods basically look for state sequences (e.g., Kalman states) which are obtained explicitly and directly from input and output data measurements. Then the state space matrices are determined from the state sequences solving a least squares problem. Classical identification methods on the other hand calculate first the system matrices. After that the states can be estimated using Kalman filter.

- The second difference is that there is no need for any parametrizations in subspace methods which eliminates the problem of ill conditioning.

- The third main difference is that subspace methods make use of numerically robust techniques such as RQ decomposition, angles between subspaces and Singular Value Decomposition (SVD). This offers numerically reliable state-space models for complex multivariable dynamic systems directly from the measured data.

- Subspace identification algorithms are non-iterative. Thus no nonlinear search is performed which means the algorithms are convergent.

- Subspace identification algorithms are faster than classical identification algorithms such as Prediction error Methods (PEM).

- They have modest computational complexity in comparison with PEM methods mainly in the identification of large complex systems.

In this thesis subspace approach is adopted for their stated advantages. Moreover Kalman gain can be directly extracted from the input/output data for use in GPC algorithm.

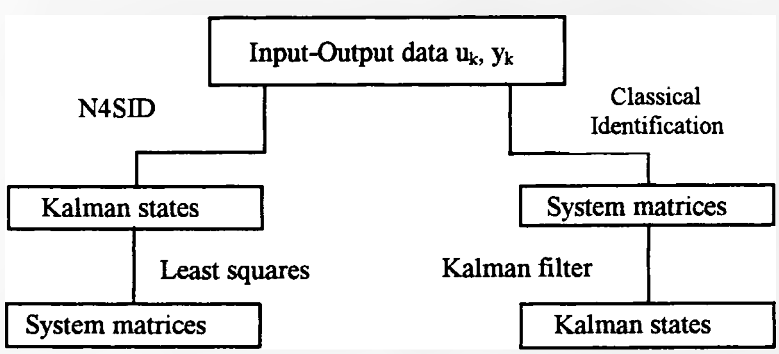


Figure 5.1 N4SID approach versus classical identification approach.

5.4 Subspace Identification Algorithm

The algorithm that is discussed here is the N4SID [73].The steps in the formulation of the algorithm are briefly discussed here:

1 A combined deterministic-stochastic linear time invariant state space system is split into two subsystems, deterministic and stochastic subsystems. The associated matrices of each subsystem are determined e.g., (Observability, controllability and lower block triangular Toeplitz matrix of markov parameters) for deterministic subsystem and (Block Toeplitz covariance and cross covariance matrices) for the stochastic subsystem.

2 Define block Hankel input and output matrices and construct matrix input-output equations from information obtained in step one.

3 Perform the main projections, orthogonal or oblique projections of the future outputs onto the past outputs and the past and future inputs using results of step two. The projections are function of the system matrices and the input-output block Hankel matrices. These projections result in formulas that can be solved in a least squares manner to get the combined system matrices.

The full derivation of the algorithm is very long, thus only the main ideas will be stated.

5.4.1 System Description

The system to be identified is a combined deterministic-stochastic model of the following form:

$$x_{k+1} = Ax_k + Bu_k + w_{wk} \tag{5.1}$$

$$y_k = Cx_k + Du_k + v_k \tag{5.2}$$

The covariance and the cross covariance matrices of the random process and output disturbances are given by:

$$E[\begin{pmatrix} w_{wk} \\ v_k \end{pmatrix}\begin{pmatrix} w_{wl}^T & v_l^T \end{pmatrix}] = \begin{pmatrix} Q_w{}^s & S^s \\ (S^s)^T & R_v{}^s \end{pmatrix}\delta_{kl} \quad \geq 0^2$$

where, δ_{kl} is the Kronecker index, the subscript s stands for stochastic, *E (.)* is the expectation operator, A is $(n \times n)$ system matrix, B is $(n \times p)$ input matrix, C is $(r \times n)$ output matrix, D is $(r \times p)$ direct transmission matrix. $Q_w{}^s$, S^s and $R_v{}^s$ are of dimensions $(n \times n)$, $(n \times p)$ and $(r \times r)$, respectively. v_k and w_{wk} are unmeasurable zero mean Gaussian white noise vector sequences. It is assumed that the pair $\{A,\ C\}$ to be observable (full rank), the pair $\left\{A, \left(B, \left(Q_w{}^s\right)^{1/2}\right)\right\}$ to be controllable this implies that all the modes of the combined system are excited by either the external input u_k or the process noise w_{wk}. The system described by equation (5.1) and (5.2) can be split into deterministic and stochastic subsystems by separating the state x_k and the output y_k into deterministic and stochastic components.

5.4.1.1 Deterministic Subsystem

The deterministic subsystem is given by:

$$x_{k+1}^d = Ax_k^d + Bu_k \tag{5.3}$$

$$y_k^d = Cx_k^d + Du_k \tag{5.4}$$

It describes the influence of the deterministic inputs u_k on the deterministic outputs y_k^d. where, the subscript d stands for deterministic, x_k^d is the deterministic state and y_k^d is the deterministic output. u_k and x_k^d are assumed to be quasi-stationary. The controllable modes of $\{A, B\}$ can be either stable or unstable.

5.4.1.2 Stochastic Subsystems

The stochastic subsystem is given by:

$$x_{k+1}^{s} = Ax_{k}^{s} + w_{wk} \tag{5.5}$$

$$y_{k}^{s} = Cx_{k}^{s} + v_{k} \tag{5.6}$$

It describes the influence of the stochastic noise sequences w_{wk} and v_k on the stochastic output y_k^s. y_k^s is assumed to be stationary. With x_k^s is being the stochastic state sequence and the controllable modes of the pair $\left\{A,(Q_w{}^s)^{1/2}\right\}$ are assumed to be stable.

5.4.2 Block Hankel Matrices

Input data block Hankel matrix is defined as:

$$U_{0/2i-1} = \begin{array}{c} \\ \\ \\ i\updownarrow \\ i\updownarrow \\ \\ \\ \\ \end{array} \left(\begin{array}{ccccc} u_o & u_1 & u_2 & \cdots & u_{j-1} \\ u_1 & u_2 & u_3 & \cdots & u_j \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ u_{i-1} & u_i & u_{i+1} & \cdots & u_{i+j-2} \\ \hline u_i & u_{i+1} & u_{i+2} & \cdots & u_{i+j-1} \\ u_{i+1} & u_{i+2} & u_{i+3} & \cdots & u_{i+j} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ u_{2i-1} & u_{2i} & u_{2i+1} & \cdots & u_{2i+j-2} \end{array}\right)$$

$$= \left(\frac{U_{0/i-1}}{U_{i/2i-1}}\right) = \left(\frac{U_P}{U_f}\right) \in R^{2pi\times j} \tag{5.7}$$

$$U_{0/2i-1} \overset{def}{=} \begin{array}{c} \\ \\ \\ i+1\updownarrow \\ i-1\updownarrow \\ \\ \\ \\ \end{array} \left(\begin{array}{ccccc} u_o & u_1 & u_2 & \cdots & u_{j-1} \\ u_1 & u_2 & u_3 & \cdots & u_j \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ u_i & u_{i+1} & u_{i+2} & \cdots & u_{i+j-1} \\ \hline u_{i+1} & u_{i+2} & u_{i+3} & \cdots & u_{i+j} \\ u_{i+2} & u_{i+3} & u_{i+4} & \cdots & u_{i+j+1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ u_{2i-1} & u_{2i} & u_{2i+1} & \cdots & u_{2i+j-2} \end{array}\right)$$

$$=\left(\frac{U_{0/i}}{U_{i+1/2i-1}}\right) \tag{5.8}$$

The same can be defined for the deterministic and stochastic outputs:

$$Y^d_{o/2i-1} \overset{def}{=} \left(\frac{Y^d_{o/i-1}}{Y^d_{i/2i-1}}\right) = \left(\frac{Y^d_P}{Y^d_f}\right) = \left(\frac{Y^d_{o/i}}{Y^d_{i+1/2i-1}}\right) \tag{5.9}$$

$$Y^s_{o/2i-1} \overset{def}{=} \left(\frac{Y^s_{o/i-1}}{Y^s_{i/2i-1}}\right) = \left(\frac{Y^s_P}{Y^s_f}\right) = \left(\frac{Y^s_{o/i}}{Y^s_{i+1/2i-1}}\right) \tag{5.10}$$

The deterministic and the stochastic outputs can be written in a vector form as:

$$Y = Y^d + Y^s \tag{5.11}$$

where i indicates the number of block row of the block Hankel matrix and is a user defined index (It should be larger than the maximum of the order to be identified), the subscript 0 refers to the first element of the first block row in a block Hankel matrix, j is the number of block columns in the block Hankel matrix (It is assumed that $j \to \infty$) and the subscripts P and f denote the past and the future respectively.

5.4.3 Matrices Associated with the Deterministic Subsystem

Observability and Controllability matrices are extensively used in Subspace identification. The extended (means $i > n$) observability matrix Γ_i with a rank equals to n is given by:

$$\Gamma_i \overset{def}{=} \begin{pmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{i-1} \end{pmatrix} \in R^{ri\times n}$$

The reversed extended controllability matrix Δ_i^d is defined as:

$$\Delta_i^d \stackrel{def}{=} \begin{pmatrix} A^{i-1}B & A^{i-2}B & \cdots & AB & B \end{pmatrix} \in R^{n\times pi}$$

The lower block triangular Toeplitz matrix H_i^d is defined as:

$$H_i^d = \begin{pmatrix} D & 0 & 0 & \cdots & 0 \\ CB & D & 0 & \cdots & 0 \\ CAB & CB & D & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ CA^{i-2}B & CA^{i-3}B & CA^{i-4} & \cdots & D \end{pmatrix}$$

The deterministic state sequence is defined as:

$$X_i^d \stackrel{def}{=} \begin{pmatrix} x_i^d & x_{i+1}^d & x_{i+2}^d & \cdots & x_{i+j-1}^d \end{pmatrix} \in R^{n\times j}$$

$X_f^d = X_i^d$, $X_P^d = X_0^d$ and $X_f^s = X_P^s = X_i^s$

For the deterministic subsystem the following matrices can be found:

$$\lim_{j\to\infty} \frac{1}{j} \begin{pmatrix} U_P \\ U_f \\ \hline X_P^d \end{pmatrix} \left((U_P)^T \quad (U_f)^T \middle| (X_P^d)^T \right) = \left(\begin{array}{cc|c} R_{11} & R_{12} & S_1^T \\ R_{21} & R_{22} & S_2^T \\ \hline S_1 & S_2 & P^d \end{array} \right) = \begin{pmatrix} R & S^T \\ S & P^d \end{pmatrix}$$

5.4.4 Matrices Associated with the Stochastic Subsystem

The output covariance matrix for the stochastic subsystem is defined as:

$$\Lambda_i = E\left[y_{k+i}^s (y_k^s)^T\right] = \begin{cases} CA^{i-1}G & i > 0 \\ \Lambda_0 & i = 0 \\ G^T (A^T)^{-i-1} C^T & i < 0 \end{cases}$$

where:

$$G = E\left[x_k^s (y_k^s)^T\right] = AP^s C^T + S^s$$

$$\Lambda_0 = E\left[y_k^s (y_k^s)^T\right] = CP^s C^T + R_v{}^s$$

$$P^s = E\left[x_k^s (x_k^s)^T\right] = AP^s A^T + Q_w{}^s$$

These equations are called positive real equations (Lyapunov equations). They describe the set of all possible stochastic realisations that have the same second order statistics. The reversed extended (covariance) stochastic controllability matrix is define as:

$$\Delta_i^s \stackrel{def}{=} (A^{i-1}G \quad A^{i-2}G \quad \cdots \quad AG \quad G) \in R^{n\times ri}$$

The block Toeplitz covariance L_i^s and cross covariance H_i^s matrices for the stochastic subsystems are found as follows.

$$\lim_{j\to\infty}\frac{1}{j}[\begin{pmatrix} Y_P^s \\ Y_f^s \end{pmatrix}\begin{pmatrix}(Y_P^s)^T & (Y_f^s)^T\end{pmatrix}]=\begin{pmatrix} L_i^s & (H_i^s)^T \\ H_i^s & L_i^s \end{pmatrix}$$

$$L_i^s = \begin{pmatrix} \Lambda_0 & \Lambda_{-1} & \Lambda_{-2} & \cdots & \Lambda_{1-i} \\ \Lambda_1 & \Lambda_0 & \Lambda_{-1} & \cdots & \Lambda_{2-i} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \Lambda_{i-1} & \Lambda_{i-2} & \Lambda_{i-3} & \cdots & \Lambda_0 \end{pmatrix}$$

$$H_i^s = \Gamma_i \Delta_i^s = \begin{pmatrix} \Lambda_i & \Lambda_{i-1} & \Lambda_{i-2} & \cdots & \Lambda_1 \\ \Lambda_{i+1} & \Lambda_i & \Lambda_{i-1} & \cdots & \Lambda_2 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \Lambda_{2i-1} & \Lambda_{2i-2} & \Lambda_{2i-3} & \cdots & \Lambda_i \end{pmatrix}$$

5.4.5 Matrix Input-Output Equations

Matrix input output equations are defined as follows:

$$Y_P = Y_{0/2i-1} = \Gamma_i X_P^d + H_i^d U_P + Y_P^s \tag{5.12}$$

$$Y_f = Y_{i/2i-1} = \Gamma_i X_f^d + H_i^d U_f + Y_f^s \tag{5.15}$$

$$X_i^d = A^i X_P^d + \Delta_i^d U_P \tag{5.14}$$

These equations can be easily formed by recursive substitution into the state space equations.

5.4.6 Orthogonal Projections

The projection of the row space of the future outputs onto the row spaces of the past and future inputs and the past outputs leads to equations which are function of the system matrices and the input-output block Hankel matrices. The proof of the following projections can be found in [72, 86].

$$Z_i = (Y_f / \begin{pmatrix} U_P \\ U_f \\ Y_P \end{pmatrix})$$

$$= \lim_{j\to\infty} \frac{1}{j} \left[Y_f \begin{pmatrix} U_P & U_f & Y_P \end{pmatrix} \left[\begin{pmatrix} U_P \\ U_f \\ Y_P \end{pmatrix} \begin{pmatrix} U_P^T & U_f^T & Y_P^T \end{pmatrix} \right]^{-1} \begin{pmatrix} U_P \\ U_f \\ Y_P \end{pmatrix} \right]$$

$$= \begin{pmatrix} \underbrace{L_i^1}_{ri\times pi} & \underbrace{L_i^2}_{ri\times pi} & \underbrace{L_i^3}_{ri\times ri} \end{pmatrix} \begin{pmatrix} U_P \\ U_f \\ Y_P \end{pmatrix} \tag{5.15}$$

where:

$$L_i^1 = \Gamma_i \left(\left[A^i - Q_i \Gamma_i \right] S(R^{-1})_{1/pi} + \Delta_i^d - Q_i H_i^d \right)$$

$$L_i^2 = H_i^d + \Gamma_i \left(\left[A^i - Q_i \Gamma_i \right] S(R^{-1})_{pi+1/2pi} \right)$$

$$L_i^3 = \Gamma_i Q_i$$

From which:

$$Z_i = \left(\underbrace{(\Gamma_i [\Delta_i^d - Q_i H_i^d] H_i^d) + \Gamma_i [A^i - Q_i \Gamma_i] S R^{-1}}_{ri\times 2pi} \middle| \underbrace{\Gamma_i Q_i}_{ri\times ri} \right) \begin{pmatrix} U_P \\ U_f \\ \hline Y_P \end{pmatrix}$$

$$Z_i = \Gamma_i \hat{X}_i + H_i^d U_{i/2i-1} \tag{5.16}$$

$$Z_{i+1} = (Y_{f+1} / \begin{pmatrix} U_{P+1} \\ U_{f+1} \\ Y_{P+1} \end{pmatrix}) = \lim_{j\to\infty} \frac{1}{j} (Y_{f+1} \begin{pmatrix} U_{P+1} \\ U_{f+1} \\ Y_{P+1} \end{pmatrix})$$

$$= \lim_{j\to\infty} \frac{1}{j} \left[Y_{f+1} \begin{pmatrix} U_{P+1} & U_{f+1} & Y_{P+1} \end{pmatrix} \left[\begin{pmatrix} U_{P+1} \\ U_{f+1} \\ Y_{P+1} \end{pmatrix} \begin{pmatrix} U^T_{P+1} & U^T_{f+1} & Y^T_{P+1} \end{pmatrix} \right]^{-1} \begin{pmatrix} U_{P+1} \\ U_{f+1} \\ Y_{P+1} \end{pmatrix} \right]$$

$$= \begin{pmatrix} \underbrace{L^1_{i+1}}_{r(i-1)\times p(i+1)} & \underbrace{L^2_{i+1}}_{r(i-1)\times p(i-1)} & \underbrace{L^3_{i+1}}_{r(i-1)\times r(i+1)} \end{pmatrix} \begin{pmatrix} U_{P+1} \\ U_{f+1} \\ Y_{P+1} \end{pmatrix} \tag{5.17}$$

where:

$$L^1_{i+1} = \Gamma_{i+1}\left(\left[A^{i+1} - Q_{i+1}\Gamma_{i+1}\right]S(R^{-1})_{1/pi} + \Delta^d_{i+1} - Q_{i+1}H^d_{i+1}\right)$$

$$L^2_{i+1} = H^d_{i+1} + \Gamma_{i+1}\left(\left[A^{i+1} - Q_{i+1}\Gamma_{i+1}\right]S(R^{-1})_{pi+1/2pi}\right)$$

$$L^3_{i+1} = \Gamma_{i+1}Q_{i+1}$$

From which,

$$Z_{i+1} = \left(\left(\Gamma_{i-1}[\Delta^d_{i+1} - Q_{i+1}H^d_{i+1}]H^d_{i-1}\right) + \Gamma_{i-1}[A^{i+1} - Q_{i+1}\Gamma_{i+1}]SR^{-1} \middle| \Gamma_{i-1}Q_{i+1}\right) \times \begin{pmatrix} \dfrac{U_{0/2i-1}}{Y_{0/i}} \end{pmatrix} \tag{5.18}$$

which can be written as

$$Z_{i+1} = \Gamma_{i-1}\hat{X}_{i+1} + H^d_{i-1}U_{i+1/2i-1} \tag{5.19}$$

where:

$$\hat{X}_i = \left(A^i - Q_i\Gamma_i \middle| \Delta^d_i - Q_iH^d_i \middle| Q_i\right) \begin{pmatrix} SR^{-1}U_{0/2i-1} \\ \dfrac{U_{0/i-1}}{Y_{0/i-1}} \end{pmatrix} \tag{5.20}$$

$$\hat{X}_{i+1} = \left(A^{i+1} - Q_{i+1}\Gamma_{i+1} \middle| \Delta^d_{i+1} - Q_{i+1}H^d_{i+1} \middle| Q_{i+1}\right)\left(\frac{\begin{matrix} SR^{-1}U_{0/2i-1} \\ U_{0/i} \end{matrix}}{Y_{0/i}}\right) \tag{5.21}$$

$$Q_i = \chi_i \Delta_i^{-1}$$

and,

$$\chi_i = A^i (P_d - SR^{-1}S^T)\Gamma_i^T + \Delta_i^s$$

$$\Delta_i = \Gamma_i (P_d - SR^{-1}S^T)\Gamma_i^T + L_i^s$$

The state sequences $\hat{X}_i$ and $\hat{X}_{i+1}$ can be interpreted as the states of a bank of non-steady state Kalman filter estimates (the proof can be found in [72, 85]). Thus one can write:

$$\hat{X}_{i+1} = A\hat{X}_i + BU_{i/i} + \begin{pmatrix} U_{0/2i-1} \\ Y_{0/i-1} \\ \hat{X}_i \end{pmatrix}^{\perp} \tag{5.22}$$

$$Y_{i/i} = C\hat{X}_i + DU_{i/i} + \begin{pmatrix} U_{i/2i-1} \\ Y_{0/i-1} \\ \hat{X}_i \end{pmatrix}^{\perp} \tag{5.23}$$

5.5 Algorithm Summary

1. Construct a concatenated block Hankel matrix consists of input and output data measurements.

$$H = \frac{1}{\sqrt{j}}\begin{pmatrix} U_{0/2i-1} \\ Y_{0/2i-1} \end{pmatrix} \tag{5.24}$$

2. Determine Γ_i, Γ_{i-1} and the order of the system n through the SVD of $Z_i - L_i^2 U_{i/2i-1}$:

$$Z_i - L_i^2 U_{i/2i-1} = \left(L_i^1 \middle| L_i^3\right)\begin{pmatrix} U_{0/i-1} \\ Y_{0/i-1} \end{pmatrix} = \begin{pmatrix} U_1 & U_2 \end{pmatrix}\begin{pmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{pmatrix} V^T$$

n=rank(Σ_1), $\Gamma_i = U_1 \Sigma_1^{1/2}$ and $\Gamma_{i-1} = \underline{\Gamma_i}$

where $\underline{\Gamma_i}$ is formed by deleting the last r rows of Γ_i

3. Determine the system matrices as follows.

Rearranging (5.16) and (5.19) to get:

$$\hat{X}_i = \Gamma_i^{-1}[Z_i - H_i^d U_{i/2i-1}]$$

$$\hat{X}_{i+1} = \Gamma_{i-1}^{-1}[Z_{i+1} - H_{i-1}^d U_{i+1/2i-1}]$$

From equation (5.22) and equation (5.23) we get:

$$\left(\frac{\hat{X}_{i+1}}{Y_{i/i}}\right) = \left(\frac{A}{C}\right)\hat{X}_i + \left(\frac{B}{D}\right)U_{i/i} + \begin{pmatrix} U_{0/2i-1} \\ Z_i \\ \hat{X}_i \end{pmatrix}^{\perp} \qquad (5.25)$$

Substitute for $\hat{X}_i$ and $\hat{X}_{i+1}$ into equation (5.25) to obtain:

$$\left(\frac{\Gamma_{i-1}^{-1} Z_{i+1}}{Y_{i/i}}\right) = \left(\frac{\kappa_{11} \quad \kappa_{12}}{\kappa_{21} \quad \kappa_{22}}\right)\left(\frac{\Gamma_i^{-1} Z_i}{Ui/2i-1}\right) + \left(\frac{\rho_1}{\rho_2}\right) \qquad (5.26)$$

where,

$$\left(\frac{\kappa_{11}}{\kappa_{21}}\right) = \left(\frac{A}{C}\right) \text{ and,}$$

$$\left(\frac{\kappa_{12}}{\kappa_{22}}\right) = \left(\frac{B - A\Gamma_i^{-1}\begin{pmatrix} D \\ \Gamma_{i-1}B \end{pmatrix}\Gamma_{i-1}^{-1}H_{i-1}^d - A\Gamma_i^{-1}\begin{pmatrix} 0 \\ H_{i-1}^d \end{pmatrix}}{D - C\Gamma_i^{-1}\left(\frac{D}{\Gamma_{i-1}}\right) - C\Gamma_i^{-1}\begin{pmatrix} 0 \\ H_{i-1}^d \end{pmatrix}}\right)$$

$\left(\frac{\rho_1}{\rho_2}\right)$ are the residuals from which the stochastic system is determined.

