University of Strathclyde
Industrial Control Centre
Department of Electronic and electrical Engineering

Distributed Nonlinear State-Dependent Model Predictive Control and Estimation for Power Generation Plants

By

Salah G. Abokhatwa

A thesis presented in fulfilment of the requirements for the degree of
Doctor of Philosophy

2014

Declaration

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

Acknowledgements

It is a pleasure to thank the many people who made this thesis possible

Foremost, I would like to express my sincere gratitude to my supervisor Dr. Reza Katebi for his continuous expert guidance, support and patience. This thesis would not have been possible without his helpful comments, advice, stimulating discussions and encouragement.

I would like to extend my sincere gratitude to the Libyan Ministry of Higher Education and Scientific Research for the financial support.

I would also like to thank my friends and colleagues in the Industrial Control Centre for their friendship and motivation.

My deepest and sincere thanks are always to my mother and all my family for their unconditional love and support.

Finally, but definitely not least, I would like to express deep appreciation to my wife Soad, without her understanding, endless patience and encouragement it would have been impossible for me to finish this work, and to my daughters Samah , Saja and Haneen.

Dedication

This thesis is dedicated to my mother, who sadly passed away four days before my Viva exam, and to my father who passed away in 1999. May Allah forgive their sins and grant them Paradise.

This is also dedicated to my wife Soad and my daughters Samah, Saja and Haneen.

Abstract

Centralized model predictive control (MPC) is often considered impractical, inflexible and unsuitable for controlling large-scale systems due to several factors such as large computational effort and difficulty to meet all operational objectives. Therefore, industrial large-scale systems are usually controlled by a distributed control framework. In this thesis, novel sequential nonlinear Distributed Model Predictive Control (DMPC) algorithms for large-scale systems that can handle constraints are proposed. The proposed algorithms are based on nonlinear MPC strategy, which uses a state-dependent nonlinear model to reduce the complexity of solving optimization problem. In this distributed framework, the overall system is divided into several interconnected subsystems and each subsystem is controlled by local MPC. These local MPCs solve convex optimization problem and exchange information via one directional communication channel at each sampling time to achieve the global performance. The proposed algorithms are applied to an industrial power plant model to improve power generation efficiency. A non-linear dynamic model of Combined Cycle Power Plant (CCPP) using the laws of physics was first developed and simulated using decentralized PID controllers. Then, a supervisory controller using linear constrained MPC was designed to tune the performance of the PID controllers. Next, a supervisory centralized nonlinear model predictive control (NMPC) algorithm based on state-dependent models was developed to control the nonlinear plant over a wide operating range. Finally, two sequential DMPC algorithms based on state-dependent models were developed. The lack of states measurement were handled by designing nonlinear distributed state estimation algorithms using state-dependent differential Riccati equation (SDDRE) Kalman filter. Numerical simulation results show that the performance of the proposed DMPC algorithms is close to the centralized NMPC but computationally more efficient compared to the centralized one.

Contents

List of Figures

List of Tables

List of Abbreviations

BLT	Biggest Log Modulus
CCPP	Combined cycle power plants
CCGT	Combined cycle gas turbine
CHP	Combined heat and power
DCS	Distributed control system
DMPC	Distributed model predictive control
EKF	Extended Kalman filtering
G_0	Steady-state gain matrix
HRSG	Heat recovery steam generator
IMC	Internal Model Control
LMI	Linear matrix inequalities
LMPC	Linear model predictive control
LPV	Linear parameter varying
LQR	Linear quadratic programming
LTV	Linear time-varying
MIMO	Multi input/multi output
MPC	Model Predictive Control
NMPC	Nonlinear model predictive control
PID	Proportional integral derivative
PI	Proportional integral
QP	Quadratic programming
RGA	Relative gain array
RHP	Right half plane
SDARE	State-dependent algebraic Riccati equations
SDC	State-dependent coefficient
SDDRE	State-dependent differential Riccati equation
SDRE	State-dependent Riccati equation
SISO	Single input/single output

List of Symbols

A	Amplitude of oscillation of relay feedback
A_d	Area of drum
A_{dc}	Area of downcomer
A,B,C,D	State space constant matrices
$A_k(x), B_k(x), C_k(x)$	State-dependent discrete matrices
C_p	Specific heat of the metal
C^1	First order differentiability class
Di	Damping coefficient of the generator
E	Disturbance matrix
F_D	Fuel demand signal from the gas turbine controller
H_i	Inertia constant of the generator
K	Gain of the first order system
K_c	PID proportional gain
K_i	PID integral gain
K_k	Kalman filter gain
K_d	PID differential gain
K_{cu}	Ultimate gain
L	Drum level
N_p	Prediction horizon
N_c	Control horizon
P	Drum pressure
P_u	Oscillation period
P_G	Boiler furnace gas pressure
P_{mG}	Power of gas turbine
P_{ms}	Power of steam turbine

$P_{k\|k-1}$	A priori estimate error covariance
P_s	Superheated steam pressure
$P(x)$	Solution of the state-dependent Algebraic Riccati Equation
Q	Weighting on the tracking error
Q_p	Heat flow rate to the risers
Q_w	Process noise covariance
R	Reference signal
R_ω	Measurement noise covariance
$\Re^n, \Re^{n\times n}$	Real numbers with *n* dimension for vectors and *nxn* dimension for matrices
S	Weighting on the control increments
S_ψ	Weight for constraints violations
T	Discretization time
T_d	Residence time
T_{Gout}	Gas turbine exhaust temperature
T_{rh}	Temperature of Reheater metal tubes
T_s	Superheated steam temperature
U	System input
U_k	Optimal control input trajectory
V_{st}	Total steam volume
V_{wt}	Total water volume
V_r	Riser volume
V_t	Total volume of the drum, downcomers, and risers
V_r	Riser volume
V_{sd}^0	Volume of steam in the drum when there is no condensation
V_{sd}	volume of steam under the liquid level in the drum
V_{wd}	Volume of water in the drum
V_{dc}	Downcomer volume

X	System states
$X(k)$	State prediction vector
Y	System Output
$d(k)$	Measured disturbance
g_c	IMC controller
$\tilde{g}$	Model of the system
gNOx	Nitrogen oxides level in gas turbine
h	Height of the relay input
h_c	Condensation enthalpy
h_G	Enthalpy of exhaust gas from the gas turbine
k	Discrete-time sequence
k_{fr}	Friction coefficient
m_t	Total metal mass
m_r	Riser metal mass
m_d	Drum mass
m_dl	Total drum liquid mass
q_{dc}	Total mass flow rate into the risers
q_f	Feedwater mass flow rate
q_s	Steam mass flow rate
t	time
t_m	Metal temperature
$u_i(k)$	Subsystem *i* input vector
$v_i(k)$	Subsystem *i* output interactions vector
w_{fG}	Fuel flow to the gas turbine combustion chamber
w_{is}	Injected steam flow into the gas turbine combustion chamber
w_f	Boiler fuel flow
w_e	Feedwater flow
w_a	Attemperator water flow
w_A	Air flow to the furnace

w_{in}	Steam flow from boiler to steam turbine
w_G	Exhaust gas flow from the gas turbine
$x_i(k)$	Subsystem *i* state vector
x_c	State of PI controllers
x_0, u_0, y_0	Nominal operating point values
$\hat{y}$	Future Output prediction
y_{max}	Upper output constraints
y_{min}	Lower constraints
$y_i(k)$	Subsystem *i* output vector
ω_g	Frequency of the generator
α_r	Steam quality
α_1, α_2	Real numbers for parameterization flexibility
α_i	Filter parameter to cancel the unstable poles
θ	Time delay
λ	IMC low pass filter time constant
ξ_k	Process noise
η_k	Measurement noise
τ	Time constant
τ_I	Integral time
ψ_k	Maximum constraints violation
ω_0	Generator synchronous frequency
$\omega_i(k)$	Subsystem *i* state interactions vector
S_i	Subsystem *i*

1. Introduction

1.1 Background

During the last few decades, the ever-growing demand for electric power, deregulation of power industry and its associated competition and more strict environmental legislation have given rise to an increasing interest in Combined Cycle Power Plants (CCPP) due to their high efficiencies and their low emission. CCPP or Combined Cycle Gas Turbine (CCGT) is a power plant system in which two types of turbines, namely a gas turbine and a steam turbine, are combined in one cycle to generate electricity. The purpose of introducing combined cycle in power plants is to reduce energy loss. Their main role is to use the waste heat of the gas turbine exhaust gases to produce additional electricity.

The power plant is a highly complex, nonlinear, and time varying system, where its control is of a multi-loop nature with interactions between different loops. The most common control strategy used is decentralized multi-loop controller to stabilize the outputs. The well-known Proportional Integral Derivative (PID) regulatory controllers have been employed in each loop due to their simplicity and ease of tuning. However, to improve the economic operation, an advanced control strategy is needed. Model based Predictive Control (MPC) has received wide acceptance in process industries because of its ability to handle constraints and its optimization based formulation (Qin and Badgwell, 2003).

In power plant applications MPC is usually implemented using a supervisory control strategy. In this strategy, the MPC provides optimum set-points to the regulatory level, based on a dynamic optimisation procedure. The basic concept of MPC method is to use a model of the system to predict the future control inputs and outputs behaviour of the process over a prediction horizon. Then, future control inputs are optimized such that the predicted response of the system has some desirable features. Therefore, to apply MPC it is required to develop a moderately

complex non-linear model that can capture the key dynamical properties over a wide range of operating conditions. There are three types of models that have been used for calculating the predicted outputs in MPC implementations:

1- First principles models (white-box) obtained from the analysis of physical systems at the fundamental level. These use scientific principles such as, Newton's law, laws of thermodynamics and mass and energy balances to derive an analytical model. The advantage of first–principle models is that they are globally valid. Therefore, these models are expected to provide better extrapolation accuracy than empirical models, and can be used to predict the process dynamic over a wide range of operations (Åström and Bell, 2000).