The only unknowns in equation (5.26) are A, C, κ_{12} and κ_{22} which can be found either by solving a set of linear equations (5.26) or by solving a least squares problem.

$$\min_{A,C,\kappa_{12},\kappa_{22}} \left\| \begin{pmatrix} \Gamma_{i-1}Z_i \\ \hline Y_{i/i} \end{pmatrix} - \begin{pmatrix} A & \kappa_{12} \\ \hline C & \kappa_{22} \end{pmatrix} \begin{pmatrix} \Gamma_i^{-1}Z_i \\ \hline U_{i/2i-1} \end{pmatrix} \right\|_F^2 \tag{5.27}$$

B and D matrices appear linearly in κ_{12} and κ_{22} from which they can be easily found.

5.5.1 Algorithm Implementation using RQ and SVD decomposition

1- Implementation of the algorithm in an efficient and stable way is achieved through the use of RQ factorisation [70] and Singular Value Decomposition SVD [72]. The concatenated block Hankel matrix defined by equation (4.24) can be decomposed into two matrices:

$$H = \frac{1}{\sqrt{j}} \begin{pmatrix} U_{0/2i-1} \\ Y_{0/2i-1} \end{pmatrix} = R_r . Q_r^T$$

$$= \begin{array}{c} pi \\ p \\ p(i-1) \\ ri \\ r \\ r(i-1) \end{array} \overbrace{\begin{pmatrix} U_{0/i-1} \\ U_{i/i} \\ U_{i+1/2i-1} \\ Y_{0/i-1} \\ Y_{i/i} \\ Y_{i+1/2i-1} \end{pmatrix}}^{j} = \begin{array}{cccccc} pi & p & p(i-1) & ri & r & r(i-1) \end{array} \atop \begin{pmatrix} R_{11} & 0 & 0 & 0 & 0 & 0 \\ R_{21} & R_{22} & 0 & 0 & 0 & 0 \\ R_{31} & R_{32} & R_{33} & 0 & 0 & 0 \\ R_{41} & R_{42} & R_{43} & R_{44} & 0 & 0 \\ R_{51} & R_{52} & R_{53} & R_{54} & R_{55} & 0 \\ R_{61} & R_{62} & R_{63} & R_{64} & R_{65} & R_{66} \end{pmatrix} \overbrace{\begin{pmatrix} Q_{r1}^T \\ Q_{r2}^T \\ Q_{r3}^T \\ Q_{r4}^T \\ Q_{r5}^T \\ Q_{r6}^T \end{pmatrix}}^{j} \tag{5.28}$$

Such that, the factor $R_r \in \Re^{2(p+r)i \times 2(p+r)i}$ and the matrix $Q_r^T \in \Re^{2(p+r)i \times j}$ is orthonormal. RQ decomposition is suitable to be used with the N4SID algorithms because it is easy to express all the geometric operations in terms of these decomposition. Also only the factor R_r is needed. This reduces the amount of computations and the memory requirements required to implement the N4SID algorithms. It is stated that [72] numerically efficient implementation of the N4SID algorithms is not unique.

$$Z_i = (Y_f / \begin{pmatrix} U_P \\ U_f \\ Y_P \end{pmatrix}) = R_{(5:6,1:4)}\, R^{-1}_{(1:4,1:4)} \begin{pmatrix} U_{0/2i-1} \\ Y_{0/i-1} \end{pmatrix} = \begin{pmatrix} L_i^1 & L_i^2 & L_i^1 \end{pmatrix} \begin{pmatrix} U_P \\ U_f \\ Y_P \end{pmatrix} \tag{5.29}$$

$$Z_{i+1} = (Y_{f+1} / \begin{pmatrix} U_{P+1} \\ U_{f+1} \\ Y_{P+1} \end{pmatrix}) = R_{(6:6,1:5)}\, R^{-1}_{(1:5,1:5)} \begin{pmatrix} U_{0/2i-1} \\ Y_{0/i} \end{pmatrix}$$

$$= \begin{pmatrix} L_{i+1}^1 & L_{i+1}^2 & L_{i+1}^1 \end{pmatrix} \begin{pmatrix} U_{P+1} \\ U_{f+1} \\ Y_{P+1} \end{pmatrix} \tag{5.30}$$

2- Determine Γ_i and n (the number of dominant Singular Values) by dropping $L_i^2 U_{i/2i-1}$ from Z_i and using SVD of $Z_i - L_i^2 U_{i/2i-1}$:

$$Z_i - L_i^2 U_{i/2i-1} = \begin{pmatrix} L_i^1 | L_i^3 \end{pmatrix} \begin{pmatrix} U_{0/i-1} \\ Y_{0/i-1} \end{pmatrix} = \begin{pmatrix} U_1 & U_2 \end{pmatrix} \begin{pmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{pmatrix} V^T$$

In terms of RQ factorisation,

$$\begin{pmatrix} U_1 & U_2 \end{pmatrix} \begin{pmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{pmatrix} V^T = \begin{pmatrix} L_i^1 & 0 & L_i^3 \end{pmatrix} R_{(1:4,;1:4)}\, Q^T_{r(1:4)}$$

From which,

n=rank(Σ_1), $\Gamma_i = U_1 \Sigma_1^{1/2}$ and $\Gamma_{i-1} = \underline{\Gamma_i}$

where $\underline{\Gamma_i}$ is formed by deleting the last *r* rows of Γ_i

3- Solve:

$$\min_{\kappa} \left\| \begin{pmatrix} \Sigma_1^{-1/2} (\underline{U}_1)^{-1} R_{(6:6,1:5)} \\ R_{(5:5,1:5)} \end{pmatrix} - \kappa \begin{pmatrix} \Sigma_1^{-1/2}\, U_1^T R_{(5:6,1:4)} \\ R_{(2:3,1:4)} \end{pmatrix} \right\|_F^2 \tag{5.31}$$

It can be shown that:

$$\begin{pmatrix} \Gamma_i^{-1} Z_i \\ U_{i/2i-1} \end{pmatrix} = \begin{pmatrix} \Sigma_1^{-1/2} U_1^T R_{(5:6,1:4)} \\ R_{(2:3,1:4)} \end{pmatrix} Q_{r(1:4)}^T$$

$$\begin{pmatrix} \Gamma_{i-1}^{-1} Z_{i+1} \\ Y_{i/i} \end{pmatrix} = \begin{pmatrix} \Sigma_1^{-1/2} (\underline{U}_1)^{-1} R_{(6:6,1:5)} \\ R_{(5:5,1:5)} \end{pmatrix} Q_{r(1:5)}^T$$

Equation (5.31) can be solved for k in a least squares sense. The first n columns of k are A and B stacked on top of each other. The next columns of k determine B and D. This algorithm is implemented in MATLAB [8] and is applied here to identify the linear models as shown in Appendix C.2

5.6 Perturbation Test Signals

In this section the problem of designing an optimal input signal for the purpose of the identification of linearized gas turbine model will be addressed. In practice the normal operating data records usually are not suitable for identification purposes due to the lack of persistency of excitation. The frequency of the input signal to the system to be identified must sufficiently span the passband of interest to excite all the dynamic modes [66]. Many test signals that have the property of persistency of excitation are available. Broad band input signals for instance, white noise, Pseudo-Random Binary signals (PRBS) and Discrete Interval Random Binary signals are examples of widely used test signals for identification with the property of persistency of excitation.

The first step in setting up an identification experiment is to determine the relevant inputs and outputs of the system to be identified. The inputs to the system must be sufficiently exciting (persistently exiting) in order to model all the dynamics of a system. These dynamics must be excited to be observed at the outputs. In most applications it is necessary to apply a well designed and sufficiently exciting signal. However, in many industrial applications it is often difficult or sometimes impossible to activate the inputs using rich signals due to different reasons such as loss of production, unallowable deviations from certain quality tolerances or as a result of safety regulations. In this case the input and the output data records that are obtained under normal operating conditions are being used to identify the system.

Pseudo-Random Binary signals (PRBS) are widely used in the identification of dynamic systems. They can be classified to binary and multi-level signals which in turn may be periodic or non-periodic. The essential requirements for all the Pseudo-random signals are to have impulsive or nearly impulsive autocorrelation function (ACF), the possibility to get uncorrelated signal sets to identify multivariable systems (signals with very low cross-correlation function (CCF) values) and the possibilities of formulating a signal with a desired energy-time distribution [66]. The main difference between periodic and non-periodic test

signals is that identification with periodic signal requires more time than the identification with non-periodic signal (approximately two periods of periodic signal against one period of the non-periodic signal). This makes identification using non-periodic signals faster. The problem associated with prolonged application of periodic test signals is the probability of drift in system parameter values.

5.7 Practical Aspects in Performing an Identification Experiment

Experiments design for the purpose of system identification is well explained in [75]. It is stated in [75] that the most important part in the identification process is the identification experiment. Data collected from a well-designed experiment can often result in a good model even using the simplest least-squares identification method. The practical aspects of designing an identification experiment can be found in [66, 75, 82, 87, 88]. A summary of these aspects is given here:

1. Output measurements of the sufficiently excited system must be taken carefully to reduce the effects of noise on the collected data.

2. Data from free run experiments is usually taken to collect information about the disturbances affecting the process. This is required to model the disturbance.

3. The time constants of the process are needed, to determine the duration of the experiment (the experiment must be longer than at least ten times the longest time constant). The time constants can be measured from a simple step response experiment.

4. The choice of sampling frequency (time) is very important. On one hand very slow sampling (long sampling intervals) compared to process time constants results in numerical sensitivity problems since the estimated poles are being clustered around origin point (0,i0) in the z-plane. On the other hand fast sampling leads to a nonminimum phase behaviour even if the original process is minimum phase (the poles and the zeros are being clustered near to z=1in the z-plane) [15, 75]. The sampling time can be found approximately from the step responses using the rule of thumb ($\tau_s = \tau_r / 4$) where τ_s is the sampling time and τ_r is the process rise time or equivalently using the frequency response (Bode plot) of the process ($\omega_s = 4\omega_b$) with ω_s is being the sampling frequency in (rad/sec), ω_b is the 3dB bandwidth and $\omega_s = 2\pi f_s$ with f_s being the sampling frequency in Hz [15].

5. Data pre-processing is necessary to reduce the effects of peaks, drifts, noise, over-sampling, dead time and outliers [75].

6. One of the difficulties in system identification is the selection of the identification algorithm because there exists many identification methods in the literature [66, 69]. Thus a suitable algorithm for a specific problem is required.

7. In the range of frequencies to be spanned the amplitude of the test signal must be large enough to be detected in the presence of noise and small enough to ensure working in the linear region around an operating point.

8- Model validation to check the model quality and how well it fits the measured data [66, 67, 72, 75].

5.8 PRBS Signal Design and Identification Experiment Set-up

One of the most useful types of periodic signals for process identification is a

Pseudo Random Binary signal [66]. The perturbation test signal that is used for exciting the dynamic modes of the closed-loop gas turbine system is an m-sequence Pseudo Random Binary Signal. The reason for using this signal is due to its popularity and the satisfactory results it gives. However, other signals such as multi-level Pseudo Random signals (ternary, quaternary, quenary, ... etc) may be used.

The generation of an m-sequence PRBS signal is carried out according to Figure 5.2 [90] where a shift register and modulo two addition are used for generation of a PRBS signal of any desired period then each bit of the period can be repeated k_a times as required. The non-zero mean signals are made zero mean by the subtraction of the non-zero mean value from the signal. Then, the desired level of the test signal is determined by multiplying the PRBS sequence by the required amplitude (approximately 10% of the steady state operating point). Finally, the steady state offsets are removed and the test signal is ready to be injected at the inputs of the system to be identified.

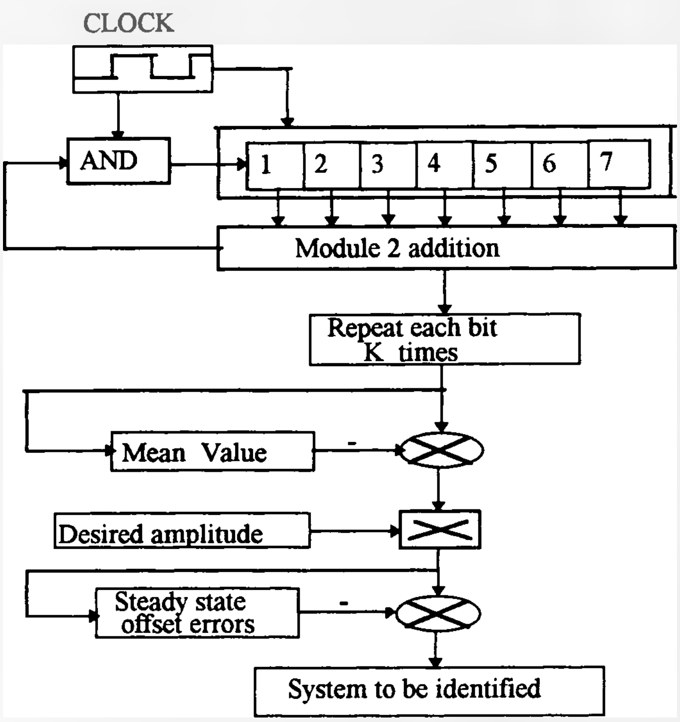


Figure 5.2. PRBS m-sequence generation and manipulation.

The gas turbine system has three input channels thus, three uncorrelated test signals need to be designed. Table 5.1 shows the three uncorrelated signals. The uncorrelated test signals are obtained according to the procedure given in [66, 88] as follows

1. First start with an m-sequence random binary signal of odd length N_a as the first input. In the gas turbine case N_a is taken to be seven which found to give satisfactory results.

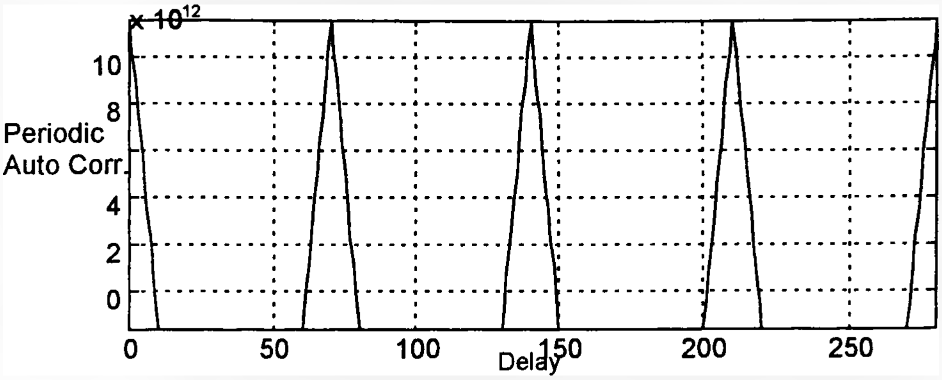
2. The second uncorrelated input is formed first by making an inverse-repeat binary signal based on the first input sequence by inverting every other digit in the PRBS by adding modulo 2 signal (0 1 0 1 0 1 0 1 . . .) to the first input sequence. The signal period becomes $2N_a$.

3. The third uncorrelated input sequence is formed by making an inverse-repeat binary signal based on the first input sequence and inverting digits (3, 4, 7, 8, ...) by adding modulo 2 signal (0 0 1 0 0 1 0 0 1 ...) to the original sequence. Then one's are transformed to +V and zero's to -V.

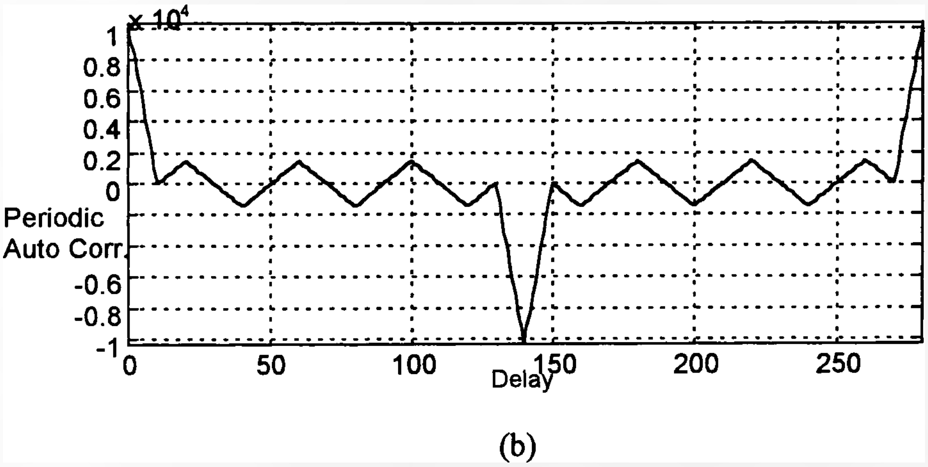
The procedure can be extended to get any number of uncorrelated PRBS sequences. A MATLAB program to generate the required PRBS m-sequence signals is given in Appendix C.1. Figure 5.3 shows the autocorrelation functions of the three uncorrelated signals. The advantage of uncorrelated signals is to suppress the nonlinear terms. The level of the ripples in the second and the third sequences is V^2/N_a which decreases by increasing N_a. The three designed signals are uncorrelated over their common period $4N_a\Delta t$ with impulsive autocorrelation functions as can be seen from Figure 5.3(a, b, c).

Table(5.1) m-sequence PRBS signals.

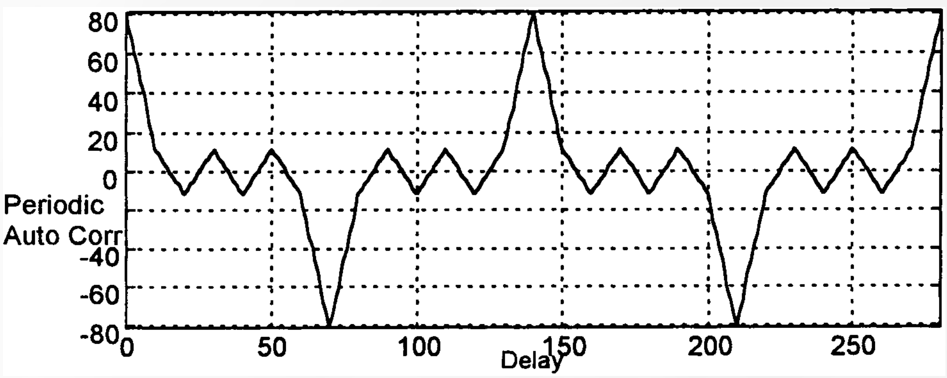
Number of clock pulses	m-sequence	u1	Modulo-2	Inverse-repeat sequence and	signal u2	Modulo-2	Inverse-repeat sequence and	signal u3
1	1	+V	0	1	+V	0	1	+V
2	0	-V	1	1	+V	0	0	-V
3	0	-V	0	0	-V	1	1	+V
4	1	+V	1	0	-V	1	0	-V
5	0	-V	0	0	-V	0	0	-V
6	1	+V	1	0	-V	0	1	+V
7	1	+V	0	1	+V	1	0	-V
1	1	+V	1	0	-V	1	0	-V
2	0	-V	0	0	-V	0	0	-V
3	0	-V	1	1	+V	0	0	-V
4	1	+V	0	1	+V	1	0	-V
5	0	-V	1	1	+V	1	1	+V
6	1	+V	0	1	+V	0	1	+V
7	1	+V	1	0	-V	0	1	+V
1	1	+V	0	1	+V	1	0	-V
2	0	-V	1	1	+V	1	1	+V
3	0	-V	0	0	-V	0	0	-V
4	1	+V	1	0	-V	0	1	+V
5	0	-V	0	0	-V	1	1	+V
6	1	+V	1	0	-V	1	0	-V
7	1	+V	0	1	+V	0	1	+V
1	1	+V	1	0	-V	0	1	+V
2	0	-V	0	0	-V	1	1	+V
3	0	-V	1	1	+V	1	1	+V
4	1	+V	0	1	+V	0	1	+V
5	0	-V	1	1	+V	0	0	-V
6	1	+V	0	1	+V	1	0	-V
7	1	+V	1	0	-V	1	0	-V



(a)



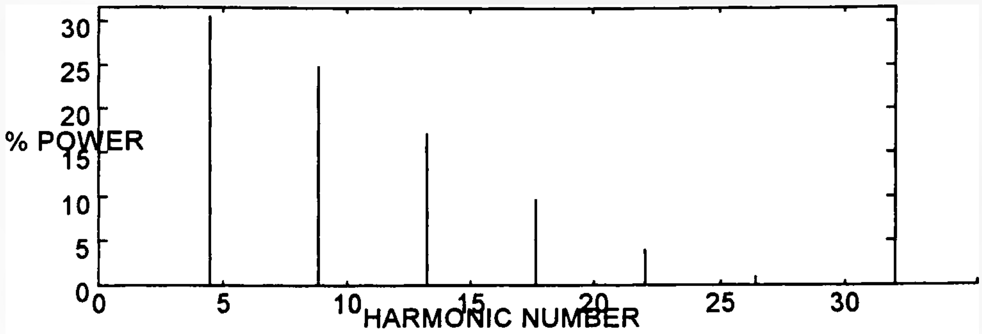
(b)



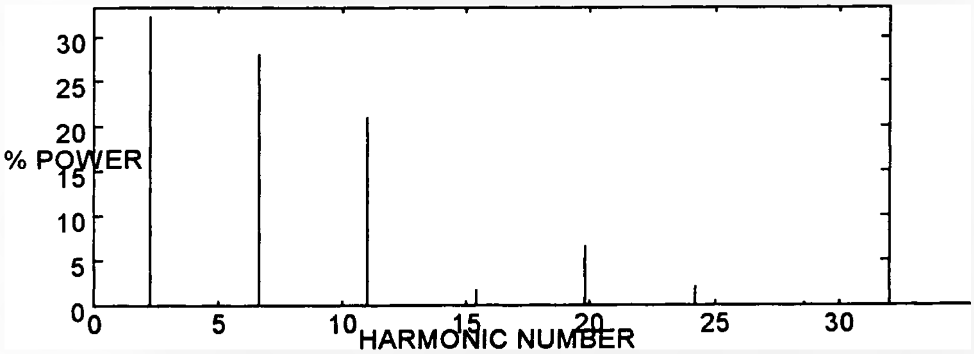
(c)

Figure 5.3 (a, b, c). Typical autocorrelation functions of the designed PRBS signals.

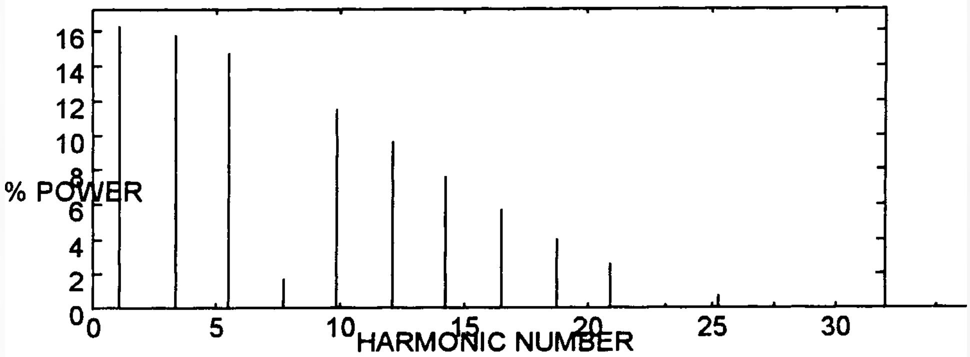
The percentage power distribution per harmonic for the three m-sequence PRBS signals is shown in Figures 5.4 (a, b, c). It can be seen that the power per harmonic is concentrated at low frequency.



(a)



(b)



(c)

Figure 5.4 (a, b, c). The percentage power per harmonic in the designed PRBS signals.

The designed m-sequence PRBS signals are used to excite the modes of the closed-loop system of the nonlinear model developed in Chapter three. The subspace algorithm discussed in Section 5.4 is used in the identification. The identification process is carried out according to Figure 5.5. The procedure is repeated at several operating points to get the linearised models of the power turbine closed-loop system. The identification experiments are designed and performed using SIMULINK building blocks and MATLAB programs (Appendix C.1 and Appendix C.2).

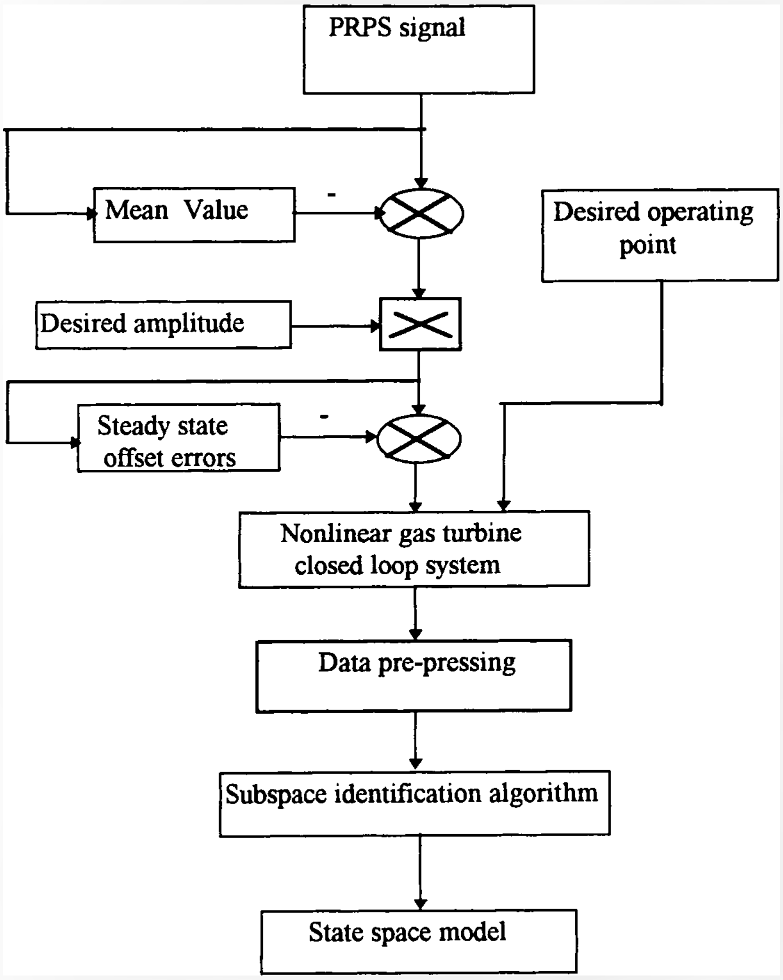
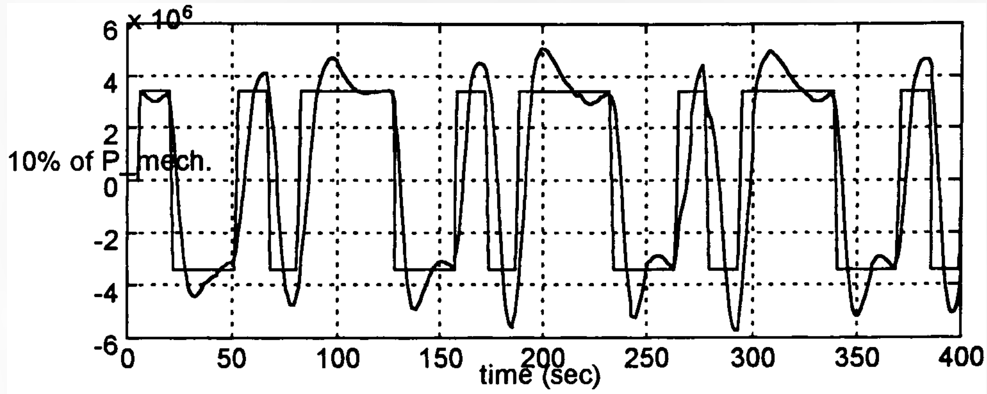
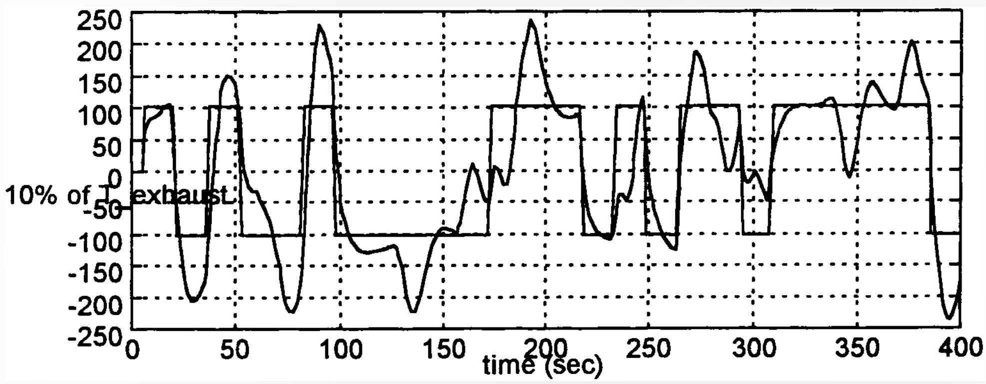


Figure 5.5 The identification procedure.

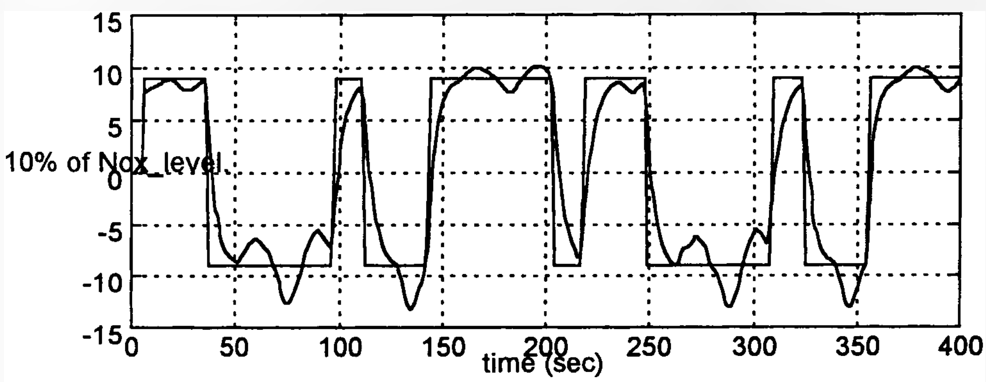
The responses of the system to the PRBS inputs are shown in Figures 5.6(a, b, c). These figures show that the output responses are distorted. This is as a result of strong coupling between the three channels which is reduced through the uncorrelated PRBS signals but not eliminated due to the presence of saturation and multiplication nonlinearities in the turbine system and the fact that the identified system is in closed-loop (feedback effects).



(a)



(b)



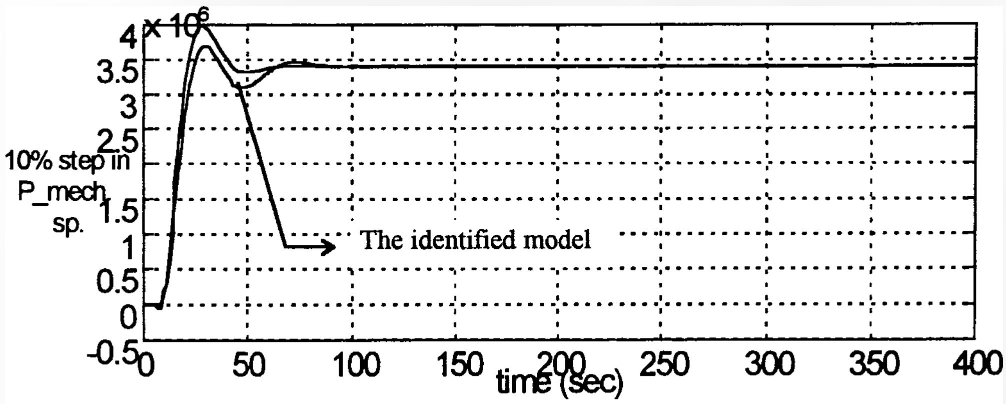
(c)

Figure 5.6(a, b, c) Typical PRBS signal inputs and the corresponding power turbine outputs.

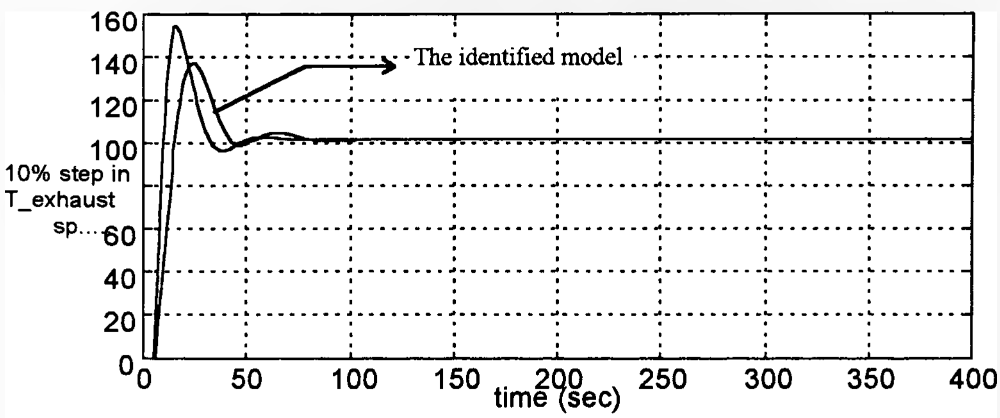
5.8.1 Simulation Results

5.8.1.1 Linear and Nonlinear Step Responses Comparison

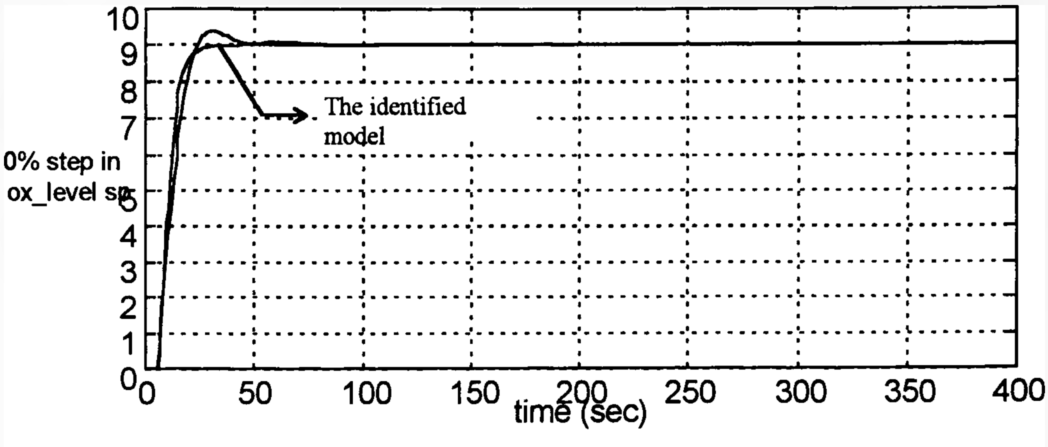
The output responses of one of the identified 6th order linear models, which is given in Appendix C.3, to step inputs are compared with that of the full order (10th order) linear model as depicted in Figures 5.7(a, b, c). It can be seen that the overshoot in the three outputs of the identified model are less than that in the full order model. This is due to model order reduction. This identified model is selected for control design.



(a)



(b)



(c)

Figure 5.7(a, b, c) Step responses of a typical identified model versus the full order model.

Part Two: Gain Scheduling GPC

5.9 Adaptive Predictive Controller Design

Gain scheduling is the direct adjustment procedure of the controller coefficients as a function of the system operating conditions. It is used in process industries which have dynamics that are nonlinear function of one or more operating parameter or that vary with time. Gain scheduling is a special kind of adaptation (open loop adaptation) [89].

The main advantage of gain scheduling is the quick response of the regulator to the changing conditions. Gain scheduling is widely used in different application such as high performance autopilot flight control, autopilot for ships, pH control, combustion control, etc. [14].

GPC controllers are based on fixed gain matrices which are used to calculate an optimal control sequence. This requires a state vector estimates of the process which can be found using a steady state Kalman filter algorithm. Solving GPC control problem at different operating point results in a set of fixed GPC and Kalman gain matrices. GPC control law can then be implemented at each operating point using a gain scheduling adaptive scheme.

The proposed technique is applied to the gas turbine power plant as a second layer of controllers. The main objective of designing the adaptive controller is to optimise the gas turbine power plant start-up and shut-down operations as well as to allow a smooth transition from one operating point to another and to improve the performance of the control design. There are alternative ways to design gain scheduling GPC supervisory control for example, using fuzzy logic GPC supervisory control technique [93] but these are not considered in the thesis.

The proposed adaptive predictive control system is composed of a set of multivariable GPC algorithms and Kalman filters to provide the turbine local PID controllers with optimised set-points as shown in Figure 5.8. The multivariable predictive control algorithm that is used for designing the adaptive predictive controller is given in Chapter two. A quadratic cost function is minimised at each operating point to calculate GPC gain matrices and Kalman filter gain. The linear models identification in the previous section that are assumed to cover the whole range of operation are used for designing six predictive controllers and Kalman filters. The MATLAB program to implement the controller is given in Appendix C.3

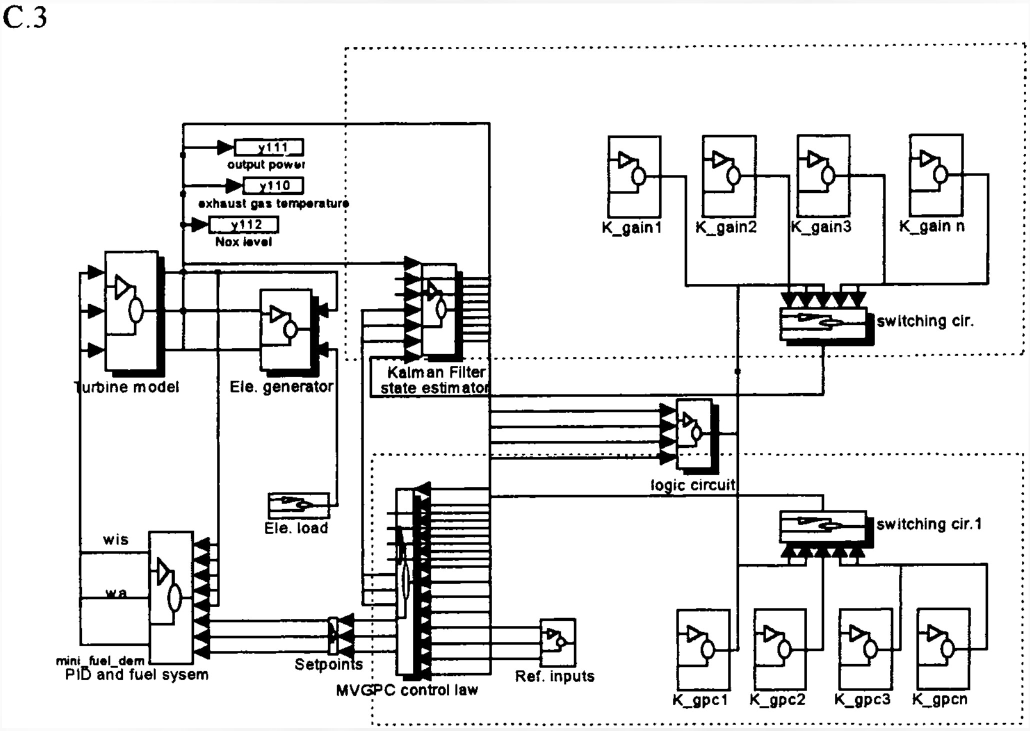


Figure 5.8 Adaptive predictive control system model.

5.9.1 Simulation Results

The power turbine is stabilised using PID regulators at an operating point which corresponds to fuel demand signal loss. The predictive controller can be either switched on manually or automatically to manoeuvre the set-points to new positions. The procedure followed to switch from one model to another is as follows:

1. Each model is assumed to operate in a linear region of $\pm 10\%$ around an operating point as shown in Figure 5.9.

2. Around each operating point some or all input and output variables can be changed. For instance, in the case of power turbine it is desired to keep Nox level at its minimum value while changing the output power to its maximum or minimum value within the current operating point. This can be easily achieved by setting the Nox reference input to zero and specifying any desired power reference input within $\pm 10\%$ around the operating point.

3. If condition one is violated the controller will switch to the next model and thus new operating point will occur. As shown in Figure 5.9 a typical operating point is between two regions, a positive region and a negative region. On one hand if condition one is violated towards the more negative region, that is one or more variable become <-10% of its value at the current operating point, then immediate switching to next model will happen. On the other hand violation of condition one towards the more positive region leads to switching to the next model in that region.

4. Condition one should be checked at each sample time to ensure the correct model is used to calculate the control law.

5. Steps one to five must be repeated at each operating point

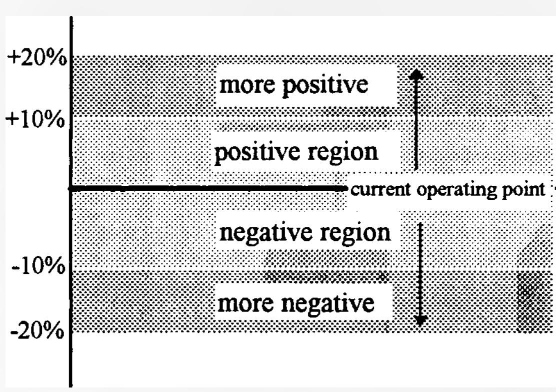


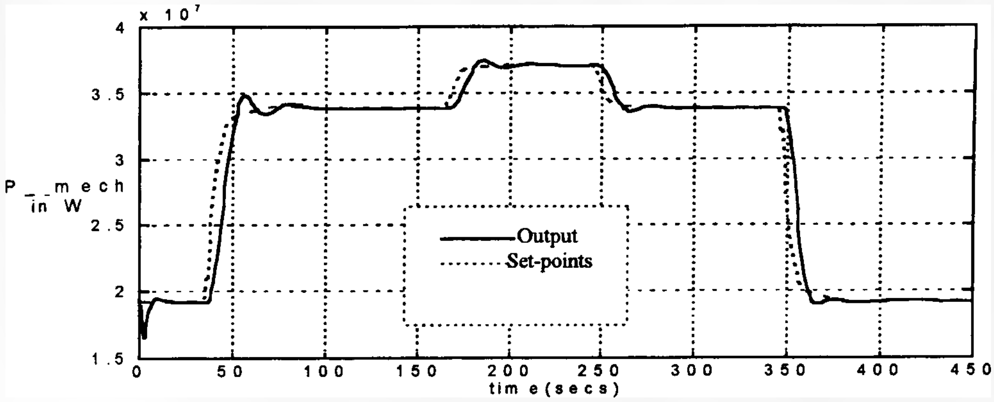
Figure 5.9: Models switching procedure.

Conditions for switching from one operating point to another are placed on the level of the controller outputs that is, the level of the optimised set-points. The state space linear models used for designing the controller as well as the MATLAB program for implementation of the adaptive controller are given in Appendix A5.

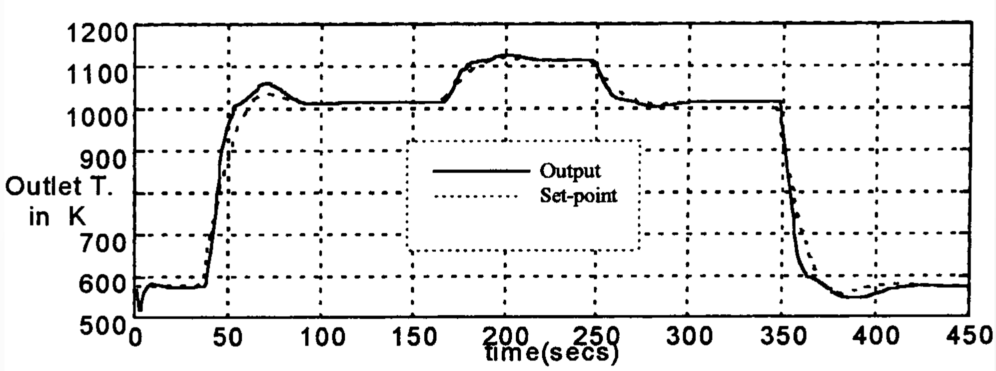
The designed adaptive predictive controller has been tested using step and ramp input signals. The output responses of the power turbine and the optimised set-points (controller outputs) for a step reference inputs are shown in Figure 5.10 (a, b, c).

It can be seen from Figure 5.10(a) that smooth transition from one operating point to another is obtained, the output power follow the optimised set-point with a slight delay, the optimised set-point is nearly overdamped but the output power have some overshoot and both the output power and the optimised set-point reached steady state almost at the same time.

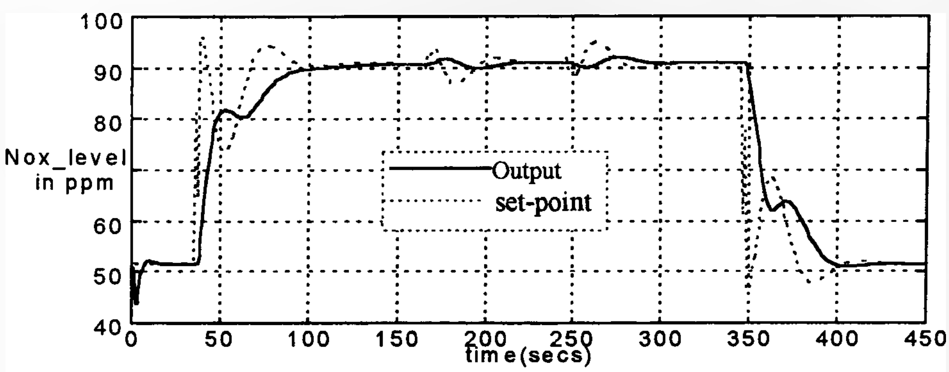
Figure 5.10(b) shows that the exhaust output temperature tracks the optimised set-point again with a slight delay, the transition from one operating point to another is smooth and the output temperature has a little more overshoot than the optimised set-point. Figure 5.10(c) shows that the optimised Nox set-point is fast and oscillatory both during start-up and shut-down. On the other hand the Nox output level is slow and overdamped. From Figure 5.10(a, b, c) it is obvious that good tracking of a step input is obtained and minimum start-up and shut-down time is achieved. After 130 sec a step input is injected to investigate the behaviour of the controller at steady state where good performance is obtained. However, it seems from Figure 5.10(a, b) that both the optimised set-points and the turbine outputs are slow in tracking the reference inputs.