2- Input-output models (black-box) obtained from the observed behaviour of the physical system (data-driven). A model structure is first selected. Then, the model parameters are identified based on measurements of input and output signals from the true system using system identification. The resulting model is called an empirical model or black–box model. Empirical models are generally easier to obtain than first-principles models because of their generally lower dimensionality but they often provide inaccurate predictions for operating conditions that are outside the range used for model identification (Qin and Badgwell, 2003).

3- Hybrid first principles empirical models (gray-box) are developed by combining the first–principle knowledge with empirical modelling approaches. This is the case when some physical insights are available, but several parameters remain to be determined from observed data. This allows the advantages of each modelling approach to be exploited (Åström and Bell, 2000).

MPC methods may be divided into two main categories: linear model predictive control (LMPC) and nonlinear model predictive control (NMPC) techniques. LMPC refers to a family of MPC schemes in which linear or linearized models are used to predict the system dynamics. LMPC is acceptable when the process operates at a single operating point and the controller is used only for disturbances rejection.

The power plant process dynamic often exhibits severely nonlinear behaviour. In addition, the demands for rapid changes in power generation require frequent changes from one operating point to another and often near the boundary of admissible region. This suggests the need for NMPC strategies. NMPC techniques involve solving nonlinear differential equations and a nonlinear dynamic optimization problem online. This computational effort is one of the main obstacles to the adoption of non-linear predictive controller in a wider context. In addition, using a nonlinear model may change the control problem from a convex optimisation problem to a non-convex non-linear program, for which global optimum solution cannot be guaranteed. This has motivated the study of alternative MPC approaches, requiring the solution of simpler optimization problems in real-time. Most of these approaches are based on linear time-varying (LTV) prediction through local Jacobian linearization (Kouvaritakis et al., 1999) or state dependent description of the nonlinear system (Cloutier, 1997). The state dependent representation of a system model avoids model linearization. Non-linearity is handled by the replacement of the original nonlinear system with a sequence of linear time-varying systems whose solutions will converge to the solution of the nonlinear problem.

The NMPC approach, based on the state-dependent model, has been demonstrated for many applications, such as for helicopter control application in (Dutka et al., 2003), for flight control in (Youssef et al., 2003) and for cruise control in (Shakouri and Ordys, 2011). Recently, Grimble (2013) pointed out the potential of using state-dependent modelling and control methods for industrial process applications. However, to the best of the author's knowledge, there have been no studies which use the state-dependent NMPC approach to control large-scale power plant systems.

For large-scale process, different control solutions can be developed using MPC, such as centralized, decentralized and distributed control techniques. The centralized MPC formulation is based on a centralized system model with a centralized control problem (single agent), where all the interactions are considered and all the control inputs are computed in one optimization problem, providing better performance. However, the use of centralized control strategy for a large scale system is often considered impractical due to the high on-line computation requirement, and

undesirable properties with respect to fault tolerance and reliability (Dunbar and Murray, 2004; Zheng et al., 2011). Therefore, distributed model predictive control (DMPC) architectures have received great attention, motivated by their advantage of providing similar performance to centralized MPC while maintaining flexibility against failures and partial shut-downs due to their decentralized nature.

DMPC control strategies for large-scale industrial process described in literature can be classified into two groups namely decentralized and distributed MPC. The distributed structure consists of decomposing the centralized MPC control problem into *n* subproblems, where each subproblem solves an MPC control problem for a particular subsystem. Then, the controllers are coordinated online among themselves via exchange of information in order to ensure feasibility. In this strategy, all distributed controllers can be evaluated in parallel as in Vaccarini et al. (2009) for example, or in sequential order as in Richards and How (2007), or based on neighbourhood optimization as in Zhang and Li, (2007). In the decentralized structure, there is no exchange of information between the control agents. A comprehensive literature review of decentralized and distributed control strategies can be found in survey papers (Rawlings and Stewart, 2008; Negenborn et al., 2009; Scattolini, 2009; Al-Gherwi et al., 2011; Christofides et al., 2012).

In literature, there are only a few papers that consider nonlinear DMPC. For example, Necoara et al. (2009) and Zheng et al. (2009) proposed a nonlinear DMPC for a nonlinear large scale process. In this method, the prediction model of each MPC is linearized using a successive on-line linearization method. Liu et al. (2009) and Chen et al. (2012) proposed nonlinear sequential DMPC methods for nonlinear systems. These strategies need a high computational effort in large-scale systems because they involve solving nonlinear differential equations and a nonlinear dynamic optimization problem online. Recently, nonlinear DMPC has attracted increasing attention for large-scale systems applications (Christofides et al., 2013).

Most of the DMPC schemes mentioned in the literature depend on the assumption of availability of the measurement of complete states (Christofides et al., 2013). However, in a large-scale control system it is possible that measurements of all states

are not available. Therefore, it is important to study distributed state estimation schemes for a large scale-system in order to integrate them with DMPC control algorithms to achieve the desired level of stability, performance and robustness. One approach is to design a centralized observer that sends the estimated state to all distributed controllers. However, for large scale systems, centralized implementation of a Kalman filter state estimator may be computationally expensive or even difficult to construct. Therefore, distributed estimation has attracted increasing attention with the development of large-scale sensor networks.

1.2 Research Objectives

Following the preceding discussion, a number of objectives for the design of a nonlinear DMPC strategy to control a large-scale CCPP system have been defined, as follows:

1- Development of a nonlinear DMPC based on a state-dependent model of an interconnected large-scale CCPP to provide a reduction in the complexity of the NMPC online optimization problem and improve performance through constrained optimisation.

2- Development of centralized NMPC based on a state-dependent model of CCPP.

3- Development of a distributed nonlinear observer based on the state-dependent Riccati equation (SDRE) Kalman filter.

4- Development of a non-linear CCPP MATLAB/SIMULINK model that is suitable for use in designing and testing model based controllers.

5- Applications of the above methods to a number of CCPP architecture to demonstrate improved performance and energy optimization.

1.3 Summary of Research Contributions

The original contributions of this thesis can be summarised as follows:

1- Novel sequential nonlinear constrained DMPC algorithms for large-scale systems are proposed. These algorithms are based on a nonlinear MPC strategy that uses a state-dependent nonlinear model to avoid the complexity of solving the non-convex nonlinear optimization problem. In the first algorithm, the interactions from all previous subsystems are considered as input disturbances, whereas in the second algorithm, interactions are modelled as functions of states and estimated from the previous time step. The proposed algorithms are applied to an industrial combined cycle power plant model. Comparison between the centralized NMPC and the DMPC schemes has been performed. Numerical simulation results show that the performance of the proposed DMPC algorithms is close to the centralized NMPC but computationally much more efficient.

2- Two sequential distributed nonlinear estimators are proposed to estimate all subsystem states. These estimation methods are based on the SDRE Kalman filter. In the first sequential estimation algorithm, each agent must include all previous subsystem models and it requires every subsystem node to send its measurements to all subsequent subsystem nodes in a one directional communication channel and once at each time step. However, in the second algorithm, each subsystem includes its model and all previous interactions, which are modelled as functions of previous states. Therefore, this sequential algorithm needs less computation effort and requires every subsystem node to send its state estimates and control input to all subsequent subsystem nodes.

3- A supervisory constrained centralized nonlinear model predictive (NMPC) control strategy based on the LTV state-dependent model is formulated and applied for the regulation of a nonlinear power plant model. The main difference with the previous work in this area is that the complete algorithm using an optimization procedure, a nonlinear estimator and constraints are

employed to design a 2nd level controller to generate optimal set points for 1st level regulating PID loops.

4- A nonlinear CCPP model is developed and simulated using a MATLAB\SIMULINK S-functions environment. This model is based on the principles of mass, energy and momentum conservations that can capture the key dynamical properties over a wide operating range. The contribution here is the modification of the boiler model described in (Ordys et al., 1994) to include shrink and swell dynamics. Another contribution is the formulation of all nonlinear power plant subsystem models into state-dependent coefficient (SDC) discrete models to be used in the NMPC control design.

5- The generalized Internal Model Control (IMC) and Relay feedback tuning methods are used to tune PID loops of power plant system. Simulation results show that the PID tuning rules based on IMC tuning method gives better closed loop performance than the Relay feedback method.

1.4 Outline of the Thesis

The structure of the thesis is described hereunder.

In **Chapter 2**, the modelling and simulation of CCPP is presented. The CCPP model is based on first principles of physics. Section 2.2 describes the configuration of the most popular types of CCPP. In section 2.3 the main subsystems of CCPP, such as a boiler, steam turbine and gas turbine are described. Dynamic models of the main components of the CCPP that have been developed and used in this thesis are described in section 2.4, which focuses on modelling of the boiler with a new model simulator being presented. This model can capture much of the system dynamics, such as the shrink and swell phenomenon. Section 2.5 describes the PID regulatory control of CCPP. It also describes the IMC and relay feedback tuning methods. Good model accuracy and control performance have been achieved, as confirmed by simulation. In section 2.6, the CCPP model is approximated by a simpler model in

order to simplify the derivation of the nonlinear state-dependent coefficient matrices that will be used in Chapter 4.

Chapter 3 presents a supervisory linear MPC controller to improve the control performance of CCPP by providing the optimal set-points for the PID regulatory level. Output constraints are also included in solving the optimization problem. A Kalman filter observer is used to estimate the system states. Section 3.2 describes the supervisory control strategy. Section 3.3 describes the MPC algorithm formulation based on a discrete-time linear state-space model. This section also describes the changes to the MPC formulation when adding measured input disturbances and output constraints. This linear MPC algorithm is used as a basis for the development of LTV state-dependent NMPC in Chapter 4. Section 3.4 presents the validation results of the linearized model. This is done by comparing the simulation of linearized and nonlinear models using the same frequency and step input signals. Simulation results of supervisory MPC are also presented in this section. Simulation results have shown that the supervisory MPC has better performance than classical PID control schemes and allows handling constraints.