(a)



(b)



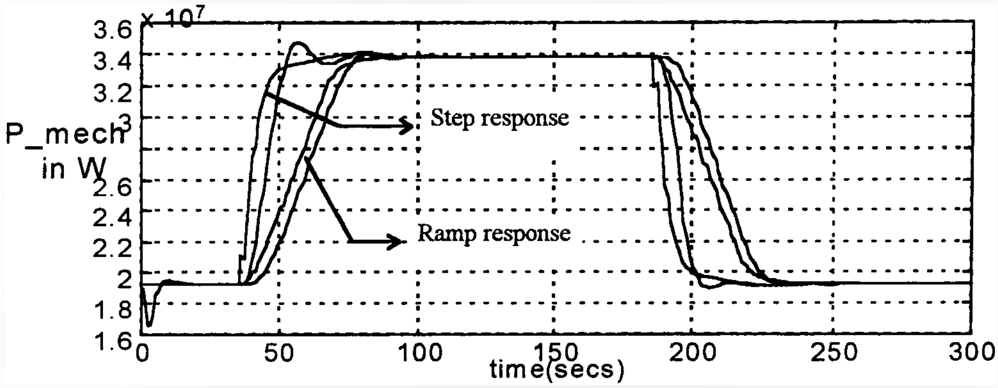
(c)

Figure 5.10(a, b, c): Start-up and shut-down operations of the turbine for step inputs

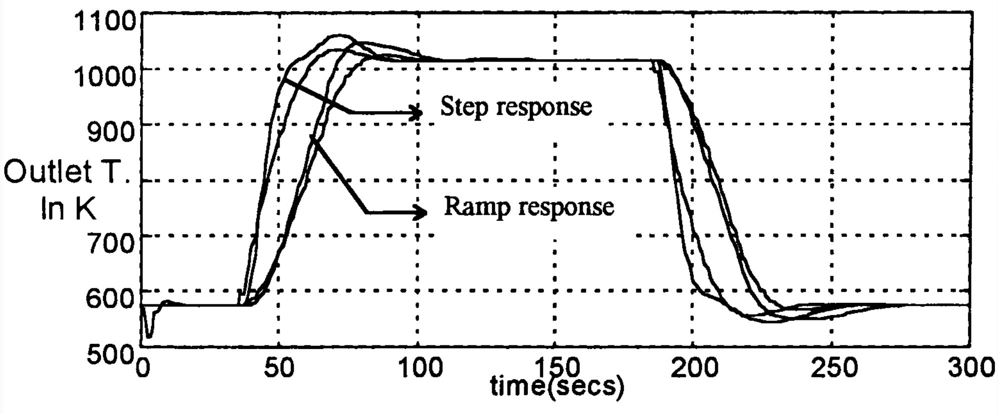
The outputs of the power turbine and the optimised set-points for both step and ramp reference inputs are compared in Figure 5.11(a, b, c). It can be observed from

Figure 5.11(a) that the overshoot in the output power is nearly disappeared and the time delay between the optimised set-point and the output power is slightly decreased with longer rise time during start-up and shut-down. Figure 5.11(b) shows that the overshoots in both the optimised exhaust temperature set-point and the output temperature for a ramp input are decreased and the time delay between the output temperature and the optimised set-point are also decreased. Changes in both the Nox optimised set-point and the Nox output level can be seen from Figure 5.11(c).

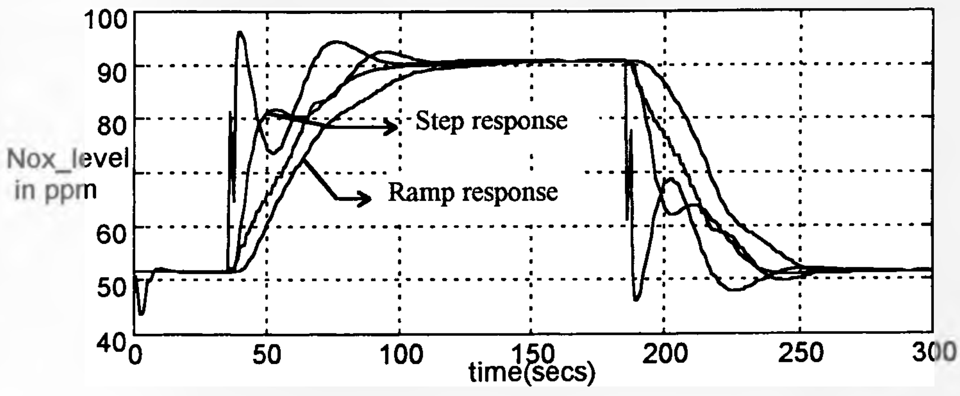
The fast and oscillatory optimised set-point became smooth and overdamped and the Nox output level became more overdamped.



(a)



(b)

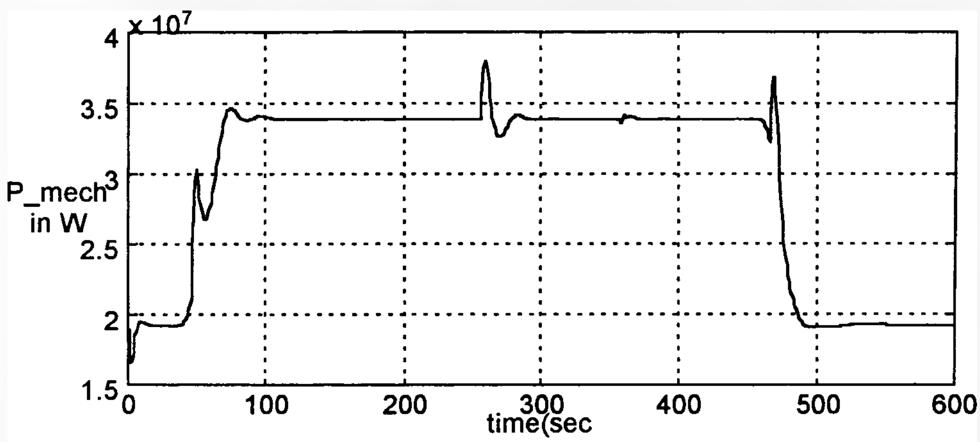


(c)

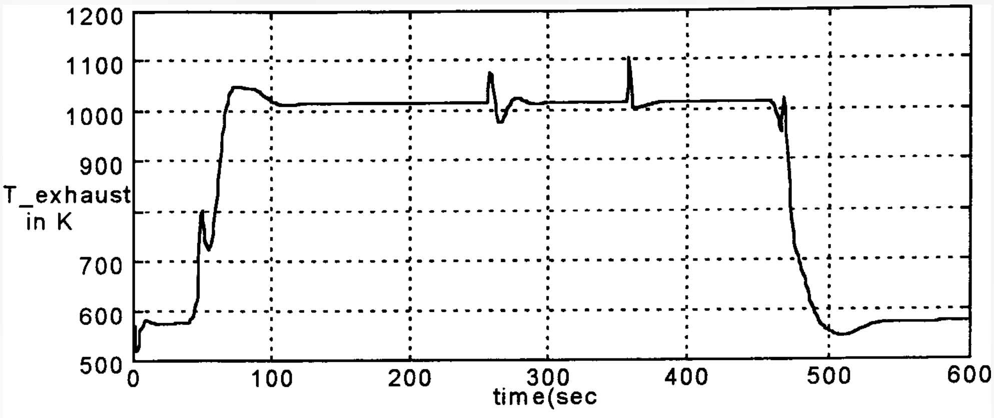
Figure 5.11(a, b, c) Start-up and shut-down operations of the turbine for ramp inputs.

5.10 Disturbance Rejection

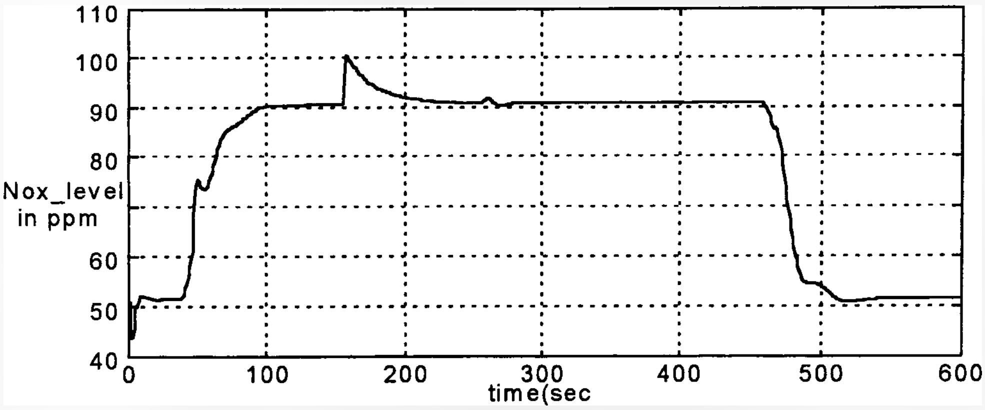
The performance of the adaptive predictive control system is investigated by injecting process disturbances (step) at the inputs of the turbine as can be seen from Figure 5.12(a, b, c). Three step disturbances are injected at the fuel demand input at different times, during start-up at 50 sec, during steady state at 250 sec and during shut-down at 470 sec. It can be seen that the disturbances are rejected with slight changes in the responses during start up and shut-down. Other two step disturbances are injected at air demand (350 sec) which is rejected quickly and at the injected steam demand (150 sec) which is rejected slowly.



(a)



(b)

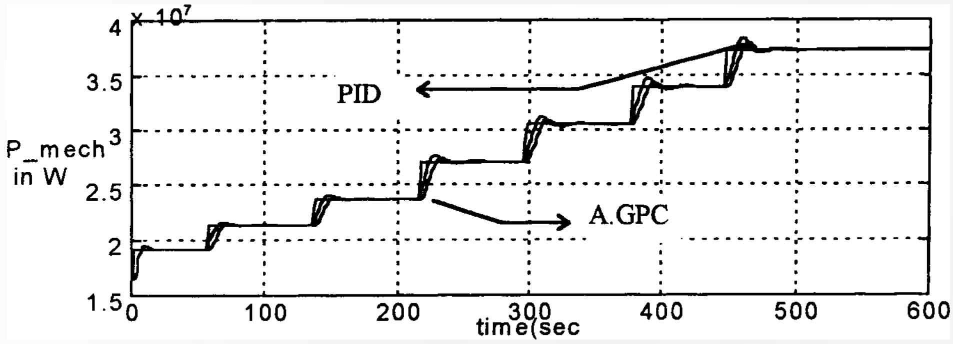


(c)

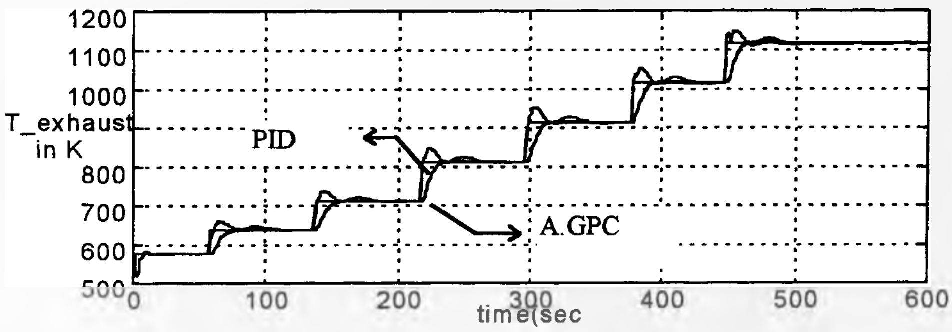
Figure 5.12(a, b, c) Disturbance rejection.

5.11 PID versus the Adaptive Predictive Control

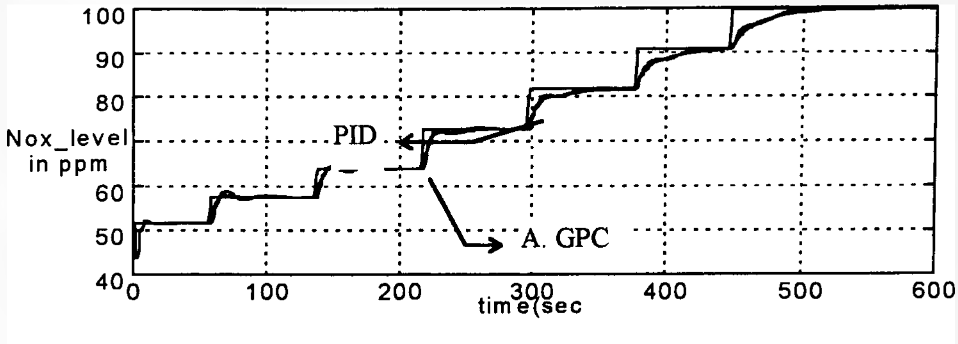
The comparison between the adaptive predictive control design and the conventional PID control design is made for two types of inputs, staircases input and step input as can be seen in Figure. 5.13(A, b, c) and Figure 5.14(a, b, c), respectively. It can be seen from Figure 5.13(a, b, c) that both the adaptive predictive controller and PID controller have achieved good tracking of the reference signal. The overshoots in the output power and the outlet temperature are nearly removed by the adaptive predictive controller. The performance of both controllers for the Nox level output are nearly the same. Figure 5.14(a, b, c) shows the start-up and the shut-down operations of the power turbine using step reference inputs for both controllers. PID control technique shows a deteriorated performance for large step inputs while the adaptive predictive controller can cope with large step inputs. The adaptive predictive controller shows smooth transition from one operating point to another and very low overshoots for step inputs compared with PID control. It is concluded that the predictive controller has optimised the start-up and the shut-down of the power turbine and has significantly improved the performance.



(a)

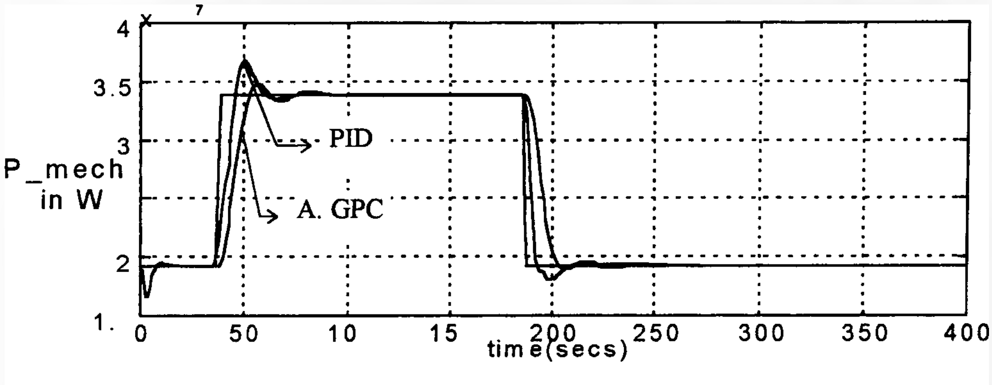


(b)

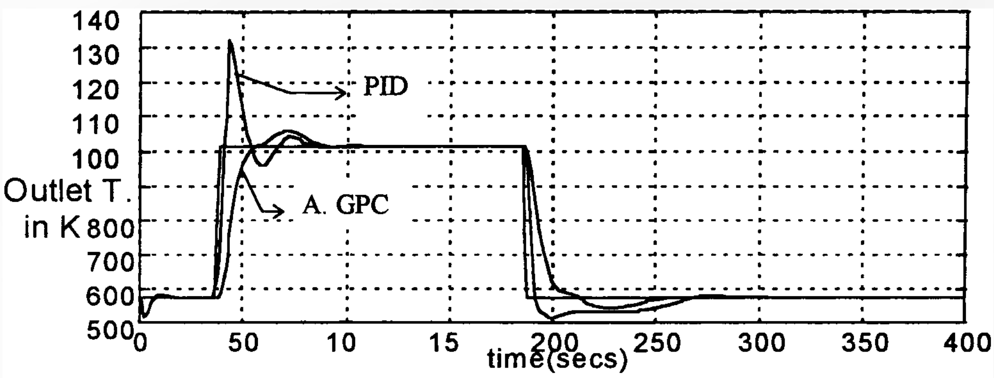


(c)

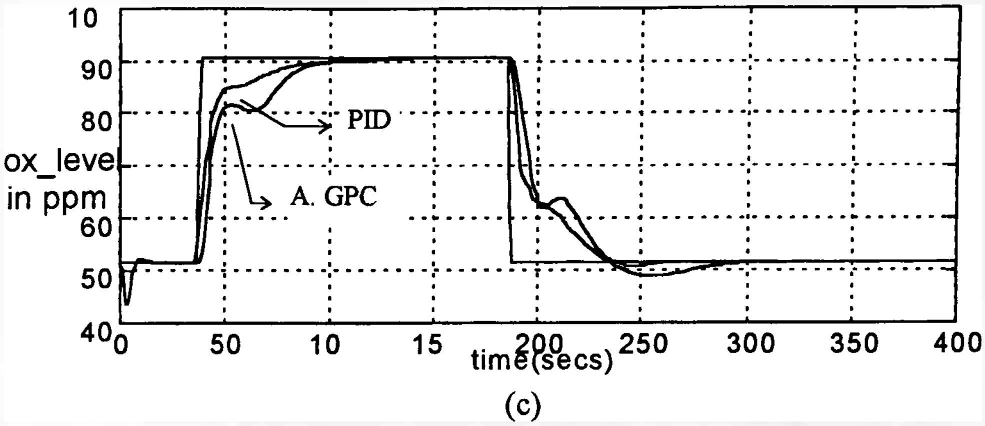
Figure 5.13(a, b, c) Power turbine output responses for staircase inputs.



(a)



(b)



(c)

Figure 5.14(a, b, c) Power turbine start-up and shut-down using PID and the adaptive predictive control techniques.

5.12 Constrained Adaptive Predictive Control Design

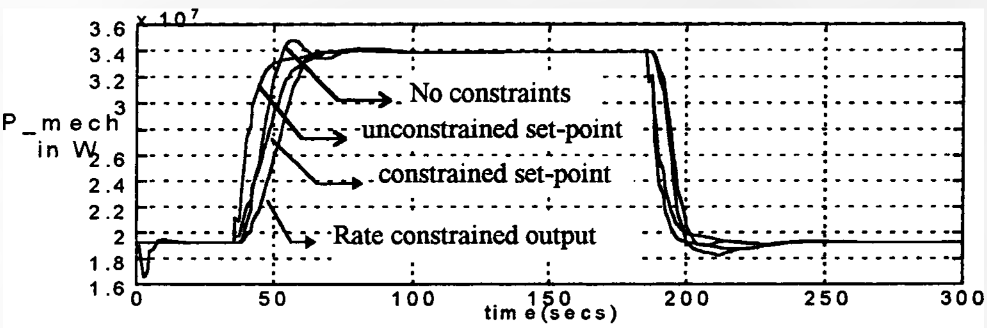
The proposed adaptive predictive controller is designed under different process and actuator constraints. Different constraints can be expressed in a compact form as an inequality. The optimisation of the cost function equation. (2.32) is made by solving a QP problem at each operating point by placing the inequality constraints as a part of the quadratic cost function [2].

In the design of the constrained adaptive predictive controller two types of constraints are considered namely, input rate constraints and overshoot constraints [3].

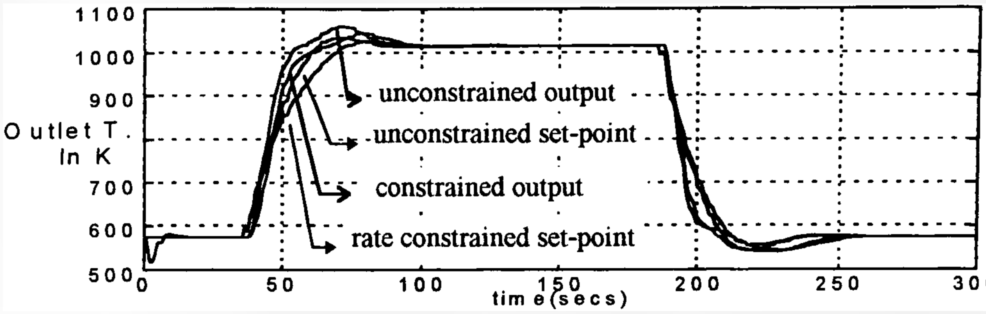
5.12.1 Simulation Results

The performance of the constrained adaptive predictive control design for start-up and shut-down of the power turbine is investigated under two types of constraints. The performance of the rate constrained control design is compared with the performance of the unconstrained control design as shown in Figure 5.15(a, b, c). It can be seen from Figure 5.15(a) that constraints on control increments result in overshoot reduction in both the power set-point and the turbine output power during start-up but with slower responses. However, during shut-down there is a slight increase in the overshoot in both the power set-point and the turbine output power with faster responses. The effects of the rate constraints on the outlet temperature responses are shown in Figure 5.15(b). It is obvious that the responses became slower with reduced overshoots. Figure 5.15(c) shows the effects of the rate constraints on the Nox output level. It can be seen that the rate constraints have different effects on the Nox set-point and the Nox level during start-up and shut-down. On one hand it is obvious that both of the set-point and the Nox output level are gradually increasing during start-up. On the other hand during shut-down one can see that the set-point changes very fast, faster than the set-point without rate constraints, with a significant increase in the overshoot and the Nox output level response became faster.

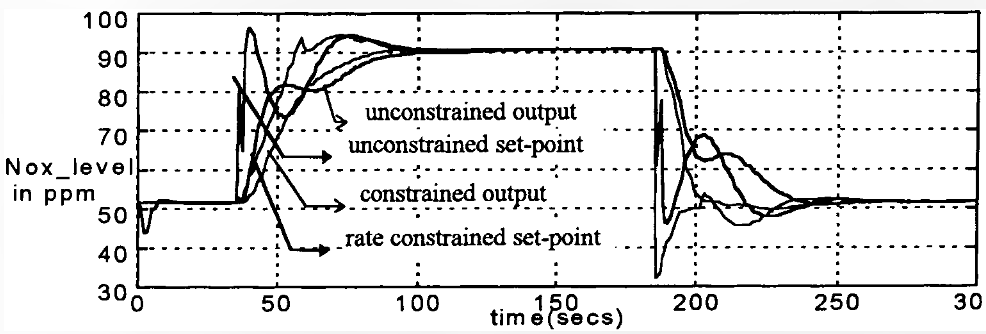
Overshoot constraints are also considered during cost function optimisation to improve the performance of the control design by reducing (or even eliminating) the undesired overshoots in the optimised set-points and the turbine outputs. Figure 5.16(a, b, c) shows the effects of the overshoot constraints on both the output power and the outlet temperature. It is obvious that overshoot constraints cause overshoots damping without affecting the speed of responses. The effects of overshoot constraints on the Nox output level and the optimised set-point are shown in Figure 5.16(c) from which one can see that the overshoot constraint has no effect on the first peak of the optimised set-point. However, the overshoot of the second peak is being decreased and the overshoot in the Nox output level is also decreased.



(a)

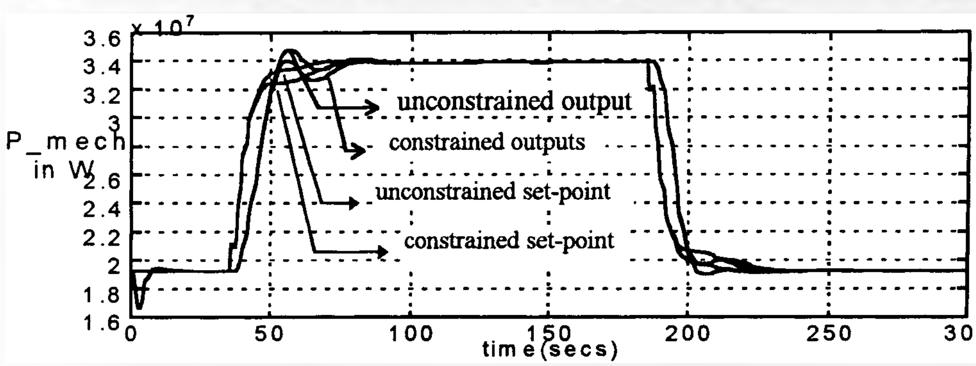


(b)

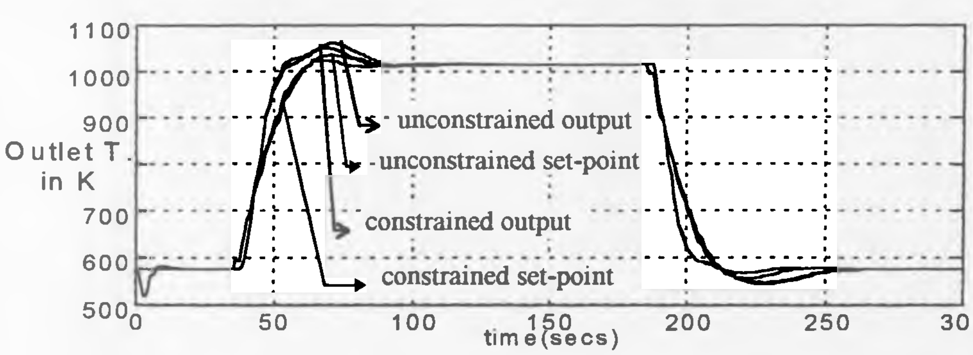


(c)

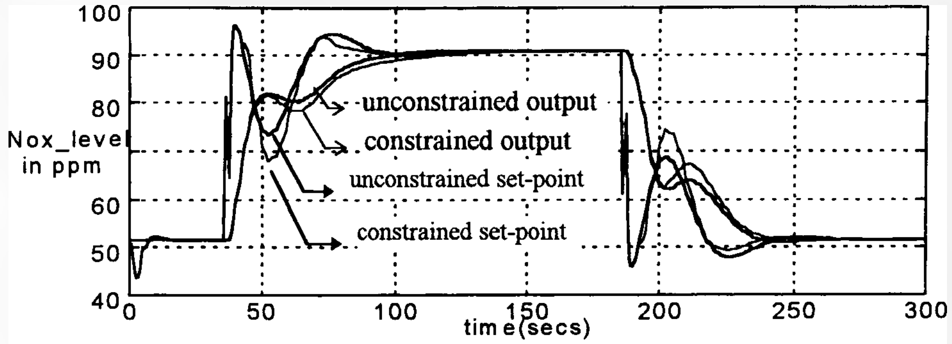
Figure 5.15(a, b, c) Rate constraints during start-up and shut-down.



(a)



(b)



(c)

Figure 5.16(a, b, c) Overshoot constraints during start up and shut-down.

5.13 Conclusions

A PRBS m-sequence signal was designed to excite the nonlinear gas turbine closed loop model dynamics for the purpose of identifying several linear models. One of the recently developed subspace-based identification methods was applied off-line to identify the linear models. The method has proved the elegance of the subspace-based identification technique. A predictive controller in an adaptive (gain scheduling) manner was proposed. The technique was applied to a nonlinear gas turbine model to optimise the start-up and the shut-down operation of the power turbine. Simulation results showed improved performance versus PID control technique. A constrained adaptive predictive controller was also designed. Simulations were made to demonstrate the systematic way of handling the actuator and the process constraints. Two types of constraints were considered in the design namely, control increment constraints and overshoot constraints. Simulation results showed an improved performance of the control design.

Chapter six
Conclusions and Future work

6.1 Conclusions

The comparison and the survey of some well-known MBPC algorithms showed that GPC algorithm seems to have a superiority over other MBPC algorithms in the sense that many MBPC algorithms can be seen as its special cases for certain values of controller tuning parameters. Moreover it can cope with a wider range of complex processes and can provide natural set-point decoupling within its control law formulation. This led to use the algorithm for gas turbine control in this thesis.

Formulation of GPC algorithms in state space representation for MIMO systems is readily available. Formulation of GPC algorithms in state space is simple in comparison with input-output formulation. The advanced control techniques and their numerical solutions are usually formulated in state space. State space models can be identified for complex multivariable processes with less problems using the recently developed subspace-based identification algorithms (see chapter 5) and the fact that most industrial plants are multivariable in nature have led to placing emphasis on the multivariable state space GPC algorithms rather than input-output based algorithms.

Simple measurements of the process performance can lead to initial values for the controller tuning parameters which with careful fine tuning can provide the desired performance.

The need for the analysis of the multivariable GPC closed-loop system to assess the performance and stability robustness of the control design led to a method for developing the indices for performance and stability robustness.

Most real life processes are subject to constraints which must be accounted for during control system design. For this the ability of GPC controller to handle process and actuator constraint in systematic way was exploited to design a controller to handle such constraints. A multivariable state space GPC constrained problem was solved as a QP optimisation problem for four types of constraints namely, constraints on control increments, constraints on control signal amplitude, constraints on overshoot and constraints on the outputs.

A simple tuning method was found to be effective to design the local PID controllers to stabilise the nonlinear model of the gas turbine power plant implemented on SIMULINK. Responses of both the nonlinear and the linear models were compared and simulation showed that the linear model obtained is a good approximation of the nonlinear model for the purpose of control design. The linear full order model was reduced, to minimise the amount of computations during GPC control law calculation, and an acceptable sixth order model was obtained. Simulation of the nonlinear model showed that the control system has achieved good reference input tracking and good disturbance rejection.

A second layer of control was introduced to optimally move the set-point of a multivariable feedback loop to a new level. GPC control technique was applied in a supervisory role as a second layer of control for the closed-loop nonlinear model of the power turbine to manoeuvre the local PID regulator set-points to new positions such that a quadratic cost function is minimised.

The developed method for GPC closed-loop analysis was applied to quantify the performance and stability robustness of the control design. Two studies were performed for the assessment of the performance and stability robustness. It was possible through the results of the two studies to make a compromise among the design objectives to design a good controller. Through these studies it was possible to know how improvements to the performance and the stability robustness of the control design can be made using different tuning parameters and weighting elements. The analysis carried out in chapter four revealed the following facts:

- The control design is stable over a wide range of tuning parameters and weighting elements. This provides a flexible design to achieve the desired performance.
- Interaction between different channels can be reduced by proper selection of the controller tuning parameters.
- Low frequency disturbance rejection can be improved in many ways by choosing suitable values for the controller tuning parameters.
- High frequency disturbance rejection is also a function of the controller tuning parameters, i.e. the effects of the high frequency disturbances can be suppressed by proper selection of the controller tuning parameters.
- The control activity depends on the controller tuning parameters that is, by changing the tuning parameters the control effort can be affected over the frequency range of interest.
- It was shown that several combinations of the controller tuning parameters can achieve the desired performance robustness.
- It was found that Kalman filter parameters could provide additional degrees of freedom for the control design by affecting the sensitivity functions.
- Stability robustness of the control design was found to be a function of various controller tuning parameters. The limits of the uncertainties that determine the stability robustness can be improved by the controller tuning parameters.
- Simulation results showed that the performance of the closed-loop system can be improved by the introduction of a second layer of GPC control.

A constrained predictive supervisory system for the power turbine was designed and simulation results showed that the performance can be improved by the introduction of suitable constraints.

A predictive controller in an adaptive (gain scheduling) manner was proposed. One of the recently developed subspace-based identification methods was applied off-line to identify the linear models for the purpose of the control design. The technique was applied to a nonlinear gas turbine model to optimise the start-up, steady state and the shut-down operation of the power turbine and to allow smooth transition from one operating point to another. Simulation results showed that the desired minimum time for start-up and shut-down operations was achieved with minimum overshoots. Simulation results also showed improved performance versus PID control technique. A constrained adaptive predictive controller was also designed. Simulations were made to demonstrate the systematic way of handling the actuator and the process constraints and it was shown that an improved performance of the control system can be achieved.

6.2 Further Work and Recommendations

1. Industrial indications are that to introduce a new generation of industrial gas turbine controllers. It is necessary to demonstrate improvements in implementation, a cost benefit in the hardware controller unit and significant improvements in operational performance. This would motivate a full feasibility study for the next phase of the research in which the measures of these cost benefits would have to be defined and assessed.
2. If a strict assessment of cost benefits showed promising results, a prototype development programme could be considered. For this test bed trials seem necessary and these would have to be carefully planned and conducted. The outcome of these trials would indicate the potential for further commercial development.
3. Some academic research aspects of multivariable state space GPC algorithms deserve further investigation. One area could be to extend the proposed (gain scheduling) method as a nonlinear GPC algorithm. Another could be to look at the potential for computational savings in the GPC and Subspace identification algorithms. Work in these areas would support any programme of industrial application trials.

References

[1] Clarke, D. W., Mohtadi, C. and Tuffs, P. S. *Generalised Predictive Control part I.* Automatica, Vol. 23, No. 2, pp. 137-148, 1987.

[2] Clarke, D. W., Mohtadi, C. and Tuffs, P. S. *Generalised Predictive Control part II.* Automatica, Vol. 23, No. 2, pp. 149-160, 1987.

[3] Clarke, D. W., Mohtadi, C. *Properties of Generalised Predictive Control.* Automatica, Vol. 25, No. 6, pp. 859-875, 1989.

[4] Clarke, D. W. *Generalised Predictive Control, internal Report.* Oxford University, Department of Engineering Science, Oxford OX1 3PJ, England.

[5] Bitmead, R. R., Gevers, M. and Wertz, V. *Adaptive Optimal Control., the Thinking Man's GPC.* Prentice-Hall, 1990.

[6] Clarke, D. W. Advances in Model-Based Predictive Control, Chapter: *Advances in Model-Based Predictive Control.* Oxford University Press, 1994.

[7] Pike, A. W. Grimble, M. J. Johnson, A. W. Ordys and Shakoor, S. *Predictive Control, Internal Report.* Industrial Control Centre, University of Strathclyde, Glasgow, Scotland, UK., 1995.

[8] Richalet, J., Rault, A., Testud, J. L. and Papon, J. *Model Predictive Heuristic Control: Applications to Industrial Processes.* Automatica, Vol. 14, pp. 413-428, 1978.

[9] De Keyser, R. M. C., Van De Velde, PH. G. A. and Dumortier, F. A. G. *A Comparative Study of Self-Adaptive Long-Range Predictive Control Methods.* Automatica, Vol. 24, No. 2, pp. 149-163, 1988.

[10] Cutler, C. R. and Ramker, B. C. *Dynamic Matrix Control a Computer Control Algorithm.* In Automatic Control Conference, San Francisco, 1980.

[11] Ydstie., B. E. *Extended Horizon Adaptive Control.* In Proc. 9th IFAC World Congress, Budapest, Hungary, 1984.

[12] De Keyser, R. M. C. and Van Cauwenberghe, A. R. *Extended Predictive Self-Adaptive Control.* In IFAC symp, on Identification and System Parameter Estimation, York, UK, pp. 1255-1260, 1985.

[13] Söeterboek, R. *Predictive Control. A unified Approach*. Prentice-Hall, 1992.

[14] Aström, K. J. and Wittenmark, B. *Computer-Controlled Systems. Theory and Design*. Prentice-Hall. Englewood Cliffs, N.J., 1990.

[15]Wellstead, P. E. and Zarrop, M. B. *Self-Tuning Systems. Control and Signal Processing*. John Wiley & Sons Ltd., 1991.

[16] Franklin, G. F., Powell, J. D. and Workman, M. L. *Digital Control of Dynamic Systems*. Addison-Wesley Publishing Company, Inc., 1990.

[17] Prada, C., Serrano, J., Vega, P. And Piera, M. A. Advances in Model-Based Predictive Control, Chapter: *A Comparative Study of DMC and GPC Controllers*. Oxford University Press, 1994.

[18] Camacho, E. F. and Bordons, C. *Model Predictive Control in the Process Industry*. Springer-verlag, 1994.

[19] Lee, T. H., Lai, W. C. and Kwek, K. H. *Generalised Predictive Control Incorporating Feedforward*. Report CI-89-3. National University of Singapore, Department of Electrical Engineering, July, 1989.

[20] Ordys, A. W. and Clarke, D. W. *A State-Space Description for GPC Controllers*. Int. J. Systems Sci., Vol. 24, No. 9, pp. 1727-1744, 1993.

[21] Clarke, D. W. and Zhang, L. *Does Long-Range Predictive Control Works*?. Oxford University, Department of Engineering Science.

[22] Dumont, G. A., Fu, Y. And Lu, G. Advances in Model-Based Predictive Control, Chapter: *Nonlinear Adaptive Generalised Predictive Control and Applications*. Oxford University Press, 1994.

[23] Zhu, Q. M., Warwick, K. and Douce, J. L. *Adaptive Generalised Predictive Controller for Nonlinear Systems*. IEE Proceedings-D, Vol. 138, No. 1, January 1991.

[24] Berlin, F. and Frank, P. M. Advances in Model-Based Predictive Control, Chapter: *Design and Realisation of a MIMO Predictive Controller for a 3-Tank System*. Oxford University Press, 1994.

[25] Krauss, P., Dass, K. And Rake, H. Advances in Model-Based Predictive Control, Chapter *Model-Based Predictive Controller with Kalman Filtering for State Estimation*. Oxford University Press, 1994.

[26] Krauss, P. and Rake, H. *Design for a Multivariable Predictive Controller in State-Space*. Proceedings of 3rd European Control Conference, Rome, Italy, September, 1995.

[27] Shahian, B. and Hassul, M. *Control System Design using MATLAB*. Englewood Cliffs, N.J.: Prentice Hall, 1993.

[28] Camacho, E. F. *Constrained Generalised Predictive Control*. IEEE Trans. On Automatic Control, Vol. 38, No. 2, pp. 327-332, 1993.

[29] Kutnetsov, A. G. and Clarke, D. W. Advances in Model-Based Predictive Control, Chapter: *Application of Constrained GPC for Improving Performance of Controlled Plants*. Oxford University Press, 1994.

[30] Tsang, T. T. C. and Clarke, D. W. *Generalised Predictive Control with Input Constraints*. IEE Proceedings-D, Vol. 135, No. 6, November, 1988.

[31] Clarke, D. W. *Implementation of Adaptive Generalised Predictive Control, Internal Report*. Oxford University, Department of Engineering Science, Oxford OX1 3PJ, UK.

[32] McIntosh, A. R., Shah, S. L. and Fisher, D. G. *Analysis and Tuning of Adaptive Generalised Predictive Control*. The Canadian Journal of Chemical Engineering, Vol. 69, February, 1991.

[33] Robinson, B. D. and Clarke, D. W. *Robustness Effects of a Prefilter in Generalised Predictive Control*. IEE Proceedings-D, Vol. 138, No. 1, January, 1991.

[34] Banerjee, P. and shah, S. L. *Tuning Guidelines for Robust Generalised Predictive Control*. Proceedings of the 31st Conference on Decision and Control, Tucson, Arizona, December, 1992.

[35] Dumor, D. and Boucher, P. Advances in Model-Based Predictive Control, Chapter: *Predictive Control Application in the Machine Tool Field*. Oxford University Press, 1994.

[36] Camacho, E. F. and Berenguel, M. Advances in Model-Based Predictive Control, Chapter: *Application of Generalised Predictive Control to a Solar Power Plant*. Oxford University Press, 1994.

[37] Lu, S. and Hogg, B. W. Advances in Model-Based Predictive Control, Chapter: *Model-based Predictive Control of a Thermal Power Plant*. Oxford University Press, 1994.

[38] Linkens, D. A. and Mahfouf, M. Advances in Model-Based Predictive Control, Chapter: *Generalised Predictive Control in Clinical Anaesthesia*. Oxford University Press, 1994.

[39].Ray, W. H. *Advanced Process Control*. McGraw-Hill Co. Ltd., 1981.

[40] Maciejowski, J. M. *Multivariable Feedback Design*. Addison-wesley Publishing Co., Inc. UK, 1989.

[41] Skogestad, S. and Postlethwaite, I. *Multivariable Feedback Control. Analysis and Design*. John wiley & Sons Ltd, 1996.

[42] Kassapakis, E.G. and Warwick, K. Advances in Model-Based Predictive Control, Chapter: *Predictive Algorithm for Autopilot Design*. Oxford University Press, 1994.

[43] Desanj, D. S. and Katebi, M. R. *Robust Multivariable Predictive Control Design*. Euraco Workshop Florence, Italy, September, 1995.

[44] Mohtadi, C., Shah, S. L. and Fisher, D. G. *Frequency Response Characteristics of MIMO GPC*. Int. Journal of Control, Vol. 55, No. 4, pp. 811-900, 1992.

[45] Katebi, R. M. *Integrated Predictive Control Design for Autonomous Underwater Vehicles*. 13th IFAC World Congress San Francisco, July, 1996.

[46] Lugand, P. and Parietti, C. *Combined Cycle Plants with Frame 9F Gas Turbines*. Transactions of the ASME Journal of Engineering for Gas Turbines and Power, Vol. 113, pp. 475-481, October, 1991.

[47] Harman, R. T. C. *Gas Turbine Engineering. Applications, Cycles and Characteristics*. The MacMillan Press Ltd, 1981.

[48] Cohen, H., Rogers, G. F. C. and Saravanamuttoo, H. I. H. *Gas Turbine Theory*. Longman Group Limited, 1987.

[49] Yaccobucci, R. B. *A Control System Retrofit for a GE Frame 5 Turbine/Generator Unit*. IEEE Transactions on Energy Conversion, Vol. 6, No. 2, June, 1991.

[50] Joel, R. *Basic Engineering Thermodynamics in SI Units*. Longman Group Limited, 1971.

[51] Lefebvre, A. H. *Gas Turbine Combustion*. Washington Hemishere Pub. Corp., 1983.

[52] Rowin, W. I. *Speedtronic Mark IV Control System*. Alsthom Gas Turbine Reference Library (AGTR 880), 1988.

[53] Rowin, W. I. *Simplified Mathematical Representations of Heavy-Duty Gas Turbines*. Transactions of the ASME Journal of Engineering for Gas Turbines and Power, Vol. 105, pp. 865-869, October, 1983.

[54] Ahluwalia, K. S. *Dynamic Modelling of a Combined-Cycle Plant*. Transactions of the ASME Journal of Engineering for Gas Turbines and Power, Vol. 112, pp. 164-167, April, 1990.

[55] Hunge, W. W. *Dynamic Simulation of Gas Turbine Generating Unit*. IEE Proceedings-C, Vol. 138, No. 4, pp. 342-350, July, 1991.

[56] Maffezzoni, C. *Issues in Modelling and Simulation of Power Plants*. IFAC Control of Power Plants and Power Systems, Munich, Germany, 1992.

[57] Ames, R. L. *A. C. Generators: Design and Applications*. John Wiley & Sons Inc., 1990.

[58] Hussain A. and seiffi, H. *Dynamic Modelling of Single Shaft Gas Turbine*. IFAC Control of Power Plants and Power Systems, Munich, Germany, 1992.

[59] Ordys, A. W., Pike, A. W., Johnson, M. A., Katebi, R. M. and Grimble, M. J. *Modelling and Simulation of Power Generation Plants*. Springer-Verlag London Limited, 1994.

[60] Frankline, G. F., Powell, J. D. and Naeini, A. E. *Feedback Control of Dynamic Systems*. Addison-Wesley Publishing Company, Inc. 1994.

[61] Aström, K. and Haggland, J. *Automatic Tuning of PID Regulators*.

[62] Klafter, R. D., Chmielewski, T. A. and Negin, M. *Robotics Engineering an Integrated Approach*. Prentice-Hall, UK., 1989.

[63] *Simulink Dynamic System Simulation Software*. User's Guide, The Math Works, Inc., 1992.

[64] Seborg, D., Edgar T.F. and Mellichamp, D. A. *Process Dynamic and Control*. John Wiley & Sons, 1989.

[65] Hsia, T. C. *System Identification*. D. C. Heath and Company, 1977.

[66] Godfrey, K. *Perturbation Signals for System Identification*. Prentice-Hall International (UK) Ltd, 1993.

[67] Stoica, P. and Soderstrom, T. *System Identification*. Prentice-Hall International (UK)Ltd. 1989.

[68] Eykhoff, P. *System Identification*. John Wiley, New York, 1974.

[69] Ljung, L. *System Identification: Theory for the User*. Prentice-Hall: Englewood Cliffs, NJ, 1987.

[70] Noble, B. and Daniel, I. W. *Applied Linear Algebra*. Prentice-Hall Englewood Cliffs, N.J., 1988.

[71] Graupe, D. *Identification of Systems*. Second Edition Robert E. Krieger Publishing Company Huntington, New York, 1976.

[72] Overschee, P. V. and De Moor, B. *Subspace Identification for Linear Systems: Theory Implementation Applications*. Kluwer Academic Publishers, 1996.

[73] Overschee, P. V. and De Moor, B. *N4SID: Subspace Algorithms for the Identification of Combined Deterministic-Stochastic Systems*. Automatica, Vol. 30, No. 1, pp. 75-93, 1994.

[74] Overschee, P. V. and De Moor, B. *Subspace Algorithms for the Stochastic Identification Problem*. Automatica, Vol. 29, No. 3, pp. 649-660, 1993.

[75] Zhu, Y. and Backx, T. *Identification of Multivariable Industrial Processes for Simulation, Diagnosis and Control*. Springer_Verlag London Limited, 1993.

[76] Viberg, M. *Subspace-based Methods for the Identification of Linear Time-Invariant Systems*. Automatica, Vol. 31, No. 12, pp. 1835-1851, 1995.