Chapter 4 presents a supervisory constrained NMPC algorithm based on the LTV state-dependent model of the nonlinear CCPP described in Chapter 2. A nonlinear estimator based on the SDRE Kalman filter is used to estimate the unmeasured states. Section 4.2 describes the SDRE control method. Section 4.3 describes how to derive the parameterization of SDC representation. Section 4.4 presents the formulation of the NMPC algorithm. In section 4.5, nonlinear estimation methods are reviewed, including the formulations of state-dependent algebraic Riccati equations (SDARE) and state-dependent differential Riccati equation (SDDRE) filters. Section 4.6 presents application of the NMPC algorithm to a nonlinear boiler model and then to the nonlinear power plant model. In this section, the state-dependent coefficient (SDC) representations for both models are presented. It also presents the simulation results of NMPC algorithms that show a good tracking and disturbance rejection performance and the ability to handle constraints.

Chapter 5 presents new nonlinear sequential DMPC algorithms based on state-dependent NMPC discussed in the previous chapter. Section 5.1 describes a state of the art review of the different DMPC methods, such as decentralized MPC, communication-based (non-cooperative) DMPC, and cooperative-based DMPC. Section 5.2 describes the sequential DMPC architecture. Section 5.3 presents the proposed two nonlinear sequential DMPC algorithm formulations. These two algorithms require decomposition of the model of the entire system into N subsystems models and the solution of corresponding N convex-optimization problems. The difference between these algorithms is related to how the interactions between subsystems in the DMPC solution are considered. Section 5.4 describes the proposed distributed nonlinear state estimation algorithms which are based on an SDDRE Kalman filter. The distributed models of the power plant model formulated in SDC representation are described in section 5.5. Simulation results of both distributed nonlinear state estimation algorithms and DMPC controllers are presented in section 5.6, which is followed by the conclusion in section 5.7.

Chapter 6 presents a summary and conclusions of this thesis, along with a discussion of its limitations and suggestion of potential areas for further future work.

1.5 Publications arisen from this research

The following publications are directly related to the work reported within this thesis:

Abokhatwa, S. and R. Katebi (2012). "Modeling and Supervisory Control Design for a Combined Cycle Power Plant." Proceedings of the IASTED International Conference ,Control and Application, Crete, Greece, ACTA Press.

Abokhatwa, S. and R. Katebi (2012). "Nonlinear Predictive Control of an Industrial Power Plant Boiler". Control & Automation (MED), 2012 20th Mediterranean Conference , Barcelna,Spain, IEEE.

Abokhatwa, S. and R. Katebi (2013). "Sequential Distributed Model Predictive Control for State-Dependent Nonlinear Systems." IEEE international conference of systems, man and cybernetics, Manchester,UK.

Abokhatwa, S. and R. Katebi (2013). "Sequential Distributed Nonlinear State-Dependent Model Predictive Control and Estimation." Submitted for IET journal of control theory and applications.

Abokhatwa, S. and R. Katebi (2013). "Nonlinear State-Dependent Model Predictive Control and Estimation for Power Plant Process." Submitted for International journal of control.

2. Modelling and Simulation of Combined Cycle Power Plant

2.1 Introduction

During the last few decades, electrical power generation has undergone several extremely significant changes. These include deregulation and privatization of the power industry and its associated competition, which has led to great interest in economic and financial concerns rather than just purely engineering. Moreover, very efficient (over 38%) gas turbines have been developed with high power output, low cost and short construction time. Additionally, there has been an increased interest in environmental issues, leading to an assessment of existing greenhouse gas emissions. As a result, the use of gas turbines in electricity generation has been growing rapidly (Flynn, 2003). These have given rise to increasing interest in CCPP due to their high efficiencies and low emissions (Wood, 2008).

CCPP is a power plant system in which two types of turbines, namely a gas turbine and a steam turbine, are combined in one cycle to generate electricity as shown in Figure 2.1. The purpose of introducing a combined cycle in power plants is to reduce losses of energy from gas turbine to produce additional electricity.

The efficiency of the modern combined cycle plant has reached a level of 60%. Most of the remaining fuel energy that is wasted in this conversion process is released into the environment as waste heat. The principle behind combined heat and power (CHP), also known as co-generation, is to recover and make beneficial use of this heat. By using the heat output from the electricity production for heating or industrial applications, CHP plants generally convert 75-80% of the fuel source into useful energy, while the most modern CHP plants reach efficiencies of 90% or more (Wood, 2008).

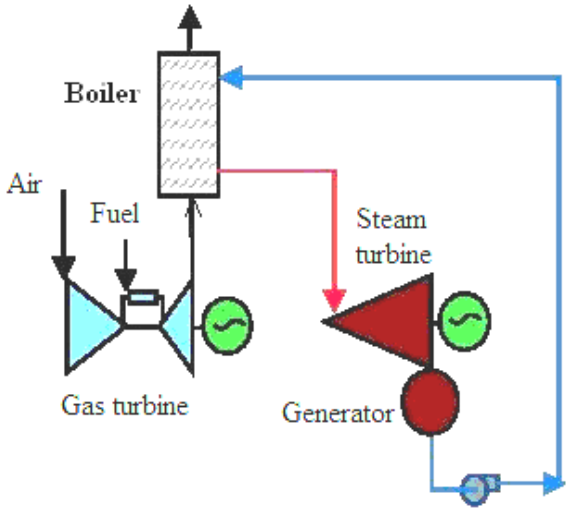


Figure 2.1: Combined cycle power plant

In this chapter, a CCPP and its subsystems, such as a boiler, steam turbine and gas turbine, will first be described. Then, a non-linear CCPP MATLAB/SIMULINK model will be developed, covering the simulator and the regulatory level controls. The non-linear model is based on the principles of mass, energy and momentum conservations that can capture the key dynamical properties over a wide operating range and is suitable for use in designing a multivariable model based controller.

2.2 Configurations of CCPP

There is a wide variety of CCPP schemes. Tomlinson and McCullough (1996) classify CCPP systems into two main types: single-shaft and multi-shaft configuration. The advantages of single-shaft arrangement include operating simplicity, higher reliability, minimum land requirements, low cost and convenient daily start-stop operation. The key advantage of a multi-shaft arrangement is its higher steam cycle efficiency (Tomlinson and McCullough, 1996). The classification of CCPP can also be based on the Heat Recovery Steam Generator (HRSG) design methods and steam turbine type. According to several authors (Ordys et al., 1994; Tomlinson and McCullough, 1996; Chase and Kehoe, 2000; Blood and Britain, 2003; Kehlhofer et al., 2009; Rayaprolu, 2010), the most popular types of CCPP are:

1- Unfired single pressure HRSG

2- CCPP with supplementary fired boiler

3- Multiple-Pressure HRSG

2.2.1 Unfired Single Pressure HRSG

As shown in Figure 2.2, this steam cycle has an unfired HRSG with superheater, evaporator, and economizer sections. Energy is recovered from the exhaust gas by convective heat transfer. This is the simplest steam cycle that can be applied in a combined cycle and it has been used extensively. It results in a low installed cost. Although it does not produce the highest combined-cycle thermal efficiency, it is an economic solution when fuel is inexpensive (Chase and Kehoe, 2000).

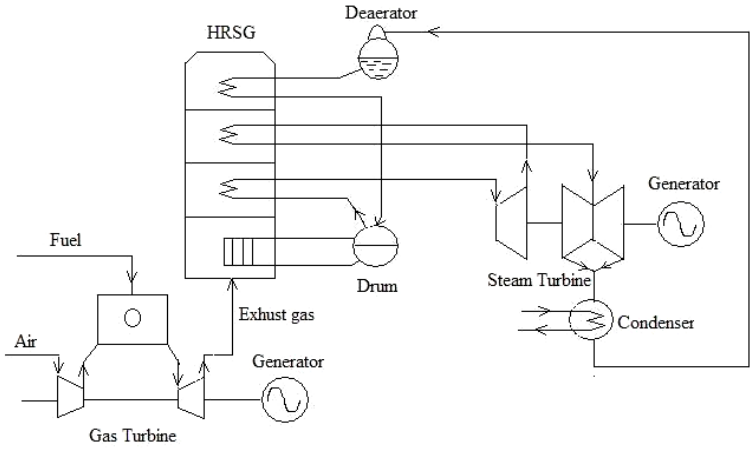


Figure 2.2: Unfired single pressure HRSG (Ordys et al., 1994)

2.2.2 CCPP with Supplementary Fired Boiler

An additional firing of the HRSG, as shown in Figure 2.3, may be used in the CCPP to increase the amount and the temperature of the generated steam. However, using supplemental firing will not raise the combined cycle efficiency. This configuration has advantages such as an increased amount and better control of system thermal output and the ability to burn fuels not suitable for gas turbines (Ordys et al., 1994).

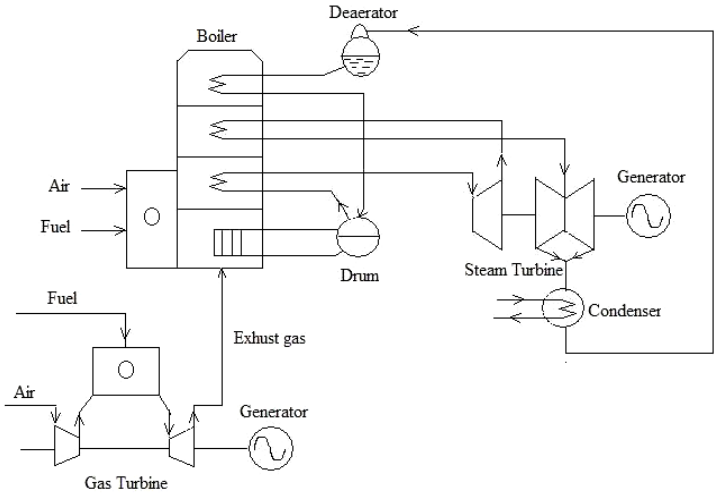


Figure 2.3: Supplementary fired HRSG (Ordys et al., 1994)

2.2.3 Multiple-pressure HRSG

Multi-pressure steam generation is used to maximize energy recovery from gas turbines that have high exhaust temperature as a means of increasing the energy conversion efficiency (Blood and Britain, 2003). A dual-pressure arrangement can increase the power output and the cycle efficiency of a single-pressure system by up to 11% and 4%, respectively (Kehlhofer et al., 2009; Rayaprolu, 2010). Most modern CCPP gas turbine of 100 MW and above that have an approximate rated exhaust gas temperature of 538°C incorporate three-pressure HRSGs (Rayaprolu, 2010). The three-pressure non-reheat CCPP is shown in Figure 2.4. The pressure levels are designated as HP, IP, and LP, denoting high, intermediate, and low, respectively. To optimize the lower pressure end of steam turbine performance and increase the combined cycle efficiency, the reheat steam circuit is commonly used with gas turbines that have an exhaust temperature of 593°C or greater. Modern CCPPs with three-pressure HRSGs and steam reheat can reach efficiency above 60% (Rayaprolu, 2010).

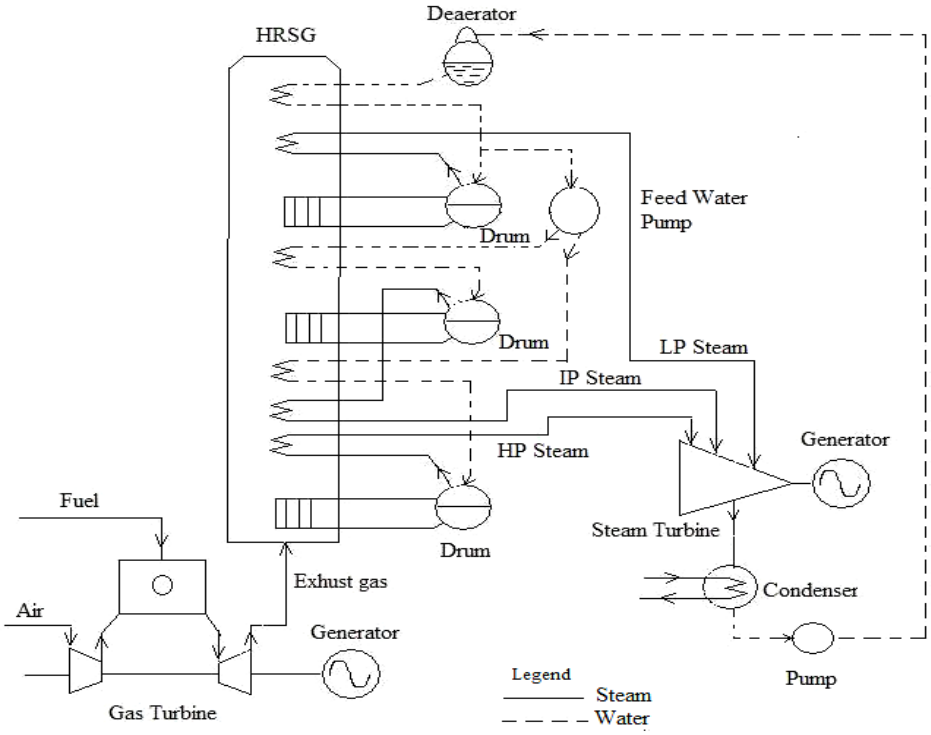


Figure 2.4: Three-pressure non-reheat CCPP

2.3 Combined Cycle Components

2.3.1 Boiler (HRSG)

A heat recovery steam generator or HRSG is an energy recovery heat exchanger that recovers heat from a hot gas stream. It produces steam that can be used to drive a steam turbine. HRSGs consist of four major components. They are the evaporator, superheater, reheater, and economizer, as shown in Figure 2.5.

The classification of an HRSG (boiler) depends on its application, design or operation. For example, it may be a fired or non-fired HRSG, single pressure or multi pressure HRSG, and horizontal or vertical design shape (Blood and Britain, 2003; Rayaprolu, 2010). One of the most common types of boiler used for steam generation in thermal power plants is the drum boiler.

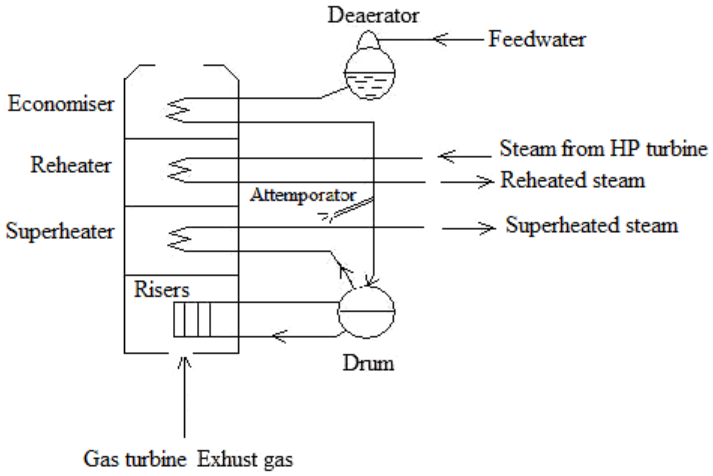


Figure 2.5: Boiler schematic diagram

2.3.1.1 Drum Type HRSGs

This boiler employs a large drum for the separation of steam and water. There are two types of drum boilers, one that uses natural circulation in the downcomer-riser loop and one where water is circulated using a pump. A drum boiler consists of several components: the drum, risers, economizer, reheater and superheater. A typical drum-type boiler is depicted in Figure 2.5. The feed-water, leaving the economizer, flows into the downcomer and then enters the risers. In the risers the heat from the gas turbine exhaust is used to increase the water temperature to cause evaporation. The generated steam leaves the drum and flows to the steam turbine through the superheater. A superheater is a heat exchanger that adds more heat to the steam created by the boiler. This increases the steam's enthalpy, allowing more work by the turbine to be generated. An attemperator is used to control the temperature of the steam exiting the superheater by mixing water at a lower temperature with the superheated steam. The advantages of the drum-type HRSG are the simple construction of the evaporator walls. A disadvantage is the pressure limit which prevents its use in steam power plants with supercritical pressures and high efficiencies (Rayaprolu, 2010).

2.3.2 Gas Turbine

The industrial power plant gas turbine consists of three major components, as shown in Figure 2.6: the compressor, the combustion chamber, and the turbine. The compressor drives air into a combustion chamber where combustion takes place when fuel (gas or oil) is added. The exhaust gas resulting from the combustion is expanded through the turbine to drive a generator and the compressor. A gas turbine has several advantages as a power source (Kehlhofer et al., 2009). It can be easily assembled and erected and it can have an efficiency ranging from 25% to 40%. It also has a high start-up speed, reaching approximately 100% load within 12 minutes in the case of a 70 MW gas turbine (Wittchow, 2002).

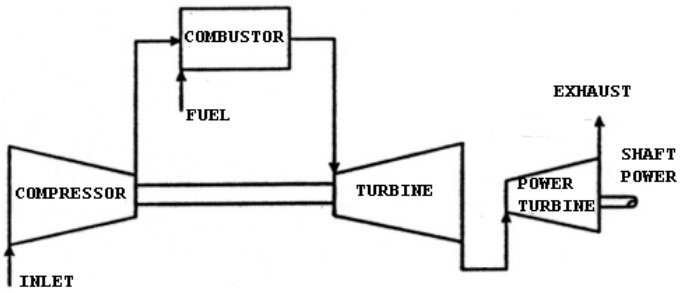


Figure 2.6: Gas turbine - schematic diagram

2.3.3 Steam Turbine

A steam turbine is a mechanical device that converts thermal steam energy into mechanical work. The turbine is directly coupled to an electrical generator to convert the rotational mechanical energy into electrical energy. Steam turbines can be classified according to their exhaust pressure as condensing or non-condensing (backpressure) types. Condensing turbines are most commonly found in electrical power plants. These turbines operate with exhaust pressures well below atmospheric pressure. Backpressure turbines operate with exhaust pressures that are equal or greater than the atmospheric pressure. These turbines are most widely used in process steam applications such as refineries, district heating units, and desalination facilities (Kehlhofer et al., 2009). In a CCPP, a condensing steam turbine uses the superheated steam from the HRSG to produce electricity. A reheat turbine can also be used to increase efficiency.

As shown in Figure 2.7, a steam turbine can be divided into three main components, namely the high pressure (HP) section, the intermediate pressure (IP) section, and the low pressure (LP) section. The steam from the HRSG enters the HP section via the main admission control valves. After passing through the HP section, the steam is circulated to the HRSG to be reheated. The steam then passes through the IP section and finally through the LP section before entering the condenser.

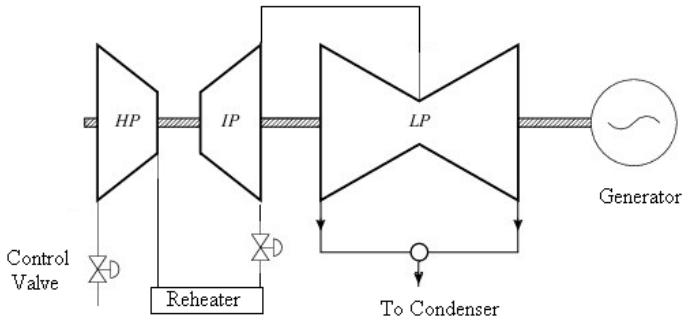


Figure 2.7: Steam turbine – schematic diagram

2.3.4 Condenser

A condenser is a device used in steam turbines to condense the steam that exits the steam turbine. The most common type of condenser applied in industry is the shell and tube type, where the heat is removed from the steam by the use of a cooling medium, usually air or water.