[77] Verhaegen, M. *Identification of the Deterministic Part of MIMO State-Space Models given in Innovations From Input-Output Data.* Automatica, Vol. 30, No. 1, pp. 61-74, 1994.

[78] Sharaf, S. M. Z., Hogg, B. W., Abdalla, O. H. and El-Sayed, M. L. *Multivariable Adaptive Controller for a Turbogenerator*. IEE Proceedings-D, Vol. 133, No. 2, pp. 83-89, March, 1986.

[79] Moonen, M., De Moor, B., Vandenberghe, L. And Vandewalle, J. *On- and Off-line Identification of Linear State-Space Models*. Int. Journal of Control, Vol. 49, No. 1, pp. 219-232, 1989.

[80] Mckelvey T. *Identification of State-Space Models from Time and Frequency Data*. Linköping Studies in Science and Technology. Dissertations No. 380., 1995.

[81] De Moor, B. and Overschee, P. V. *Numerical Algorithms for Subspace State-Space System Identification*. Trends in Control A European Perspective, Springer-Verlag, 1994.

[82] Chen, C. H., Kerlin, T. W. And Fry, D. N. *Experiments with Binary Periodic Signals for Dynamic Testing at the HFIR2*. IEEE Trans. on Nucl. Sci. Vol. 19, pp. 828-836, 1972.

[83] De moor, B. *Mathematical Concepts and Techniques for Modelling of Static and Dynamic Systems*. PhD Thesis, 1988.

[84] Ovrschee, V. P. and De Moor, B. *A Unifying Theorem for Three Subspace System Identification Algorithms*. Automatica Vol. 31, No. 12, pp 1853-1864, 1995.

[85] Verhaegen, M. and Dewilde, P. *Subspace Model Identification part 1*, 2 and 3. Int. Journal of Control, Vol. 56, No. 5, pp. 1187-1210, 1992.

[86] Overschee, P. V. and De Moor, B. *N4SID: Subspace Algorithms for the Identification of Combined Deterministic and Stochastic Systems*. ESAT/SISTA, 34-I, 1992.

[87] Bezanson, L. W. and Harris, S. L. *Identification and Control of an Extruder using Multivariable Algorithms*. IEE Proceedings-D, Vol. 133, No. 4, July, 1986.

[88] Briggs, P. A. N. and Godfrey, K. R. *Pseudorandom Signals for the Dynamic Analysis of Multivariable Systems*. IEE Proceedings-D, Vol. 113, No. 7, July, 1966.

[89] Aström, K. J. and Wittenmark, B. *Adaptive Control*. Addison-Wesley Publishing Company, 1989.

[90] Hill, D. *System Identification of Gas Turbines*. Thesis for the degree of Doctor of Philosophy, University of Birmingham. December, 1993.

[91] De Madrid, A. P., Santos, Dormido, S. and Morilla, F. Advances in Model-Based Predictive Control, Chapter: *Constrained generalised predictive control with dynamic programming*. Oxford University Press, 1994.

[92] Kanjilal, P. P. *Adaptive Prediction and PredictiveCcontrol*. Peter peregrinus Ltd., 1995.

[93] Brdys, M. A. and Quevedo, J. *Fuzzy Logic Supervisory Control of Induction Motor Under Full Range of Operating Conditions*. ESTYLF, 1995.

Appendix A.1

****

Gas Turbine Model Data

1. COMPRESSOR DATA

Parameters

Ao=.01 [m^2]; Zetrans=.99; zet_ic=.9; cp=1005 [J/(kg °K)];

gam_a=1.4; Rair=287 [J/(kg °K)]; r_{cold}=10

Inputs

wa=46.137 [k_g/sec]; rho_i=1.21 [k_g/m^3];

Pcin=1×10^5 [Pa]; Tcin=288 [°K]

2. COMBUSTOR DATA

Parameters

Cpg=1147 [J/(kg °K)]; k1=1; Am=1 [m^2];

delh25=-4.0*10^7 [J/kg]; Rcg=287 [J/(kg °K)];

href=1.2041*10^6 [J/kg]; Tref=1000 [°K];

Cpa=1005 [J/(kg °K)]; Cps=2160 [J/(kg °K)]; k2=1;

Inputs

Pcout=1.0033*10^6 [Pa]; Tcout=598.8 [°K];

Tis=601.69 [°K]; wa=46.137 [k_g/sec];

wf=1.8484 [k_g/sec]; Wis=.18566 [k_g/sec];

3. TURBINE DATA

Parameters and Constants;

zet_it=.9; Aot=.14 [m^2]; Cpg=1147 [J/(kg °K)];

gam_cg=1.333; Rcg=287 [J/(kg °K)]; r_{told}=0.1

Inputs;

h_{tin}=$1.1804*10^6$ [J/kg]; Ttin=1892 [°K]; Ptin=1.0027*10^6 [Pa];

Pc=1.4547*10^7 [W]; Wg=46.939 [k_g/sec];

Appendix A.2

****

Compressor Model

```
%S-Function    **Simulink Block**
function [sys,x0]=alicompr(t,x,u,flag)
if flag==3;
wa=u(1);  rho_i=u(2);  Pcin=u(3);  Tcin=u(4);  Ao=u(5);
Zetrans=u(6);  zet_ic=u(7);  cp=u(8);  gam_a=u(9);  Rair=u(10)
[Pc,Tcout,Pcout]=compres1(wa,rho_i,Pcin,Tcin, Ao, Zetrans, zet_ic, cp,
gam_a, Rair);
sys=[Pc,Tcout,Pcout];
elseif flag==0;
%La to return no. of continuous states; % da No. of discrete states;
% ma No.of outputs; % na No. of inputs; % dra No. of discontinuous roots;
%To find algebriac loop any(Pcout~=0);;
La=0; da=0; ma=3; na=10; dra=0;
sys=[0,0,ma,na,0,any(Pcout~=0)];
else;sys=[];end ;
% m.file     **This subroutine simulates the compressor**
function [Pc,Tcout,Pcout]=compres1(wa,rho_i,Pcin,Tcin)
R=0;  rc2=12;  q1=1e-20;  R=R+1;
% Solve equation (3.6) using Newton-Raphson method
% let gam_a=Cpa/Cva;
ma=gam_a/(gam_a-(gam_a-1)/zet_ic);
ma1=(gam_a-1)/(zet_ic*gam_a);
for i=1:1:50;
f=(rc2^(2/ma)+((zet_ic*(ma-1))/(2*ma*rho_i*Pcin))*(wa/Ao)^2-
rc2^((ma+1)/ma));
G=((2/ma)*rc2^((2-ma)/ma)-((ma+1)/ma)*rc2^(1/ma));
y1=f/(G);  rc2=rc2-(f/(G));
if abs(y1)<q1,break,end;
Pcout=rc2*Pcin;  Tcout=Tcin*rc2^(ma1);
zetaC=(1-rc2^((ma1)*(zet_ic)))/(1-rc2^(ma1));
delhI=cp*(Tcin)*(rc2^(Rair/cp)-1);
Pc=wa*delhI/(zetaC*Zetrans); end;  end;
```

Appendix A.3

****

Combustor Model

```
%S-Function   **Simulink Block**
function [sys,x0]=alicombs(t,x,u,flag)
if flag==3;
wa=u(1);  Tcout=u(2);  Pcout=u(3);  Wis=u(4);  Tis=u(5);  wf=u(6);
[Ttin,Ptin,Wg,htin,gcnox,gcco]=combustr(wa,Tcout,Pcout,Wis,Tis,wf);
sys=[Ttin,Ptin,Wg,htin,gcnox,gcco];
elseif flag==0;
%Lab to return no. of continuous states;% dab No. of discrete states;
% mab No. of outputs; % nab No. of inputs; % drab No. of discontinuous roots;
%To find algebriac loop sizes(6);
Lab=0;  dab=0;  mab=6;  nab=6;  drab=0;
sys=[0,0,mab,nab,0,any(Pcout~=0)];
else;  sys=[];  end ;
% m.file    **This subroutine simulates the Combustion Chamber**
function[Ttin,Ptin,Wg,htin,gcnox,gcco]=combustr(wa,Tcout,Pcout,Wis,Tis,wf)
%parameters
Cpg=1147 [J/(kg °K)];  k1=1;  Am=1 [m^2];  delh25=-4.0*10^7 [J/kg]; Rcg=287
[J/(kg °K)];  href=1.2041*10^6 [J/kg];  Tref=1000 [°K];
Cpa=1005 [J/(kg °K)];  Cps=2160 [J/(kg °K)];  k2=1;
%-----
Wg=(wa+wf+Wis);
Ttin=(1/((Wg)*Cpg))*(-wf*delh25-wa*Cpa*(298-Tcout)-Wis*Cps*(298-
Tis))+298;
delp=Pcout*(k1+k2*((Ttin/Tcout)-1))*(Rcg/2)*Tcout*(Wg/(Am*Pcout))^2 ;
Ptin=Pcout-delp;
%gcnox=fgt2*(Wis/(wf));
gcnox=23.333*(Wis/(wf))^2-94.333*(Wis/(wf))+100.;
%gcco=fgt3*(Wis/(wf));
gcco=38.1959*(Wis/(wf))^2-25.4874*(Wis/(wf))+8;
htin=href+Cpg*((Ttin-Tref));  end;
```

Appendix A.4

****

Turbine model

```
%S-Function   **Simulink Block**
function [sys,x0]=aliturbn(t,x,u,flag)
if flag==3;
Pc=u(1);  Ttin=u(2);  Ptin=u(3);  Wg=u(4);  htin=u(5);
[Pmech,Ttout,Ptout,htout]=turbine(Pc,Ttin,Ptin,Wg,htin);
sys=[Pmech,Ttout,Ptout,htout];elseif flag==0;
%Lat to return no. of continuous states;% dat No. of discrete states;
% mat No. of outputs; % nat No. of inputs; % drat No. of discontinuous roots;
%To find algebriac loop sizes(6);
Lat=0;  dat=0;  mat=4;  nat=5;  drat=0;
sys=[Lat,dat,mat,nat,drat,any(Ptout~=0)];
else;  sys=[];  end ;
% m.file     **This subroutine simulates the Turbine**
function [Pmech,Ttout,Ptout,htout]=turbine(Pc,Ttin,Ptin,Wg,htin);
%parameters;
zet_it=.9;  Aot=.14 [m^2];  Cpg=1147 [J/(kg °K)];
gam_cg=1.333;  Rcg=2871147 [J/(kg °K)];
%Algebraic equations;
mcg=gam_cg/(gam_cg-(gam_cg-1)/zet_it);
rho_tin=Ptin/(Rcg*Ttin);
% Solve using the secant method
rt=.01; rt1=.5; Qa=1.e-6;
%------------------------ -------------------------------
ft=(((Wg/Aot)^2)*((mcg-1)/(2*zet_it*mcg*rho_tin*Ptin))-
rt^(2/mcg)+rt^((mcg+1)/mcg));
Gft=(((Wg/Aot)^2)*((mcg-1)/(2*zet_it*mcg*rho_tin*Ptin))-
rt1^(2/mcg)+rt1^((mcg+1)/mcg));
for j=1:50;
zf=rt1-(rt-rt1)*Gft/((ft-Gft));
rt=rt1; rt1=zf; ft=Gft;
Gft=((Wg/Aot)^2)*((mcg-1)/(2*zet_it*mcg*rho_tin*Ptin))-
rt1^(2/mcg)+rt1^((mcg+1)/mcg);
```

```
if abs(ft-Gft) == 0,break,end; if abs(rt1-rt) < Qa,break,end;
Ptout=rt*Ptin;  Ttout=Ttin*(rt)^(zet_it*((gam_cg-1)/gam_cg));
delhI=Cpg*Ttin*((rt)^(Rcg/Cpg)-1);
zet_t=(1-(rt)^((gam_cg-1)*zet_it/gam_cg))/(1-(rt)^((gam_cg-1)/gam_cg));
Pt=zet_t*Wg*abs(delhI);  htout=htin+zet_t*delhI;  Pmech=Pt-Pc;  end;  end
```

Appendix3 A.5

```
*************************************************
```

Closed-loop gas turbine linear model,

```
Ash=1.0e+2*[-0.00172324657585 -0.00030654641863  0.00000000000000...
             0.00000000000287 -0.00000000000141  0.13021299119328
            -0.00030654641863 -0.00005453120180  0.00000000000000...
            -0.00000000000007  0.00000000000003 -0.00278023918671
             0.00000000000000  0.00000000000000  0.00000000000000...
            -0.00000182368432  0.00032986705929  0.02614266521006
             0.00000000000000  0.00000217996696  0.06540095427907...
            -0.00477278138866  0.00621033885460 -0.01313639966487
             0.00000000000000  0.00000670020923 -1.00106044303307...
             0.00622132693911 -0.09959426990136 -0.00085863131982
             0.00000000000000  0.00782087911916  0.00083938677440...
            -0.00003187319762 -0.00000207864263 -0.10048729978217];
Bsh=[ -0.00000000103276  0.00000000000000 -0.00000043481993
       0.00000000580564  0.00000000000000  0.00000000928405
       0.00000000000000 -0.00019448799240  0.00000010954020
       0.00000000000003 -0.00128410328354  0.00893731206807
       0.00000000000010  0.01965514136742  0.00058388705031
       0.00000000011351 -0.00001648078878 -0.00000298973994];
Csh=1.0e+008*[0.00000000000000  0.00000000000000  0.00000000000000...
              0.00000000000013 -0.00000000005264  4.33163232202350
              0.00000000000000  0.00000000000000  0.00000000000000...
             -0.00000000937685  0.00000169607931  0.00013441788878
              0.00000000000000  0.00000000000000  0.00000000000000...
              0.00000048411533  0.00000003285142  0.00000145898664];
Dsh=[0.00000000000000  0.00000000000117 -14.46443805915778
     0.00000000000000  0.00000000000000  0.00056321858105
     0.00000000000000  0.00000000000000  0.00484740536946];
```

Appendix3 B.1

GPC linear model simulation, a MATLAB program

```
Ts=.3
[a,b,c,d]=c2dm(Ash,Bsh,Csh,Dsh,Ts);AA=a;Bm=b;CC=c;Dm=d;
%----------------------------------------------------------------
%GPC final tuning parameters---------------------------------------------
%N1=5;N2=19;Nu=11;tkf=250;
%gama1=0.0000000007;delta1=0.00000007;
%gama2=.25;delta2=19;
%gama3=5;delta3=.3;
N1=3; N2=30; Nu=5;tkf=250;
%initial control paameters
gama1= 0.00000002947999;delta1=0.0000008;
gama2= 320;delta2=800;
gama3= 850;delta3=230;
%----------------------------------------------------------------
% delta=lambda
ya1=[];wyy=[];wyy1=[];u00=[];ux11=[];ux12=[];ux13=[];
yx11=[];yx12=[];yx13=[];wy=[];kk=[]
%-------
[nAA mAA] =size(AA);[nBm mBm]=size(Bm);[nCC mCC]=size(CC);
[nDm mDm]=size(Dm);delmu=zeros(mBm,1);delmu1=zeros(mBm,1);
uo=zeros(3,1)
%-----***************Kalman**********---------------------------------
xhh=zeros(nAA+mBm,1);yhc1=[];
%-------reference trajectory ----------------------------------------
for j=1:1:150;    w1(j)=3.391e7; end;
for j=150:1:250;  w1(j)=3.95e7;end;
for iu=1:1:250;   w2(iu)=101.5;end;
for iuj=1:1:250;  w3(iuj)=90;end;
%------------Multivariable SSGPC controler matrices---------------------
[F,GGGHm]=aagpcm6(AA,nAA,Bm,mBm,CC,nCC,Dm,N1,N2,Nu,gama1,gama2,
gama3,delta1,delta2,delta3);
%-----------**************Kalman filter  parameters***************--
q=[1 0.0001 0.000001 0.01 0.01 0.000001 100000 40 25]*2;
```

```
Q=diag(q);R=diag([.05 1400 .5]*1500);
Gx1=eye(9)*.0001;[nGx mGx]=size(Gx1);
%------------Process and output disturbances----------------------
nZZz=mGx-1;zz=randn([1,1]);Zz=[zz zeros(1,nZZz)]';vk=randn(nCC,1);
%-----------The augmented linear model-----------------------------
AA1=[AA  Bm;zeros(mBm,nAA)  eye(mBm)];
Bm1=[zeros(nAA,mBm);eye(mBm)];
CC1=[CC Dm];Dm1=zeros(mBm,mBm);Dm1=zeros(mBm,mBm);
%--------------------Kalman filter gain----M-----------------------
[M,l,p,E]=dlqe(AA1,Gx1,CC1,Q,R);x1=zeros(9,1);
%---------------compute the random disturbances z(k) and v(k)---------
for dft=1:tkf;zz=randn([1,1]);Zz=[zz zeros(1,nZZz)]';vk=randn(nCC,1);
%-------------------------Process model--Augmented model--------------
x1new=AA1*x1+Bm1*delmu+Gx1*Zz*.002;
ya=CC1*x1+Dm1*delmu+vk;
x1=x1new;
ya1=[ya1 ya];
%-----------------------------------------------------------------
%x1new=AA*x1+Bm*uo+G*Zz*.002;
%ya=CC*x1+Dm*uo+vk;%x1=x1new;
%ya1=[ya1 ya];
%------------**********Kalman states estimator***********-------------
xhhn=AA1*xhh+Bm1*delmu+M*(ya-CC1*xhh);
%--------------------Compute mvarssgps control law-------------------------
%legth of w=length(nCC*(N2-N1)
nw=length(N1:N2);
w11=w1(dft)*ones(nw,1);w1a=w11;
w22=w2(dft)*ones(nw,1);w2a=w22;
w33=w3(dft)*ones(nw,1);w3a=w33;for iwe=1:nw;w1a1=w11(iwe);
w2a1=w22(iwe);
w3a1=w33(iwe);wx=[w1a1 w2a1 w3a1];wy=[wy wx];end;wy1=wy';
GGGHm1=GGGHm(1,:);
GGGHm2=GGGHm(2,:);
GGGHm3=GGGHm(3,:);
Kgpc=[GGGHm1;GGGHm2;GGGHm3];
```

```
delmux1=Kgpc*(wy1-F*xhh);
delmu=delmux1;xhh=xhhn;wy=[];wyy=[];
uo=uo+delmu;u00=[u00 delmu];end;
ux11=[ux11 u00(1,:)];ux12=[ux12 u00(2,:)];ux13=[ux13 u00(3,:)];
yx11=[yx11 ya1(1,:)];yx12=[yx12 ya1(2,:)];yx13=[yx13 ya1(3,:)];
%-------------------------------------------------------------
%now display all the results obtained
figure
%----------------------------------
subplot(311);plot(yx11,'b'); hold on;stairs(w1(1:dft));
%----------------------------------
subplot(312);plot(yx12,'b'); hold on;stairs(w2(1:dft)) ;
%----------------------------------
subplot(313);plot(yx13,'b'); hold on;stairs(w3(1:dft));
%----------------------------------
```

Appendix3 B.2

A MATLAB program for GPC Closed loop analysis

```
Ash;Bsh;Csh;Dsh;Ts;
[a,b,c,d]=c2dm(Ash,Bsh,Csh,Dsh,Ts); AA=a; Bm=b; CC=c; Dm=d;
%-------------------------------------
%tuning parameters used for comparison % delta=lambda
%N1=5;N2=22;Nu=13;
%gama1=0.0000000007;delta1=0.00000001;
%gama2=.55;delta2=19;
%gama3=5;delta3=.3;
%Tuning parameter of final controller design
N1=5;N2=19;Nu=11;
gama1=0.0000000007;delta1=0.00000007;
gama2=.25;delta2=19;
gama3=5;delta3=.3;
% initial controller tuning patameters
%N1=3; N2=30; Nu=5;tkf=250;
%gama1= 0.00000002947999;delta1=0.0000008;
%gama2= 320;delta2=800;
```

```
%gama3= 850;delta3=230;
%--------------------------------
[nAA mAA] =size(AA);[nBm mBm]=size(Bm);
[nCC mCC]=size(CC);[nDm mDm]=size(Dm);
%------------------the augmented model---------------------------
AA1=[AA  Bm;zeros(mBm,nAA)  eye(mBm)];
Bm1=[zeros(nAA,mBm);eye(mBm)];
CC1=[CC Dm];Dm1=zeros(mBm,mBm);
Dm1=zeros(mBm,mBm);
%-----------------Kalman filter parameters-------------------------
Gx1=eye(9)*.0001
q=[1 0.0001 0.000001 0.01 0.01 0.000001 100000 40 25]*2;
Q=diag(q);R=diag([0.05 1400 0.5]*1500);
%-----------------Multivariable SSGPC control matrices----------------------
[F,GGGHm]=aagpcm6(AA,nAA,Bm,mBm,CC,nCC,Dm,N1,N2,Nu,gama1,gama2,
gama3,delta1,delta2,delta3);
%-----------**************Kalman filter gain***************------------
[M,l,p,E]=dlqe(AA1,Gx1,CC1,Q,R);
%----------------------------------------------------------------------
GGGHm1=GGGHm(1,:);GGGHm2=GGGHm(2,:);
GGGHm3=GGGHm(3,:);Kgpc=[GGGHm1;GGGHm2;GGGHm3];
%--------------------------------------
w=logspace(-8,1,198);
%Process model
Go=pck(AA1,Bm1,CC1,Dm1);Af=AA1-M*CC1;
[nAA1,mAA1]=size(AA1);I=diag(ones(1,nAA1));
[nz1,mz1]=size(Bm1);[nz2,mz2]=size(I);
df=zeros(nz2,mz1);
%Kfr--------------------
Kfr=pck(Af,Bm1,I,df);
%Kfz--F1--------------------
Kfz=pck(Af,M,I,df);
%F--and--F1----------------
F1=pck([],[],[],F);
%-F1 and Kfr in series results in kk------
```

```
F_Kfr=mmult(F1,Kfr);kk=F_Kfr;
%------------------kt-------------------------------
%--find kt by kt=(inv(I+Kgpc*F*Kfr))Kgpc*F*Kfr with kk1=Kgpc*KK
Kgpc_KK=mmult(Kgpc,kk);
kk1=Kgpc_KK;[mat_1,row_1,col_1,num1]=minfo(kk1);
I1=eye(row_1);kk2=minv(madd(I1,kk1));
%Kgpc--and--Kgpc1-----------------------------
Kgpc1=pck([],[],[],Kgpc);
%----Kt -------and its state space form---------
Kt=mmult(kk2,Kgpc1);[ak,bk,ck,dk]=unpck(Kt);
%------------put in series Go and Kt-------and find state space form--
Go_Kt=mmult(Go,Kt);kk3=Go_Kt;
[ap,bp,cp,dp]=unpck(kk3);
%------kk4---put F and Kfz in series-------------------------------
F_Kfz=mmult(F1,Kfz);kk4=F_Kfz;
[akk4,bkk4,ckk4,dkk4]=unpck(kk4);
%------kk3=Go*Kt and kk4=F*Kfz-----------------------------------
%---sensitivity function  S=kk6=inv(I+Go*kt*F*Kfz)---------------
kk4_kk3=mmult(kk3,kk4);kk5=kk4_kk3;
[mat_2,row_2,col_2,num_2]=minfo(kk5);
I2=eye(row_2);
%*************************************************************
%--sensitivity function-----------***********-------------------
kk6=minv(madd(I2,kk5));[a2,b2,c2,d2]=unpck(kk6);
[sv1 w]=dsigma(a2,b2,c2,d2,Ts,w);
figure;grid;semilogx(w,20*log10(sv1(1,:)),'b');
grid;xlabel('frequency (rad/s)')
ylabel(' max sv (dB)')title('sensitivity')
%*************************************************************
%---------------------------------***********-----------------
%---complementary sensitivity function T=kk7=S*kk5 kk5=kk4*kk3-----
kk7=mmult(kk6,kk5);[a1,b1,c1,d1]=unpck(kk7);
[sv w]=dsigma(a1,b1,c1,d1,Ts,w);
figure;grid;semilogx(w,20*log10(sv(1,:)),'b');grid;xlabel('frequency (rad/s)');
ylabel('singular values (dB)'); title('complementary sensitivity')
```

```
%*************************************************************
%----control sensitivity  Cs=kk8*kt-------*********------
[mat_o,row_o,col_0,num0]=minfo(kk1);
kk8=minv(madd(eye(row_o),mmult(Kt,kk4,Go)));
kk9=mmult(kk8,Kt);[a3,b3,c3,d3]=unpck(kk9);
[sv3 w]=dsigma(a3,b3,c3,d3,Ts,w);
figure;semilogx(w,20*log10(sv3(1,:)),'b');
grid;xlabel('frequency (rad/s)');
ylabel(' max. sv (dB)');title('control sensitivity');
%*************************************************************
%---additive perturbation-------------------*********------
yy=minv(madd(eye(row_o),mmult(Go,Kt,kk4)));yyz=mmult(Kt,kk4);
[ay,by,cy,dy]=unpck(mmult(yy,yyz));[svy w]=dsigma(ay,by,cy,dy,Ts,w);
figure;grid;semilogx(w,20*log10(ones(1,length(svy))./svy(1,:)),'b');
grid;xlabel('frequency (rad/s)');ylabel(' max. sv (dB)');
title('additive perturbation');
%---------------------------------------**********-----------
%--------multiplicative input perturbation---------------------
yy1=mmult(minv(madd(eye(row_o),mmult(Kt,kk4,Go))),mmult(Kt,kk4,Go));
[ay1,by1,cy1,dy1]=unpck(yy1);[svy1 w]=dsigma(ay1,by1,cy1,dy1,Ts,w);
figure;grid;semilogx(w,20*log10(ones(1,length(svy1))./svy1(1,:)),'b')
grid;xlabel('frequency (rad/s)');title('multiplicative input perturbation')
ylabel(' max. sv (dB)')
%-------------------------------------------*******---------
%-----------multiplicative output perturbation-----------------
yy2=mmult(minv(madd(eye(row_o),mmult(Go,Kt,kk4))),mmult(Go,Kt,kk4));
[ay2,by2,cy2,dy2]=unpck(yy2);[svy2 w]=dsigma(ay2,by2,cy2,dy2,Ts,w);
figure;grid;semilogx(w,20*log10(ones(1,length(svy2))./svy2(1,:)),'b');
grid;xlabel('frequency (rad/s)');ylabel(' max. sv (dB)');
title('output multiplicative perturbation')
%---------------------------------------************-------
% sensitivity and complementary sensitivity comparison
figure;grid;semilogx(w,20*log10(sv1(1,:)),'b');
hold;semilogx(w,20*log10(sv(1,:)),'b-');
grid;xlabel('frequency (rad/s)');ylabel(' max sv (dB)')
```

```
title('singular values of s and cs')
ylabel('The maximum singular values of Ts and Ss (dB)');
%-------------------------------------------*******----------
```