2.4 Non-linear Model of CCPP

The first step in the modelling procedure of large-scale CCPP using first principles is to break each unit down into smaller subsystems called modules. The next step is to establish suitable mathematical equations for each module that can represent the actual plant behaviour. The output of one module serves as the input to subsequent modules, along with additional input parameters that may be required.

The dynamic model of the main components of the fired CCPP that were developed and used in this thesis is described in this section. All models were implemented using MATLAB/SIMULINK S-functions.

2.4.1 HRSG (Boiler) Model

In the literature, boiler models vary between simple ones (for example, Astrom and Bell (1988) and De Mello (1991)) and very complex ones (for example, McDonald and Kwatny (1970) and Busi and Cori (1977)). Ordys et al. (1994) describe a moderately complicated boiler model for a CCPP that is suitable for model based control. It consists of the following subsystems: furnace, risers, drum, superheater, reheater, and economizer. This model is based mainly on the paper of Chien et al. (1958) with some extensions from the paper of Nicholson (1964). The furnace model was created from Nicholson (1964), Busi and Cori (1977) and Rhine et al. (1991). This model has been widely used for model based control systems (for example, Saez et al. (2005) and Harish et al. (2010)). However, it has the drawback that the boiler model cannot capture the boiler shrink and swell phenomenon, where bubbles of steam below the water surface level will shrink or swell, causing the level to initially move in the opposite direction to that expected. This phenomenon causes the non-minimum phase behaviour of level dynamics.

In this thesis, the boiler model described in Ordys et al (1994) is modified by replacing the risers and drum model with a simple fourth order non-linear drum model (Åström and Bell, 2000) that can capture much of the system dynamics, such as the shrink and swell phenomenon. The complete boiler system consists of an economizer, reheater, superheater, furnace, risers, and drum, as shown in Figure 2.5.

The analytical model of the drum system is developed based upon mass and energy balance. The global mass and energy balances are given by:

$$\frac{d}{dt}\left[\rho_s V_{st} + \rho_w V_{wt}\right] = q_f - q_s \tag{2.1}$$

$$\frac{d}{dt}\left[\rho_s h_s V_{st} + \rho_w h_w V_{wt} - pV_t + m_t C_p t_m\right] = Q_p + q_f h_f - q_s h_s \tag{2.2}$$

where V_{st} and V_{wt} represent the total steam and water volumes, respectively, Q_p is the heat flow rate to the risers, q_f is the feedwater mass flow rate, q_s is the steam mass flow rate, and h_s, h_w h_f are the specific enthalpy of steam, water and feedwater,

respectively. ρ_s and ρ_w are steam and water density. C_p and t_m is the specific heat and temperature of metal. m_t is the total metal mass. The total volume of the drum, downcomers, and risers is:

$$V_t = V_s + V_w \tag{2.3}$$

The global energy balance for the riser section is given by:

$$\begin{aligned} \frac{d}{dt}\left[\rho_s h_s \bar{\alpha}_v V_r + \rho_w h_w (1-\bar{\alpha}_v) V_r - pV + m_r C_p t_s\right] \\ = Q_p + q_{dc} h_w - (\alpha_r h_c + h_w) q_r \end{aligned} \tag{2.4}$$

where q_r is the total mass flow rate out of the risers and q_{dc} is the total mass flow rate into the risers, V_r represents the riser volume, α_r is the steam quality at the riser outlet, and $\bar{\alpha}_v$ is the average volume fraction. The nominal of the dynamics equation can be summarized as:

$$\begin{aligned} e_{11}\frac{dV_{wt}}{dt} + e_{12}\frac{dp}{dt} &= q_f - q_s \\ e_{21}\frac{dV_{wt}}{dt} + e_{22}\frac{dp}{dt} &= Q_p + q_f h_f - q_s h_s \\ e_{32}\frac{dp}{dt} + e_{33}\frac{d\alpha_r}{dt} &= Q_p - \alpha_r h_c q_{dc} \\ e_{42}\frac{dp}{dt} + e_{43}\frac{d\alpha_r}{dt} + e_{44}\frac{dV_{sd}}{dt} &= \frac{\rho_s}{T_d}\left(V_{sd}^0 - V_{sd}\right) + \frac{h_f - h_w}{h_c} q_f \end{aligned} \tag{2.5}$$

The model consists of four states: drum pressure p, total water volume V_{wt}, steam quality at the riser outlet α_r, and volume of steam under the liquid level in the drum V_{sd}. V_{sd}^0 is the volume of steam in the drum when there is no condensation. The coefficients e_{11}, e_{12}, e_{21}, e_{22}, e_{32}, e_{33}, e_{42}, e_{43}, and e_{44} can be obtained from (Åström and Bell, 2000).

The drum level ℓ measured from its normal operating level is:

$$\ell = \frac{V_{wd} + V_{sd}}{A_d} \tag{2.6}$$

The volume of water in the drum is:

$$V_{wd} = V_{wt} - V_{dc} - (1 - \bar{\alpha}_v) V_r \tag{2.7}$$

The simulation is carried out based on an approximation of steam tables with quadratic functions as follows (Mrunalini et al., 2006):

$$\begin{aligned}
h_s &= a_{01} + \left(a_{11} + a_{21}\left(p - 10\right)\right)\left(p - 10\right) \\
\rho_s &= a_{02} + \left(a_{12} + a_{22}\left(p - 10\right)\right)\left(p - 10\right) \\
h_w &= a_{03} + \left(a_{13} + a_{23}\left(p - 10\right)\right)\left(p - 10\right) \\
\rho_w &= a_{04} + \left(a_{14} + a_{24}\left(p - 10\right)\right)\left(p - 10\right) \\
t_s &= a_{05} + \left(a_{15} + a_{25}\left(p - 10\right)\right)\left(p - 10\right) \\
h_c &= h_s - h_w
\end{aligned} \tag{2.8}$$

$$\begin{aligned}
&\frac{\partial h_s}{\partial p} = a_{11} + 2a_{21}\left(p - 10\right) \quad \frac{\partial \rho_s}{\partial p} = a_{12} + 2a_{22}\left(p - 10\right) \\
&\frac{\partial h_w}{\partial p} = a_{13} + 2a_{23}\left(p - 10\right) \quad \frac{\partial \rho_w}{\partial p} = a_{14} + 2a_{24}\left(p - 10\right) \\
&\frac{\partial t_s}{\partial p} = a_{15} + 2a_{25}\left(p - 10\right)
\end{aligned} \tag{2.9}$$

Where

$$\begin{aligned}
&a_{01} = 2.728 \times 10^6 \quad a_{11} = 1.792 \times 10^4 \quad a_{21} = -924.0 \\
&a_{02} = 55.43 \quad a_{12} = 7.136 \quad a_{22} = 0.224 \\
&a_{03} = 1.408 \times 10^6 \quad a_{13} = 4.565 \times 10^4 \quad a_{23} = -1010.0 \\
&a_{04} = 691.35 \quad a_{14} = -1.867 \quad a_{24} = 0.081 \\
&a_{05} = 311.0 \quad a_{15} = 7.822 \quad a_{25} = -0.32
\end{aligned}$$

The feedwater specific enthalpy h_f can be calculated as follows (Åström and Bell, 2000):

$$h_f = 1000\left(C_{fw} t_f + p\frac{1000}{\rho_w} \right) \tag{2.10}$$

In this study, the drum model was simulated using Skegton power plant data, as presented in Ordys et al. (1994). Due to a lack of data availability, some parameters (V_{sd}^0, T_d and K) were assumed by scaling down the values from Åström and Bell (2000). The total drum volume was calculated from Ordys et al. (1994) as follows:

$$V_d = \frac{m\_dl}{\rho_w} \tag{2.11}$$

where m_dl and ρ_w represent the total drum liquid mass and the drum water density, respectively. The steam quality α_r is obtained by solving the following two non-linear equilibrium equations using the Fsolve MATLAB function:

$$Q_p = \alpha_r h_c \sqrt{\frac{2\rho_w A_{dc}\left(\rho_w - \rho_s\right) g \bar{\alpha}_v V_r}{k_{fr}}} \tag{2.12}$$

$$\bar{\alpha}_v = \frac{\rho_w}{\rho_w - \rho_s}\left(1 - \frac{\rho_s}{(\rho_w - \rho_s)\alpha_r} \ln\left(1 + \frac{\rho_w - \rho_s}{\rho_s}\alpha_r \right) \right) \tag{2.13}$$

The steam volume in the drum is calculated as follows:

$$V_{sd} = V_{sd}^0 - \frac{T_d\left(h_w - h_f\right)}{\rho_s h_c} q_f \tag{2.14}$$

Parameters and the calculated equilibrium values used in the simulation are listed in Table 2.1 and Table 2.2.

Table 2.1: Boiler parameters

Parameter	Value	**Parameter**	Value
Total metal mass m_t	45,000 kg	Drum mass m_d	15,000 kg
Riser metal mass m_r	22,500 kg	Riser volume V_r	6.53 m^3
Residence time T_d	2.93 s	Drum volume V_d	9.25 m^3
Friction coefficient k_{fr}	2.98	Area of drum A_d	1.17 m^2
Empirical parameter B	0.3	Area – downcomer A_{dc}	0.1 m^2
V^0_{sd}	1.572 m^3	Downcomer volume V_{dc}	2.1 m^3

Table 2.2: The inputs and initial states for boiler

Inputs	Value	**Initial states**	Value
Steam flow rate	12 kg/s	Total water volume	10.87 m3
Water flow rate	12 kg/s	Drum pressure	4.5417 Mpa
Heating value	24.48 MW	Steam quality	0.02334
Feedwater enthalpy	5.6105 kJ/kg	Volume of steam in drum	0.9882 m3

In order to verify the correct behaviour of the simulated model, open loop tests were performed by simulating the response to 3.4 kg/s step change in steam flow rate, as shown in Figure 2.8.

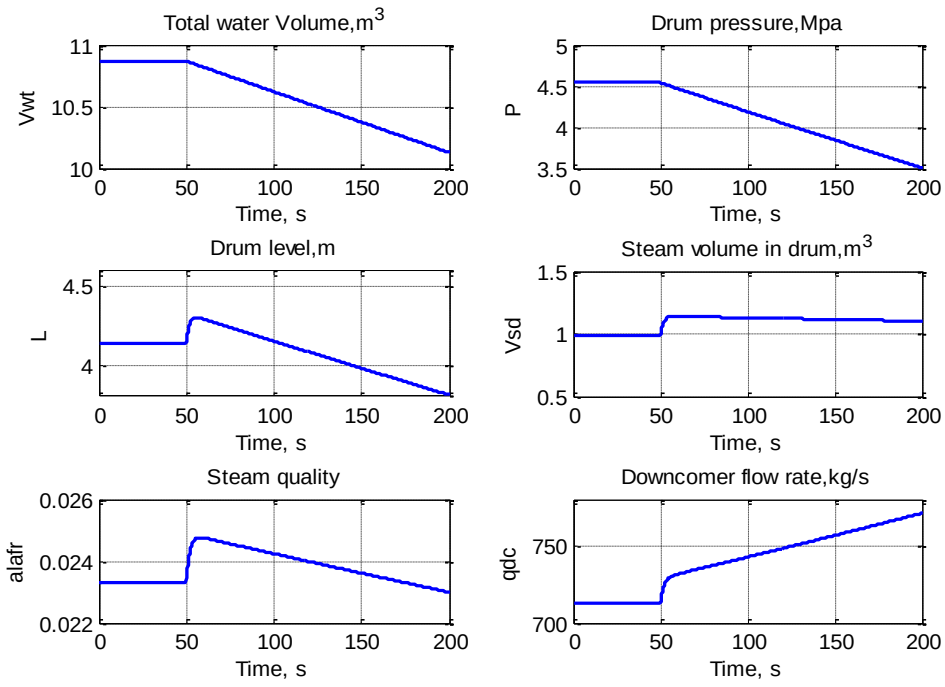


Figure 2.8: Open loop response to step change in steam flow rate

As shown in Figure 2.8, the pressure and the water volume both decreased due to the increase in steam flow from boiler-drum to steam turbine. The decrease in pressure also led to an increase in the evaporation rate which in turn led to a decrease in total water volume. The pressure drop also caused the steam quality at the riser outlet to increase initially and then decrease due to the increased steam flow rate. The steam

volume in the drum (V_{sd}) increased slightly due to a pressure drop. Finally, the figure shows the initial increase of the drum water level due to the shrink and swell phenomenon. The open loop simulation results were in good agreement with Åström and Bell's (2000) results.

A comparison between the open loop results of this new drum model and the drum and riser model of Ordys et al.(1994) using the Skegton power plant data is shown in Figure 2.9. The results show the advantage of this model in terms of capturing the shrink and swell non-minimum phase behaviour.

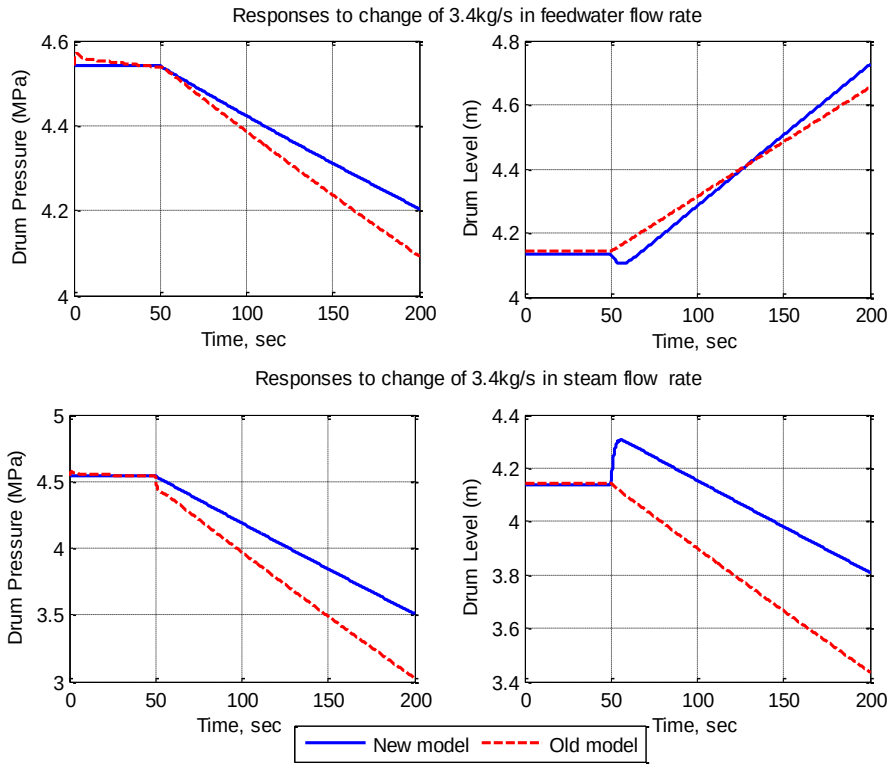


Figure 2.9: Comparison between two drum boiler models

To create a complete boiler model, the boiler drum model was integrated with the other boiler subsystems models, which are the furnace, economiser, reheater, and superheater, as described in Ordys et al.(1994) and given in Appendix A. The complete boiler process was represented by a 16th order non-linear model using the MATLAB/SIMULINK S-function.

Figure 2.10 presents a block diagram of the boiler model including the interconnected subsystems.

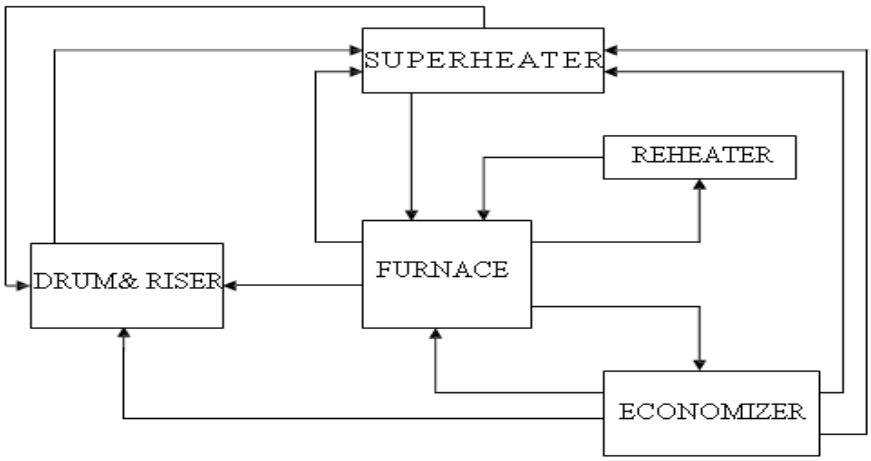


Figure 2.10: Boiler block diagram

2.4.2 Gas Turbine Model

Gas turbine complex dynamic models based upon fundamental mass, momentum and energy balances have been reported by various authors (Hussain and Seifi, 1992; Schobeiri et al., 1994; Camporeale et al., 2006). Rowen (1983) presented a simplified mathematical model of a heavy-duty single-shaft gas turbine. Ordys et al. (1994) described an intermediate gas turbine model based on Rowen (1983) and Hussain and Seifi (1992). This model has been validated using real data from a 350-MW CCPP in Chile (Saez et al., 2007). It has been used in a predictive supervisory controller (for example, Saez et al.(2007) and Hooshmandi et al.(2009)).

In this thesis, the Skegton (34 MW output power) gas turbine model described in Ordys et al.(1994) was used to construct a MATLAB/SIMULINK simulator. The gas turbine model is divided into four modules: fuel system (fuel valve and actuator), compressor, combustor, and turbine as shown in Figure 2.11.

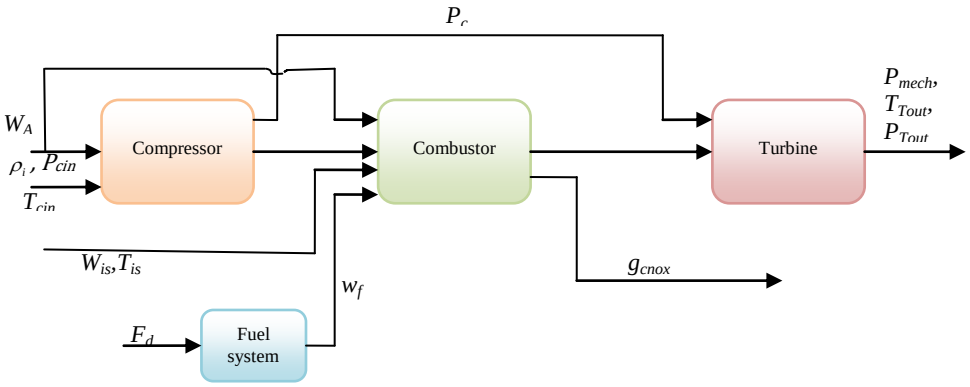


Figure 2.11: Gas turbine block diagram

The fuel dynamic module shown in Figure 2.11 is designed to provide energy input to the gas turbine in proportion to the product of command signal times the unit speed, as shown in Figure 2.12 (Rowen, 1983). The input signal F_D is the fuel demand signal from the gas turbine control block. The output signal is the fuel flow w_{fG} to the combustion. The fuel limit is used to ensure that the fuel signal remains within physically obtainable values. The 0.23 offset value is the no load self-sustaining condition essential to maintain the compressor in operation. The fuel is fed into the combustor through two valves in series. The first valve controls the pressure between the two valves, and the second regulates the actual fuel flow. The transport delay is related to the combustion reaction (Rowen, 1983).