Appendix B.3

******************Simulink model*************

Standard GPC

Mutivariable State-Space GPC Algorithm******************************

```
%                    **GPC control law S-function**
function [sys,x0,str,ts]=algk6m18(t,x,u,flag,A;B;C;D,Ts);
global F GGGHm
if flag==3;
%Ts sampling time; A,B;C;D are the model matrices
[ac,bc,cc,dc]=c2dm(A,B,C,D,Ts);
AA=ac;Bm=bc;CC=cc; Dm=dc;
%ti is a clock; xh1 to xh9 are the estimated states;uo1 to uo3 are the control
%signals;w1 to w3 are the set-points
ti=u(1);xh1=u(2);xh2=u(3);xh3=u(4);xh4=u(5);
xh5=u(6);xh6=u(7);xh7=u(8);xh8=u(9);xh9=u(10);
uo1=u(11);uo2=u(12);uo3=u(13);w1=u(14);w2=u(15);w3=u(16);
uo=[uo1;uo2;uo3];xhh=[xh1;xh2;xh3;xh4;xh5;xh6;xh7;xh8;xh9];
if ti==1;xhh=zeros(9,1);%intial states
uo=zeros(3,1);%initial control signals
end;
[nBm mBm]=size(Bm);
%--------------------Controller tuning parameters-------------------
N1=5;N2=19;Nu=11;
gama1=0.0000000007;delta1=0.00000007;
gama2=.25;delta2=19;
gama3=5;delta3=.3;
%---------------------
u00=[];wy=[]; ya1=[];
[nAA mAA] =size(AA);[nBm mBm]=size(Bm);[nGx mGx]=size(Gx);
```

[nCC mCC]=size(CC);[nDm mDm]=size(Dm);
%-------**GPC switching time----**
if ti<=80;w1=0;w2=0;w3=0;end;
%-**Go to subroutine aagpcm6 and compute GPC matrices F and GGGHM--**
if
ti==1;[F,GGGHm]=**aagpcm6**(AA,nAA,Bm,mBm,CC,nCC,Dm,N1,N2,Nu,gama1,gama2,gama3,delta1,delta2,delta3);end;
%---
%-------------------**construct w(k+j)--j=N1:N2 at each sample time-------------**
%legth of w=length(nCC*(N2-N1))-
nw=length(N1:N2);w11=w1*ones(nw,1);
w1a=w11;w22=w2*ones(nw,1);w2a=w22;w33=w3*ones(nw,1);
w3a=w33;size(GGGHm);size(F);for iwe=1:nw;w1a1=w11(iwe);
w2a1=w22(iwe);w3a1=w33(iwe);wx=[w1a1 w2a1 w3a1];wy=[wy wx];
size(wy);end;wy1=wy';
% --------- **Compute the multivariable SSGPC control law----------------------**
delmux=GGGHm*(wy1-F*xhh);
%**select the first control increment** $\Delta_p(k)$
for id=1:3;delmu(id,:)=delmux(id,:);end;
% **construct the control signal u(t)**
uo=uo+delmu
del1=delmu(1,1);del2=delmu(2,1);del3=delmu(3,1);
uo4=uo(1,1);uo5=uo(2,1);uo6=uo(3,1);
% **clear the future set-point vector for the next sample time calculations**
wy=[];
%**del1 to del3 are thecontrol increments and uo4 to uo6 are the control**
%**signals**
sys=[del1,del2,del3,uo4,uo5,uo6];
elseif flag==0;
Lab1=0;dab1=0;mab1=6;nab1=16;drab1=0;
sys=[0,6,mab,nab,0,0];else;sys=[];end;end;
*GPC matrices calculator***
function[F,GGGHm]=**aagpcm6**(AA,nAA,Bm,mBm,CC,nCC,Dm,N1,N2,Nu,gama1,gama2,gama3,delta1,delta2,delta3)

```
r1=nCC;nv=nAA;p1=mBm;[mBm nBm]=size(Bm);
eye(nBm);zeros(mBm,nAA);
%           Construct the augmented model
AA2=[AA  Bm;zeros(nBm,nAA) eye(nBm)];
Bm2=[zeros(nAA,nBm);eye(nBm)];
CC2=[CC Dm];Dm2=zeros(nBm,nBm);
 [nc mc]=size(CC2);[nb1 mB1]=size(Bm2);
p1=mB1;r1=nc;

%--------------------------Compute F matrix--------------
for ip=N1:N2;cat=CC2*AA2^(ip);
for jp=1:r1;for k=1:mc;
frell(jp+(ip-N1)*r1,k)=cat(jp,k);end;end;end;
%----------------------------------
for i1=N1:N2;xc1= (i1-N1)*r1;
j1=1;ki=i1-j1;
%------------------------------Compute Hm matrix--------------------------
for ix=ki:-1:0;cat11=CC2*(AA2^(i1-j1))*Bm2;
for ii=1:r1;for jj=1:p1;frew11(ii+xc1,jj+(j1-1)*p1)=cat11(ii,jj);
end;end;
j1=j1+1;
if j1-Nu> 0,break,end ; end; while i1-j1== -1 ;
if j1-Nu> 0,break,end ;
%----------------------------------------
cat11=Dm2;for ii=1:r1; for jj=1:p1;
frew11(ii+xc1,jj+(j1-1)*p1)=cat11(ii,jj);end; end;
j1=j1+1;if j1-Nu> 0,break,end ;end; end
%------------------------------------------
F=frell;Hm=frew11;xx1=size(Hm,1);
xx=xx1/3;yy1=size(Hm,2);yy=yy1/3;
%---------------------------------------------
R11=gama1;R2=gama2;R3=gama3; R=[R11 R2 R3];gam1=[];
for ih=1:xx; gam1=[gam1 R]; end;gamax=diag(gam1);
%---------------------------- Compute GGGHm matrix --------------------
Q1=delta1;Q2=delta2;Q3=delta3; q=[Q1 Q2 Q3];delt=[];
```

```
for iH=1:yy; delt=[delt q]; end;deltax=diag(delt);
GG1=Hm'*gamax*Hm+deltax;GG=inv(Hm'*gamax*Hm+deltax);
gamHm=gamax*Hm;GHm=Hm'*gamax;
GGGHm=GG*GHm;GGGHm;end;
```

Kalm gain calculator and state estimation update ************************

```
function [sys,x0,str,ts]=algk6m17(t,x,u,flag, A;B;C;D,G,Q,R,Ts);
% G is a diagonal matrix
global M
if flag==3;[ac,bc,cc,dc]=c2dm(A,B,C,D,Ts);
AA=ac;Bm=bc;CC=cc;Dm=dc;ti=u(1);
%-------------------------process outputs vector --------------------
P=u(2);T=u(3);N=u(4);yaa=[P;T;N] ;
%                    control signals and control incements
uo1=u(5);uo2=u(6);uo3=u(7);
delmu1=u(8);delmu2=u(9);delmu3=u(10);
xh1=u(8);xh2=u(9);xh3=u(10);
xh4=u(11);xh5=u(12);xh6=u(13);xh7=u(17);xh8=u(18);xh9=u(19);
uo=[uo1;uo2;uo3];delmu=[delmu1;delmu2;delmu3];
xhh=[xh1;xh2;xh3;xh4;xh5;xh6;xh7;xh8;xh9];
if ti==1;xhh=zeros(9,1);uo=zeros(3,1);end;delmu=zeros(3,1);
[nBm mBm]=size(Bm);
%----------------------------------------
[nAA mAA] =size(AA);[nBm mBm]=size(Bm);
[nCC mCC]=size(CC);[nDm mDm]=size(Dm);
dimnR=size(nCC);[nR mR]=size(nCC);
%                    Augmented Model
AA1=[AA  Bm;zeros(mBm,nAA) eye(mBm)];
Bm1=[zeros(nAA,mBm);eye(mBm)];
CC1=[CC  Dm];Dm1=zeros(mBm,mBm);yhc1=[];

%---------------------Kalman filter gain-----------------------------
if ti==1;[M,l,p,E]=dlqe(AA1,G,CC1,Q,R);end
%-------------------Kalman filter esrimator----------------------------
M;CC1;xhh;xhhn=AA1*xhh1+Bm1*delmu+M*(yaa-CC1*xhh1);
%----------------------------------------------------------------------
```

```
xh1=xhhn(1,1);xh2=xhhn(2,1);xh3=xhhn(3,1);xh4=xhhn(4,1);
xh5=xhhn(5,1);xh6=xhhn(6,1);xh7=xhhn(7,1);xh8= xhhn(8,1);xh9= xhhn(9,1);
sys=[xh1,xh2,xh3,xh4,xh5,xh6,xh7,xh8,xh9];
elseif flag==0;
Lab=0; dab=0; mab=9; nab=19; drab=0;
sys=[0,0,mab,nab,0,0];else; sys=[]; end; end;
```

Appendix B.4

****

Constrained GPC

Constrained Multivariable State-Space GPC Algorithm

```
function [sys,x0,str,ts]=acgk6m18(t,x,u,flag);
global F GG1 Hm gamHm GGGHm
if flag==3;
A;B;C;D;
 [ac,bc,cc,dc]=c2dm(A,B,C,D,.3);AA=ac;eig(AA);Bm=bc;
CC=cc;Dm=dc;ti=u(1);xh1=u(2);xh2=u(3);xh3=u(4);xh4=u(5);
xh5=u(6);xh6=u(7);xh7=u(8);xh8=u(9);xh9=u(10);uo1=u(11);
uo2=u(12);uo3=u(13);w1=u(14);w2=u(15);w3=u(16);
uo=[uo1;uo2;uo3];xhh=[xh1;xh2;xh3;xh4;xh5;xh6;xh7;xh8;xh9];
if ti==1;xhh=zeros(9,1);uo=zeros(3,1);end;[nBm mBm]=size(Bm);
%--------------------
% GPC tuning parameters
%%%%%%%%%%%%%%STANDAR GPC
N1=5;N2=19;Nu=11
gama1=0.0000000007;delta1=0.00000007;
gama2=.25;delta2=19;
gama3=5;delta3=.3;
%%%%CONSTRAINED GPC%%%%%%%21/4/97
%rate 1 NU
Nldu=1;Nudu=11;
%Amplitude 1 Nu
Nlu=1;Nuu=11;
%overshoot N1 N2
Nlo=5;Nuo=19;
```

```
%output N1 N2
Nly=5;Nuy=19;
%-----------------------
ymxy=[];ymny=[];ymmx=[];ymmpu=[];ymmpL=[];wmmnx=[];
Umaa=[];u00=[];wy=[];wyy=[];wyy1=[];
[nAA mAA] =size(AA);[nBm mBm]=size(Bm);
[nGx mGx]=size(Gx);[nCC mCC]=size(CC);[nDm mDm]=size(Dm);
%-------GPC switching time ----
if ti<=80;w1=0;w2=0;w3=0;end;ya1=[];
%------------Multivariable SSGPC control law parameters-----------------
if
ti==1;[F,GG1,Hm,gamHm,GGGHm]=ccgpcm7(AA,nAA,Bm,mBm,CC,nCC,Dm,
N1,N2,Nu,gama1,gama2,gama3,delta1,delta2,delta3);end;
%--------------------Compute mvarssgpc control law------------------------
nw=length(N1:N2);w11=w1*ones(nw,1);
w1a=w11;w22=w2*ones(nw,1);w2a=w22;
w33=w3*ones(nw,1);w3a=w33;size(GGGHm);size(F);
for iwe=1:nw;w1a1=w11(iwe);w2a1=w22(iwe);w3a1=w33(iwe);
wx=[w1a1 w2a1 w3a1];wy=[wy wx];size(wy);end;wy1=wy';
%constained problem solution
% min 0.5*x'Hx +f'x   subject to:  Ax <= b;
%x;%[x,LAMBDA]=QP(H,f,A,bc);
%X=QP(H,f,A,b,VLB,VUB);
%*****************
f=-[gamHm'*(wy1-F*xhh)];H=GG1;
[nH,mH]=size(H);vc=mH/3;
%Rate constraint   ok
uuu=length(Nldu:Nudu);
for iy1=1:uuu;rcu1=3.4e5/50;rcu2=101.5/50;rcu3=9/4.5;
%for iy1=1:uuu;rcu1=3.393e7;rcu2=1002;rcu3=90.1;
wxx1=[rcu1 rcu2 rcu3];wyy1=[wyy1 wxx1];end;
duLvrr=-wyy1';duUvrr=wyy1';I=[eye(3*vc)];[Ir Ic]=size(I);
for iiI=1:3*uuu;for iiII=1:Ic;dumxx(iiI,iiII)=I(iiI,iiII);
end;end;dumxr=dumxx;dumnr=-dumxx;
[duLvr duLvc]=size(duLvrr);[acor acoc]=size(dumxr);
```

```
%Amplitude constraint      ok
uuuLc=length(Nlu:Nuu);
for iy=1:uuuLc;acu1=3.396e7/1;acu2=1002;acu3=90.2;
wxx=[acu1 acu2 acu3];wyy=[wyy wxx];end;
VLB1=-wyy';VUB1=wyy';[vub1r vub1c]=size(VUB1);
for iau=1:uuuLc;Umaa=[Umaa uo'];end;
U0=Umaa';[U0r U0c]=size(U0);ak=mBm;
aCa=diag(ones(1,ak));aDa=zeros(ak,ak);[ar1 ap1]=size(aCa);
for ai1=1:Nu;axc1= (ai1-1)*ar1;aj1=1;aki=ai1-aj1;
%----------------------------------------L matrix----------
for aix=aki:-1:0;acat11=aCa;for aii=1:ar1;for ajj=1:ap1;
afrew11(aii+axc1,ajj+(aj1-1)*ap1)=acat11(aii,ajj);
end;end;aj1=aj1+1;if  aj1-Nu> 0,break,end; end;
while ai1-aj1== -1 ;if  aj1-Nu> 0,break,end ;
%----------------------------------------
acat11=aDa;for aii=1:ar1;for ajj=1:ap1;
afrew11(aii+axc1,ajj+(aj1-1)*ap1)=acat11(aii,ajj);
end;end;aj1=aj1+1;if  aj1-Nu> 0,break,end ;
end;end;
%if  aj1-Nu>= 0,break,end ;
AKc=afrew11;LAkc=afrew11;[LAKr LAKc]=size(LAkc);
for iiK=1:3*uuuLc;for iiKK=1:LAKc;
LAc(iiK,iiKK)=LAkc(iiK,iiKK);end;end;
LAc=LAc;size(LAc);
%-----output constraint--  ok
%ymx1=3.397e5;ymx2=110.8;ymx3=9.3;ymn1=-3397;ymn2=-120;ymn3=-3.3;
ymx1=3.4e5;ymx2=102;ymx3=9;ymn1=-3397;ymn2=-120;ymn3=-3.3;
%------constraint on the output---
nwot=length(Nly:Nuy);nyot=length(Nly:Nuy);
ymx11=ymx1*ones(nyot,1);ymx1a=ymx11;ymx22=ymx2*ones(nwot,1);ymx2a=ym
x22;ymx33=ymx3*ones(nwot,1);ymx3a=ymx33;
%---------------------------
for iwe=1:nwot;
ymx1a1=ymx11(iwe);ymx2a1=ymx22(iwe);
ymx3a1=ymx33(iwe);ymxx=[ymx1a1 ymx2a1 ymx3a1];
```

```
ymxy=[ymxy ymxx];end;ymxy1=ymxy';
[ymxy1r ymxy1c]=size(ymxy1);[Frot Fcot]=size(F);
[hrot hcot]=size(Hm);for iifrot=1:ymxy1r;for iifcot=1:Fcot;
Fot(iifrot,iifcot)=F(iifrot,iifcot);end;end;
Fot=Fot;Ymax=(ymxy1-Fot*xhh);
%Ymno=(ymin-Fot*xhh);
for iihot=1:ymxy1r;for iigot=1:hcot;
Hmot(iihot,iigot)=Hm(iihot,iigot);end;end;Hmot=Hmot;size(Hmot);
%----minimum output constraint--
ymn11=ymn1*ones(nyot,1);ymn1a=ymn11;ymn22=ymn2*ones(nwot,1);ymxna=ym
n22;ymn33=ymn3*ones(nwot,1);ymn3a=ymn33;
%----------------------------
for iwe=1:nwot;
ymn1a1=ymn11(iwe);ymn2a1=ymn22(iwe);ymn3a1=ymn33(iwe);
ymnx=[ymn1a1 ymn2a1 ymn3a1];ymny=[ymny ymnx];size(ymny);end;
ymny1=ymny';Ymin=(ymny1-Fot*xhh);
%---overshoot constraint------ok
for ww=Nlo:Nuo;wmx1=3.397e7;wmx2=1010.52;wmx3=80.05;
%wmnx1=-10000000;wmnx2=-5000;wmnx3=-100.5;
wmxx=[wmx1 wmx2 wmx3];wmmx=[wmmx wmxx];
end;wmax=wmmx';[wmaxr wmaxc]=size(wmax);
[Fr Fc]=size(F);[hr hc]=size(Hm);
for iifr=1:wmaxr;for iifc=1:Fc;Fos(iifr,iifc)=F(iifr,iifc);end;end;
Fos=Fos;Ymxo=(wmax-Fos*xhh);Ymno=-Ymxo;
for iih=1:wmaxr;foriig=1:hc;Hmos(iih,iig)=Hm(iih,iig);end;end;

Hmos=Hmos;Ac1=[dumxr;-dumnr;LAc;-LAc;Hmot;-Hmot;Hmos;-Hmos];
bc=[duUvrr;-duLvrr;VUB1-U0;-VLB1+U0;Ymax;-Ymin;Ymxo;-Ymno];
%[delmux1,lambda,how]=qp(H,f,[],[],VLB,VUB);
size(VUB1);
% QP solution
[delmux1,lambda,how]=qp(H,f,Ac1,bc);
%delmux=GGGHm*(wy1-F*xhh);
for id=1:3;delmu(id,:)=delmux1(id,:);end;uo=uo+delmu
del1=delmu(1,1);del2=delmu(2,1);del3=delmu(3,1);
```

```
uo4=uo(1,1);uo5=uo(2,1);uo6=uo(3,1);
wy=[];wwy=[];wyy1=[];Ac=[];
sys=[del1,del2,del3,uo4,uo5,uo6];
elseif flag==0;Lab=0;dab=0;mab=6;
nab=16;drab=0;sys=[0,6,mab,nab,0,0];else;sys=[];end;end;
function
[F,GG1,Hm,gamHm,GGGHm]=ccgpcm7(AA,nAA,Bm,mBm,CC,nCC,Dm,N1,N2,
Nu,gama1,gama2,gama3,delta1,delta2,delta3)
n11=N1;n12=N2;nu3=Nu;r1=nCC;p1=mBm;
[mBm nBm]=size(Bm);eye(nBm);zeros(mBm,nAA);
AA2=[AA  Bm;zeros(nBm,nAA) eye(nBm)];
Bm2=[zeros(nAA,nBm);eye(nBm)];CC2=[CC Dm];
Dm2=zeros(nBm,nBm);
 [nc mc]=size(CC2);[nb1 mB1]=size(Bm2);p1=mB1;r1=nc;
%----------------------------------F
for ip=N1:N2;cat=CC2*AA2^(ip);for jp=1:r1;for k=1:mc;
frell(jp+(ip-N1)*r1,k)=cat(jp,k);end;end;end;
%----------------------------------
for i1=N1:N2;xc1= (i1-N1)*r1;j1=1;ki=i1-j1;
%------------------------------------------Hm------
for ix=ki:-1:0;
cat11=CC2*(AA2^(i1-j1))*Bm2;
for ii=1:r1;for jj=1:p1;
frew11(ii+xc1,jj+(j1-1)*p1)=cat11(ii,jj);
end;end;j1=j1+1;
if j1-Nu> 0,break,end ; end;
while i1-j1== -1 ;if j1-Nu> 0,break,end ;
%-----------------------------------------
cat11=Dm2;for ii=1:r1;for jj=1:p1;
frew11(ii+xc1,jj+(j1-1)*p1)=cat11(ii,jj); end; end;
j1=j1+1;if j1-Nu> 0,break,end ;end;end
%--------------
F=frell;Hm=frew11;F;
xx1=size(Hm,1);xx=xx1/3;
yy1=size(Hm,2);yy=yy1/3;
```

```
%-----------------------------------------------
R11=gama1;R2=gama2;R3=gama3; R=[R11 R2 R3];gam1=[];
for ih=1:xx;gam1=[gam1 R];end;gamax=diag(gam1);gamax;
%-----------------------------------------------
Q1=delta1;Q2=delta2;Q3=delta3; q=[Q1 Q2 Q3];delt=[];
for iH=1:yy;delt=[delt q];end;deltax=diag(delt);
GG1=Hm'*gamax*Hm+deltax;
GG=inv(Hm'*gamax*Hm+deltax);
gamHm=gamax*Hm;  GHm=Hm'*gamax;
GGGHm=GG*GHm;  GGGHm;  end;
```