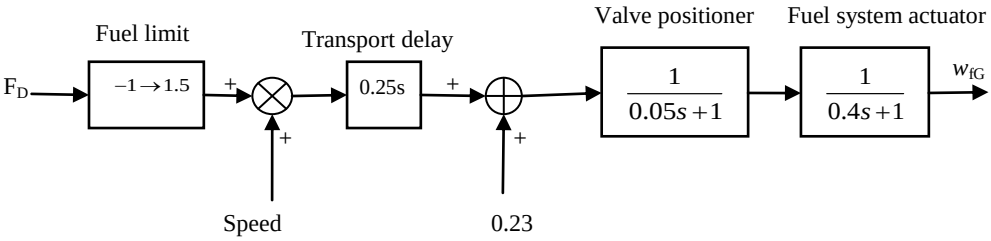


Figure 2.12: Fuel dynamic block diagram

The dynamic behaviour of the compressor, combustor and turbine are described by means of algebraic equations representing the thermodynamic transformations and kinematic balance, as described in Ordys et al.(1994) and listed in Appendix A.

2.4.3 Steam Turbine Model

A complex steam turbine dynamic model based upon fundamental mass, momentum and energy balances has been described by various authors (Committee, 1973; Ray, 1980; De Mello, 1991). A simplified linear model for transient stability studies of electrical network has also been proposed by them (Committee, 1973; Ray, 1980; De Mello, 1991). Ordys et al. (1994) presented an intermediate model that includes the more important non-linearities associated with the turbine. This model has been used for model based control (for example, in Flynn (2003) and Saez et al. (2005)).

In this thesis, the steam turbine model equations are adopted from Ordys et al. (1994). These equations are programmed using MATLAB S-Function and simulated with the reheater module by using Skegton (11 MW output power) Steam turbine data. The steam turbine model is divided into three modules: high pressure section (HP), intermediate pressure section (IP), and low pressure section (LP). Figure 2.13 shows the complete steam turbine model including the boiler reheater module.

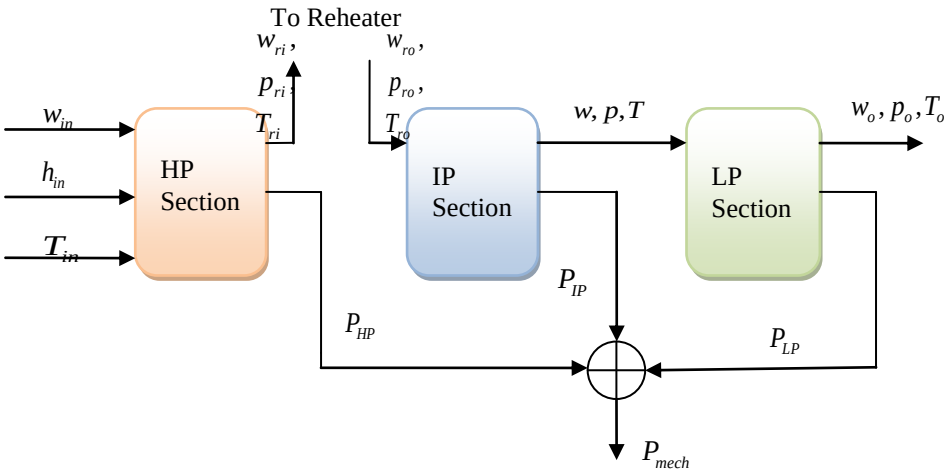


Figure 2.13: Steam turbine block diagram

2.4.4 Electrical Generator Model

In this study, a simple generator model was used which included only the real power and frequency variation, as follows (Ordys et al., 1994):

$$\frac{d\omega_g}{dt} = \frac{\omega_0}{2H_i}\left[P_{mech} - P_e - D_i(\omega_g - \omega_0)\right] \tag{2.15}$$

where ω_g, H_i and D_i are the frequency of the generator, the inertia constant and the damping coefficient, respectively, and ω_0 represents the synchronous frequency. The model block diagram is shown in Figure 2.14.

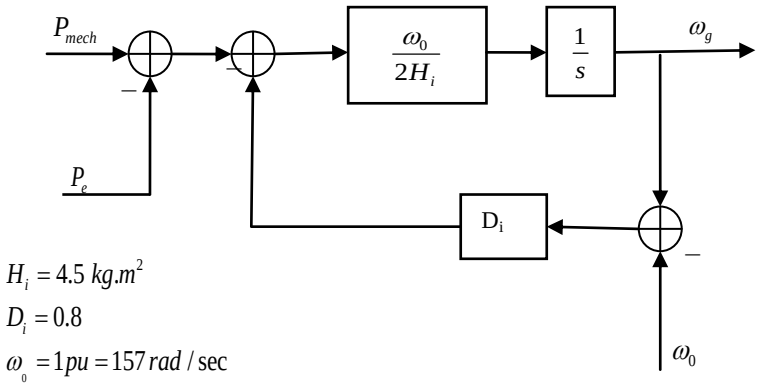


Figure 2.14: Electrical generator block diagram

2.5 Control System for Combined Cycle Power Plant

2.5.1 Control Configurations

The complexity and size of the combined cycle power plant requires large interacting control loops. For example, the boiler with its control loops and the steam and gas turbines with their associated loops and auxiliary control loops. Coordinated total plant-wide is required for optimized system operation. Consequently, a control system of a power plant consists of a hierarchy Distributed Control System (DCS), as shown in Figure 2.15. The DCS distributes the hardware physically throughout the plant. The system reads process inputs such as thermocouples and pressure sensors, calculates the optimal outputs, allows operator interface, and drives output devices such as valve actuators and pumps. To reduce the amount of wiring to a central location, control is done by digital controllers located near the process instruments, as shown in Figure 2.15.

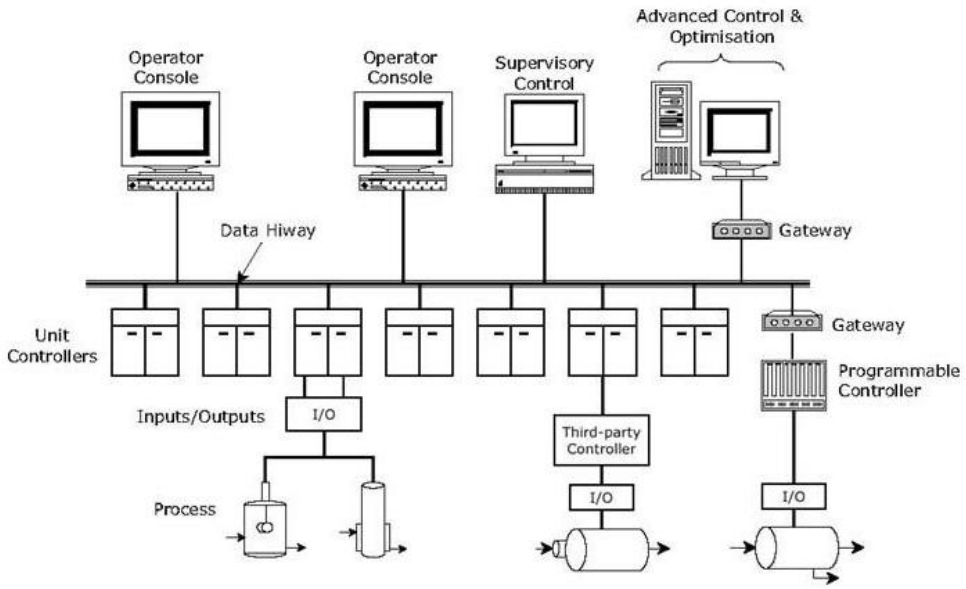


Figure 2.15: Distributed control system (Chokshi and McFarlane, 2008)

As shown in Figure 2.15, the basic function of a DCS can be classified into two groups: regulatory control level and supervisory control. The main purpose of the regulatory control layer is to keep the plant in safe and stable operation. The supervisory controller provides the regulatory level set-points based on the optimization of the objective function. Therefore, the main tasks of the supervisory level are: to move the system from one operating point to another without oscillations, to handle the process output and input constraints and to provide a high level of disturbance rejection.

At the regulatory level, the decentralized or multi-loop PID control configuration is the most common practice for most industrial processes. In this configuration, the multi input/multi output (MIMO) system is divided into individual single input/single output (SISO) loops according to a previous selection of the pairings between inputs and outputs. This configuration might not be optimal for systems where the different loops are highly coupled and the controllers are tuned individually. Therefore, a tuning procedure is needed to consider the interactions between subsystems.

There are many methods for the optimal pairing of controlled variables but the most common is the Relative Gain Array (RGA) due to its simplicity and utility. The RGA

was introduced by Bristol (1966) as a steady-state measure of process interaction in decentralized control. Skogestad and Morari (1987) demonstrated that the RGA calculation can be expressed in matrix notation Λ as function of steady-state gain matrix G_0 as follows :

$$\Lambda = G_0 \times (G_0^T)^{-1} \tag{2.16}$$

2.5.2 Decentralized PID Tuning

Control of interacting multivariable processes can be realized either by centralized MIMO controllers or by decentralized controllers. Decentralized PID control has been widely used in MIMO industrial processes due to the simplicity in implementation and loop failure tolerance of the resulting control system. In this configuration, the MIMO system is divided into individual SISO PID loops and tuned mainly on a single loop basis. However, the process interactions in MIMO process makes the proper tuning of decentralized controllers much more difficult than that of SISO PID controllers. This is because adjusting one loop may affect the performance of the others or destabilize the entire system. For this reason, the number of tuning methods for a MIMO system is limited.

The tuning methods for decentralized controllers described in the literature can be classified under three groups, namely, the detuning method, sequential closing method, and independent design method. In the detuning method, each controller of the multi loop control system is first designed using single loop tuning rules, ignoring process interactions from other loops. Then, each controller is detuned to preserve stability or to meet some performance specification. Among these methods is the Biggest Log Modulus (BLT) proposed by Luyben (1986). In the sequential closing methods such as relay feedback method, each controller is tuned sequentially, where the input-output pair of the fastest loops is tuned and this loop is closed. Then, the controller of the lower loops is tuned while the first controller is closed and so on (Mayne, 1973; Loh et al., 1993). Finally, in the independent design methods, for example IMC methods, each controller is designed based on the open-loop and closed-loop paired transfer functions, while satisfying some constraints to guarantee stability and performance (Skogestad and Morari, 1989; Vu and Lee, 2010).