Appendix C.1

------------m-sequece PRBS signals generation

```
uu1=[];uu13=[];us1=[];us13=[];uu2=[];us2=[];uufl=[];usfl=[];xxx=[];xxx3=[];prbb
=[];prbb3=[];st=.5
%st is the sampling time; repeat each bit K1 times;PRBS signal length=2^n1-1
% (startnum1)  start at the third Bit
n1=3;K1=70;bitno1=[];startnum1=3;
%The generation of antisymetric PRBS signal using inverse repeat sequence.
%------------Maximum length Binary sequence
[x1,nex1]=mlbs(n1,bitno1,startnum1);N=length(x1);x1=x1;
%-----------construct the first PRBS signal
%repeat x1 that is signal period=2*N
xfl=[x1;x1];
% totat signal period=2*N*K1 repeat each bit K1 times (to allow reaching %steady
state)
for ifl=1:length(xfl);for k1=1:K1;uufl=[uufl xfl(ifl,1)];end;
usfl=[usfl uufl];uufl=[];endprbffl=[usfl usfl]';prbfl1=prbffl;
TTffl=[1:length(prbffl)]';
%-----------------------------------------------
prb1=[x1;x1];NB=(length(prbffl)/(length(usfl)/2));TT1=[1:length(prb1)]';
xx=[-1 1];for ii1=1:length(prb1)/2;xxx=[xxx  xx];end;xxx=xxx';
% construct the second uncorrelated PRBS signal
prb11=prb1+xxx;for jj=1:length(TT1);if prb11((jj)) == 0;prb11((jj))=1;
else;prb11((jj))=-1;end;prbb=[prbb  prb11((jj))];end;prbb=prbb';
```

```
% total sigal time in sec
tt=0:st:(st*N*K1*NB)-st; kk=1/(N*K1*NB)
tt=tt';for i1=1:length(prbb);for k1=1:K1;uu1=[uu1 prbb(i1,1)];end;
us1=[us1 uu1]; uu1=[];end;prbf=[us1 us1]';
prbf22=prbf;TTf1=[1:length(prbf)]';
%-------Construct the third uncorrelated PRBS signal--
prb3=[x1;x1;x1;x1];xx3=[-1 -1 1 1];for ii1=1:length(prb3)/4;
xxx3=[xxx3 xx3];end;xxx3=xxx3';prb113=prb3+xxx3;pp3=prb113;
TTf31=[1:length(prb113)]';for jj3=1:length(TTf31);if prb113((jj3)) == 0;
prb113((jj3))=1;else;prb113((jj3))=-1;end;prbb3=[prbb3  prb113((jj3))];
end;prbb3=prbb3';for i13=1:length(prbb3); for k11=1:K1;
uu13=[uu13 prbb3(i13,1)]; end; us13=[us13 uu13]; uu13=[];end;
prbf3=[us13]';  prbf33=prbf3;TTf1=[1:length(prbf)]';
%------ Subtract the mean values
prbf11=prbf11-mean(prbf11);prbf22=prbf22-mean(prbf22);
prbf33=prbf33-mean(prbf33);a_a=3.4e6;  b_b=9;  c_c=101.5;
%-------- Remove steady state off-set errors
prbf11a_a=max(abs(a_a*prbf11))-a_a;prbf22b_b=max(abs(b_b*prbf22))-b_b;
prbf33c_c=max(abs(c_c*prbf33))-c_c;prbfa_a=a_a*prbf11+prbf11a_a;
prbfb_b=b_b*prbf22+prbf22b_b;prbfc_c=c_c*prbf33+prbf33c_c;
%----------Plot the uncorrelated PRBS signals
figure;plot(tt,prbfa_a,'b');figure;plot(tt,prbfb_b,'b');
figure;plot(tt,prbfc_c,'b');
```

Appendix C.2

```
%---------S-Function which returns the identifeid, Subspace-based model
%The Subspace function which is based on the algotithm presented in %Chapter 5
is called subid. This functions and others are available in matlable toolbox
%developed by De Moor and Van Overschee.
function [sys,x0,str,ts]=subsii(t,x,u,flag);
if flag==3;
u1=u(1);u2=u(2);u3=u(3);y1=u(4);y2=u(5);y3=u(6);
ti=u(7);
%remember the sytem matrices A1, B1, C1, D!
```

```
global A1 B1 C1 D1
% remeber the past inputs and outputs to construct the input and output data
%vectors
global uuxa yyxa;
% initially let the vector containts zero, required for repeated experiments
if ti==0; uuxa=[];yyxa=[]; end
% construct input and output data vectors with 880 data samples
if ti>=1;
uo=[u1;u2;u3];yo=[y1;y2;y3];yyxa=[yyxa yo];uuxa=[uuxa uo];
% the condition to satisfy the requierd number of data samples
[z m]=size(yyxa);if m ==880;s=880;
%now perform subspace identification
u=uuxa(:,1:s)';y=yyxa(:,1:s)';[A1,B1,C1,D1,K1] =subid(y,u,3,6,[],[]);
% return u1,u2,u3 included just make S-function works
sys=[u1,u2,u3];end;end;
% Now return the system matricesA1;B1;C1;D1 and Kalman gain K1in the
%matlab command
A1;B1;C1;D1;K1;elseif flag==0;
Lab=0;dab=0;mab=3;nab=7;drab=0;
sys=[0,6,mab,nab,0,0];else;sys=[];end;end
```

Appendix C.3

```
%Adaptive predictive controller
function [sys,x0,str,ts]=algk6m18(t,x,u,flag);
global F01 GGGHm01  F1 GGGHm1 F2 GGGHm2 F3 GGGHm3 F4 GGGHm4
F5 GGGHm5 F6 GGGHm6
if flag==3;
A_01=[0.94574577653994  -0.24027800305579  0.03972216226354
       -0.02281617314499  0.45346116017397  -0.03016537852163;
        0.03003874310475  0.90082209589777  -0.37218494539215
        0.14105746290328  -2.80228403778543  0.18641418959130;
        0.00320396386888  -0.01464590736354  0.61546285491626
       -0.62514613686980  12.43939348101715  -0.82733437253491;
        0.00000768409291  -0.00160187412171  0.00419735084792
```

0.90175423433782 -0.51355995881888 0.32381782306926;
-0.00000332108843 -0.00003066115351 -0.00036898411647
0.00251249337720 0.93301468616398 0.02176236150529;
0.00000152703344 -0.00012108538383 -0.00244996933967
-0.01943492006732 -0.08758913533780 0.85064833460883];

B_01 =[0.00002515324359 0.32473551836850 -2.81417453249721
-0.00003295885934 0.47593773865529 -16.84337516420245
-0.00000661292766 0.50432531842300 -21.31633300442425
-0.00000024718674 0.01302194779815 0.42225103097788
-0.00000000868565 0.00048219707602 0.01403363782072
-0.00000005440656 0.00512084012921 -0.02884168459048]

C_01=1.0e+003*[1.07268989849930 0.00285385080866 0.00154087045115
-0.00000979162177 0.0000000993934 -0.0000093613828;
-0.00000084930322 -0.0000989150087 0.00013989637837
0.0047701710731 0.0003465147869 -0.00136898049886;
0.00000017986773 -0.0000003056784 -0.00000221038289
-0.00000013093109 0.00141529053296 0.0002784320358]

D_01 =[0.04529259500681 0 0
0.00000131177093 0.22579487298950 0.31815506514996
0.00000002385369 -0.00331088548575 0.17521539910778];

A_1=[0.91314110014693 -0.29648946311766 0.04390765670027
0.06264022425500 0.77882124104635 0.06677148513050;
0.04600179894280 0.95745039904718 -0.37548438953524
-0.33754318628111 -4.19768363931919 -0.35988475062395;
0.00164229482509 -0.02565833912987 0.49910735693566
1.36371331483664 16.9474570030773 1.45286509420488;
-0.00031460421450 0.00213366579434 0.02122991502090
0.85708992447106 -0.71973333790550 -0.30396745640288;
-0.00000502588795 -0.00011267269508 -0.00272931504008
0.01176098546948 0.97307424597343 -0.00122238138344;
-0.00002044984251 0.00028773606112 0.00141103146949

0.00220745487552 -0.18606420903129 0.84652693517458];

B_1=[-0.00003049700890 0.36557549072100 -7.59013748334172
0.00006844400654 0.65758724549325 -46.14112419708054
0.00004175988938 0.54128660823168 -67.19189034944421
-0.00000266096868 -0.01561518033486 3.47220926533416
0.00000024175035 0.00344118498595 -0.40850730724745
-0.00000018576031 -0.00031116593224 0.19988113846944];

C_1=1.0e+003*[-1.08102304469777 -0.30310823979638 -0.02539722802578
0.00001372672025-0.00000015619583 0.00000011314565;
0.00000300891728 0.00007053117487 0.00018876514200
0.00460798773195 0.00053373392532 0.00115966057408;
-0.00000018387448 0.00000157784113 -0.00000776415495
0.00005983996715 0.00124102592874 -0.00043103800727]

D_1 =[0.09267865041341 0 0
0.00000272730728 0.21962670813517 -0.49366008188047
0.00000002475617 -0.00183576810363 0.21077045526851];

A_2=[0.89578475543913 -0.32985050923127 -0.03765238057117
0.00220605633380 0.09190186356913 0.12959323522209;
0.04454507928598 0.97829402104196 0.27041056797451
-0.01075808994940 -0.46042236856855 -0.64925365246687;
-0.00084529705488 -0.00120990345993 0.85464701870049
-0.08128345800023 -3.17032291029212 -4.46866173505662;
0.00022146416624 -0.00075919965866 -0.00374455832848
0.92849914351895 0.01074532925747 -0.47831319037426;
-0.00005540555982 -0.00003099338563 0.00121825047808
-0.01868494403672 0.87471352883862 -0.01738101831824;
0.00003582136051 -0.00004796231382 0.00137147541399
0.00571215491961 0.02351104822322 0.92338930298027];

B_2=[0.00006814478548 -0.05665584726619 -6.69078713924056
-0.00005965309691 0.22932078653848 -8.22942476145287

0.00000139594212 -0.12152620842584 7.36194639324935
-0.00000071554945 0.00276031156975 0.43724990677330
0.00000006504490 0.00341723542513 0.05748769091155
-0.00000008988082 -0.00010200065799 -0.04274774273818];

C_2=1.0e+002*[8.50506123552767 2.57001275850377 -0.12669706925809
-0.00045490701615 -0.00000163254196 0.00000259444076;
-0.00002680220753 -0.00081412592123 -0.00355852629573
0.04123316081623 -0.01248493064843 0.00285496452306;
-0.00000171046420 -0.00002742179385 0.00000786355190
0.00055653930804 0.00286493978294 0.01040812547251];

D_2 =[0.06739785728420 0 0
0.00000206487409 0.17068428675339 -0.17152434861186
0.00000006923504 -0.00095632391556 0.35673611082552];

A_3 =[0.89442612378919 -0.33141910774754 -0.05644377041552
0.00683424616747 0.19212111916867 0.79543193357696;
0.04200579682101 0.97145615614152 0.37194829295441
-0.03389144240393 -0.95590478128814 -3.95769022611835;
-0.00496116938522 0.01433508620828 1.30700662161016
-0.21331321804385 -5.95573496724178 -24.65941233965870;
-0.00006469464038 0.00116731354500 0.00144398376115
0.94867064643651 -0.32259550265200 -0.28567866271235;
-0.00008695102855 -0.00006963066471 0.00055009587366
0.01540191448845 0.86554126607190 -0.07687350154464;
-0.00000056628140 0.00018337715154 0.00688277230136
-0.00757852761594 -0.06346875006740 0.57255090895182];

B_3 =[-0.00010262510487 0.12174848274004 28.58529590292834
0.00008963388226 -0.54664418114680 29.68938515117146
-0.00000164323388 0.32856869239896 -10.49535134465505
-0.00000061112679 0.01785024675424 -0.70667904262023
-0.00000019481560 -0.00061790436836 -0.09933445305903
0.00000025810578 0.00354756647404 -0.02045058589899];

```
C_3=1.0e+002*[-7.60260964792414 -2.31305234015937 0.12599448298581
              0.00015675784126 0.00000136398006 0.00000315042230;
              0.00006962973289 0.00062998235006 -0.00083275705924
              0.0518837111712 0.01279870383044 -0.00322556612975;
              0.00000276499427 0.00003742703089 0.0001170456869
              0.00065037153749 -0.0035164124507 -0.0094926058569];

D_3=[0.10664697068196 0 0
     0.00000313680336 0.13059924576944 -0.05541119633431
     0.00000016309172 -0.00224308888695 0.38470967614246];
%-------pidsid44
A_4=[0.89138907082938 -0.31715923439370 0.04479872687220
     0.00461176613777 0.20033380225537 0.13185819341564;
     0.04439959335404 0.96452419017479 -0.34068853730048
     -0.02378019823000 -1.04098921481905 -0.68516998121703;
     0.00182803955403 -0.00080406445626 0.79241787413569
     0.12953474179784 5.51407666521890 3.62887856801311;
     0.00002469588506 -0.00089721312670 0.00309828418457
     0.95717524100778 -0.03132020324060 -0.26144670114616;
     -0.00003040212232 -0.00012445065634 -0.00105271198204
     -0.00982641833813 0.88612459785084 -0.00361797721852;
     0.00006125745641 -0.00008755269630 -0.00196522808627
     0.00559919925788 -0.00609796934944 0.87837565111989];

B_4=[0.00008460555703 0.01999081627388 -09.42905379646954
     -0.00008131805460 0.23512312560550 -18.16001471544498
     -0.00000568885288 0.16708593371098 -15.18124490577212
     -0.00000039816268 0.00104558141117 00.31723797872346
     0.00000001124554 0.00210647538456 00.01201178323374
     -0.00000023857906 -0.00135967301594 -0.06937172384312];

C_4=1.0e+002 *[6.62188447140704 1.92960073275637 0.12370171999135
              -0.00021012617257 -0.00000020648512 0.00000045490852;
              -0.00002054575690 -0.00047281412188 0.00156857342491
```

```
0.04254208868182  -0.00820595583188  0.00518974809326;
-0.00000365964123 -0.00003352326976  0.00002455165236
0.00058095299639  0.00529673521498   0.00675326070108]

D_4=[0.11463516342806  0                  0
     0.00000275317094  0.09951009283401  -0.14419714730373
     0.00000023142220 -0.00080066464757   0.41853674543554];
%-----------pidsid55
A_5=[00.88530256901348  -00.31951477042073  -0.03046027191358
     00.12994358168601   00.59803589515181  00.01762742234988
     00.04437842063604   00.97669186087945  00.28307551895184
    -00.68025663783266  -03.13121938331732 -00.09229176334079
    -00.00418065995376   00.00364863392604  00.30260324770888
    -03.15150989128992  -14.49772390989676 -00.42712038341238
     00.00041872938528  -00.00103103528779 -00.03737186734413
     00.67121956793527  -01.24544985273694 -00.25980327265340
     00.00001492472344  -00.00005659779433  00.00250998342947
    -00.00087562340042   00.88745923092787  00.02108653578396
    -00.00001290252613  -00.00003200281078 -00.00801793064630
    -00.02806933122762  -00.21409378692769  00.89699400074994]

B_5=1.0e+002*[0.00000097221410  0.00000000000000 -0.17708685651489
             -0.00000093070100  0.00000000000000 -1.91005273653379
              0.00000018919847  0.00000000000000  7.21866205201430
             -0.00000001333657  0.00000000000000  0.49605009683960
             -0.00000000210364 -0.00007247918288 -0.02298631480705
              0.00000000178943 -0.00020393638279  0.08720102124763]

C_5=1.0e+002*[6.23241690872057  1.80359162895710 -0.13244069980771
             -0.00011400979014  0.00000046518289 -0.00000093522602
             -0.00002326371529 -0.00083510932465 -0.00190456200880
              0.02319520910074 -0.00041639075970  0.00520941932382
             -0.00000490845075 -0.00004339431494 -0.00003225067051
              0.00058703117675  0.00757986489558  0.00015930044787]
```

```
D_5=[0.13417690041473  0                 0
     0.00000399353747  0                 0.01074783450829
     0.00000036431155  0.04444368465508  0.48477482750737];
ti=u(1);
xh1=u(2);xh2=u(3);xh3=u(4);xh4=u(5);xh5=u(6);xh6=u(7);
xh7=u(8);xh8=u(9);xh9=u(10);uo1=u(11);uo2=u(12);uo3=u(13);
w1=u(14);w2=u(15);w3=u(16);uo=[uo1;uo2;uo3];
xhh=[xh1;xh2;xh3;xh4;xh5;xh6;xh7;xh8;xh9];
uo11=uo1;if ti==1;xhh=zeros(9,1);uo=zeros(3,1);end;
%new values
N1=5;N2=19;Nu=11;gama1=0.0000000007;delta1=0.00000001;
gama2=.25;delta2=19;gama3=5;delta3=.3; ya1=[];u00=[]; wy=[];
AA=A_3;Bm=B_3;CC=C_3;Dm=D_3;
 [nAA mAA] =size(AA);[nBm mBm]=size(Bm);
[nGx mGx]=size(Gx);[nCC mCC]=size(CC);
[nDm mDm]=size(Dm);if ti<=30;w1=0;w2=0;w3=0;end;
%--------------------------------------------------------------------
%------------Multivariable SSGPC control law parameters-----------------
if     ti==1;AA=A_01;Bm=B_01;CC=C_01;Dm=D_01;[F01,GGGHm01=aagpcm6
(AA,nAA,Bm,mBm,CC,nCC,Dm,N1,N2,Nu,gama1,gama2,gama3,delta1,delta2,del
ta3);end;
if ti==1;AA=A_1;Bm=B_1;CC=C_1;Dm=D_1;[F1,GGGHm1]=aagpcm6
(AA,nAA,Bm,mBm,CC,nCC,Dm,N1,N2,Nu,gama1,gama2,gama3,delta1,delta2,del
ta3);end;
if ti==1;AA=A_2;Bm=B_2;CC=C_2;Dm=D_2;[F2,GGGHm2]=aagpcm6
(AA,nAA,Bm,mBm,CC,nCC,Dm,N1,N2,Nu,gama1,gama2,gama3,delta1,delta2,del
ta3);end;
if ti==1;AA=A_3;Bm=B_3;CC=C_3;Dm=D_3;[F3,GGGHm3]=aagpcm6
(AA,nAA,Bm,mBm,CC,nCC,Dm,N1,N2,Nu,gama1,gama2,gama3,delta1,delta2,del
ta3);end;
if ti==1;AA=A_4;Bm=B_4;CC=C_4;Dm=D_4;[F4,GGGHm4]=aagpcm6
(AA,nAA,Bm,mBm,CC,nCC,Dm,N1,N2,Nu,gama1,gama2,gama3,delta1,delta2,del
ta3);end;
if ti==1;AA=A_5;Bm=B_5;CC=C_5;Dm=D_5;[F5,GGGHm5]=aagpcm6
```

```
(AA,nAA,Bm,mBm,CC,nCC,Dm,N1,N2,Nu,gama1,gama2,gama3,delta1,delta2,del
ta3);end;
if ti==1;AA=A_6;Bm=B_6;CC=C_6;Dm=D_6;[F6,GGGHm6]=aagpcm6
(AA,nAA,Bm,mBm,CC,nCC,Dm,N1,N2,Nu,gama1,gama2,gama3,delta1,delta2,del
ta3);end;
%------------------------------------------------------------------------
uo11=3.3988e7+uo11;
if uo11>=3.3988e7;F=F01;GGGHm=GGGHm01;end;
if uo11>=3.3988e7-1*3.4e6;F=F1;GGGHm=GGGHm1;end;
if uo11>=3.3988e7-2*3.4e6;F=F2;GGGHm=GGGHm2;end;
if uo11>=3.3988e7-3*3.4e6;F=F3;GGGHm=GGGHm3;end;
if uo11>=3.3988e7-3*3.4e6-2.35e6;F=F4;GGGHm=GGGHm4;end;
if uo11>=3.3988e7-3*3.4e6-2.35e6-2.135e6;F=F5;GGGHm=GGGHm5;end;
if uo11<3.3988e7-3*3.4e6-2.35e6-2.135e6;F=F5;GGGHm=GGGHm5;end;
uo11=uo11-3.3988e7;
%--------------------Compute mvarssgpc control law------------------------
%legth of w=length(nCC*(N2-N1))--for each process output we need a setpoint
nw=length(N1:N2);
w11=w1*ones(nw,1);w1a=w11;w22=w2*ones(nw,1);w2a=w22;
w33=w3*ones(nw,1);w3a=w33;size(GGGHm);size(F);
for iwe=1:nw;w1a1=w11(iwe);w2a1=w22(iwe);w3a1=w33(iwe);
wx=[w1a1 w2a1 w3a1];wy=[wy wx];size(wy);end;wy1=wy';
delmux=GGGHm*(wy1-F*xhh);for id=1:3;delmu(id,:)=delmux(id,:);end;
uo=uo+delmu;del1=delmu(1,1);del2=delmu(2,1);del3=delmu(3,1);
uo4=uo(1,1);uo5=uo(2,1);uo6=uo(3,1);wy=[];
sys=[del1,del2,del3,uo4,uo5,uo6];elseif flag==0;Lab=0;dab=0;mab=6;
nab=16;drab=0;sys=[0,6,mab,nab,0,0];else;sys=[];end;end;
```