Various advanced methods have been used to obtain optimum PID parameters, such as genetic algorithms (Herreros et al., 2002), particle swarm optimization (de Moura Oliveira, 2005), and multi-objective $h_{\inf}$ robust techniques based on linear matrix inequalities (LMIs) (Takahashi et al., 1996; Goncalves et al., 2008). However, these methods have problems of either enormous computation efforts or difficulty in tackling non-convex optimization problems (Goncalves et al., 2008).

In this study, relay feedback and IMC tuning methods for decentralized control are considered to tune PID parameters of power plant regulatory controllers.

2.5.2.1 Relay Feedback Tuning Method

A relay feedback auto-tuning technique based in the Ziegler-Nichols method for closed loop systems was proposed by Åström and Hägglund (1984; 1988) . In this method, the controller is replaced by a relay which induces a sustained oscillation. The frequency of this oscillation (P_u) and its amplitude (A), as shown in Figure 2.16, can be used to determine the parameters of the PID controller based on Ziegler-Nichols tuning rules. The advantages of the relay feedback technique are that the tuning can be done online and there is less chance of the system becoming unstable during the tuning of the PID controller since it is a closed-loop method. The ultimate gain can be computed as:

$$K_{cu} = \frac{4h}{\pi A} \tag{2.17}$$

where h is the height of the relay input and A is the amplitude of oscillation, as shown in Figure 2.16.

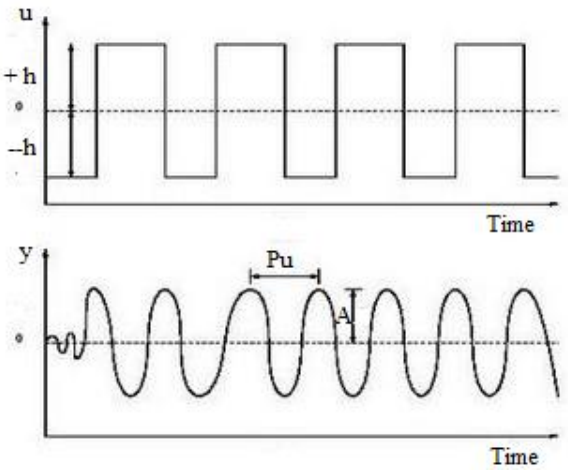


Figure 2.16: Closed-loop response of relay feedback

Having determined the ultimate gain K_{cu} and the oscillation period P_u, the PI controller tuning parameters can be obtained using Ziegler-Nichols or Tyreus-Luyben rules (Luyben and Luyben, 1997), as shown in Table 2.3.

Table 2.3: Ziegler-Nichols and Tyreus-Luyben tuning rules for PI

Controller method	K_p	T_i
Ziegler-Nichols	K_{cu} / 2.2	P_u /1.2
Tyreus-Luyben	$0.31 K_{cu}$	$2.2 P_u$

In relay auto-tuning for a MIMO control system, the relay feedback method is usually applied using a sequential design strategy. This strategy involves closing each loop once it is tuned, until all loops are covered. Example of this approach can be found in Loh et al. (1993). The tuning sequence should be repeated in an iterative manner to account for the effect of loop interactions. Faster convergence can be achieved when the fast loop is tuned first (Yu, 2006). However, for unstable control systems it is better to close the unstable loop first in order to cope with instabilities.

2.5.2.2 Internal Model Control Tuning Method

The PID parameter tuning method based on the IMC was developed by Rlvera et al.(1986). In this approach, the control tuning is based on a priori process model and a low pass filter is included for robustness. This method has the advantage of only using a single tuning parameter to achieve a clear tradeoff between closed-loop performance and robustness to model inaccuracies. Lee et al. (2006) suggested modifying IMC_PID tuning methods for different types of processes. In these methods, the PID parameters are obtained by a Maclaurin series expansion in the Laplace variable of the simple feedback form of a multivariable IMC controller. Skogestad (2003) proposed a modified SIMC method to improve the load disturbance rejection. Lieslehto and Koivo (1987) developed a multivariable tuning method based on the IMC technique. In the multivariable tuning method based on IMC technique, the PID parameters are usually designed by ignoring process interactions and then the individual loops are detuned iteratively to satisfy robust stability and to meet some performance requirement (Grosdidier and Morari, 1986; Lee et al., 2004; Vu et al., 2007).

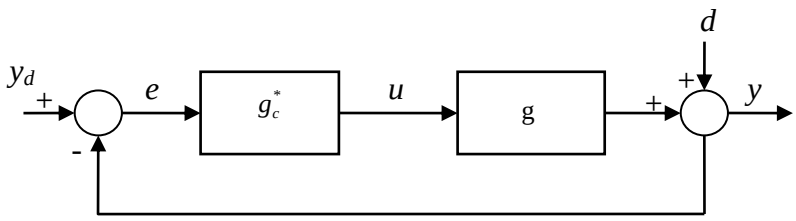
The block diagram for conventional feedback control is shown in Figure 2.17(a), where g_c^* is the PI controller and given by:

$$g_c^* = K_c\left(1+\frac{1}{\tau_I s}\right) \tag{2.18}$$

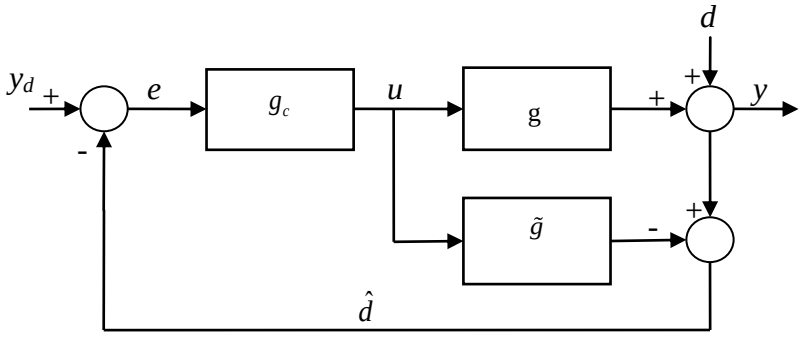
where K_c and τ_I are the proportional gain and the integral time, respectively.

The structure of the IMC is shown in Figure 2.17(b), where g_c and $\tilde{g}$ are the IMC controller and the internal model, respectively. In the IMC design, the process model is factored as:

$$\tilde{g} = \tilde{g}_+\tilde{g}_- \tag{2.19}$$



(a) Classical feedback control



(b) Internal model control

Figure 2.17: (a) conventional configuration and (b) IMC configuration

where $\tilde{g}_+$ contains the time delay and RHP zeros of the model, and $\tilde{g}_-$ is the invertible part which can be used as the controller g_c. A low pass filter is added in series with the controller to attenuate the modelling error. The low pass filter can be defined as:

$$f = \frac{1}{(1+\lambda s)^n} \tag{2.20}$$

where the time constant λ is used as a fine tuning parameter of the IMC method and represents the closed loop speed of response. The parameter n represents the order of the filter and is usually selected as first order (n=1) in a first order open-loop stable process.

Given the Laplace transfer function of a process model without delay:

$$\tilde{g} = \frac{K}{\tau s + 1} \tag{2.21}$$

The IMC controller g_c is given by:

$$gc = \frac{1}{\tilde{g}} f = \frac{\tau s + 1}{K(1 + \lambda s)} \tag{2.22}$$

By comparing Figure 2.17(a) with Figure 2.17(b), the following relation is given:

$$g_c^* = \frac{g_c}{1 - \tilde{g} g_c} = \frac{\tau}{K\lambda}\left(1 + \frac{1}{\tau s}\right) \tag{2.23}$$

From this equation, the gain and integral time of the ideal PI controller can be defined as:

$$K_c = \frac{\tau}{K\lambda} \tag{2.24}$$

$$\tau_I = \tau \tag{2.25}$$

For first or second order models with relatively small delay, the standard IMC controllers provide sluggish disturbance rejections because τ_I is very large. To solve this problem, Skogestad (2003) has proposed limiting the value of τ_I :

$$\tau_I = \min\{\tau_1, 4(\lambda + \theta)\} \tag{2.26}$$

where τ_1 and θ are the largest time constant and the time delay, respectively.

The generalized IMC-PID approach can be extended to unstable processes. Given a first order delayed unstable process:

$$G(s) = \frac{Ke^{-\theta s}}{\tau s - 1} \tag{2.27}$$

The tuning rules based on the IMC_PID method for this process can be calculated as follows (Lee et al., 2000):

$$K_c = \frac{\tau_I}{-K(2\lambda + \theta - \alpha_i)} \tag{2.28}$$

$$\tau_I = -\tau + \alpha_i - \frac{\lambda^2 + \alpha_i\theta - \frac{\theta^2}{2}}{2\lambda + \theta - \alpha_i} \tag{2.29}$$

where

$$\alpha_i = \tau\left[\left(\frac{\lambda}{\tau} + 1\right)^2 e^{\frac{\theta}{\tau}} - 1\right] \tag{2.30}$$

$$K_i = \frac{K_c}{\tau_I} \tag{2.31}$$

2.5.3 Regulatory Control of CCPP

The main purpose of the regulatory control layer is to keep the plant in safe and stable operation, by keeping the controlled variables at or close to their set-points.

2.5.3.1 Boiler Control Scheme

The boiler control system consists of four simple proportional integral (PI) control loops designed to meet the set-point requirements, as shown in Figure 2.18. They are: adjusting the induced draught fan speed to control the air pressure in the furnace, regulating a feed-water supply to maintain the drum water level, adjusting the superheat spray water flow to control the superheat steam temperature, and adjusting the fuel supply to the boiler to control the superheated steam pressure. In this case, the controlled variables are superheated steam pressure (P_s), drum level (l), superheated steam temperature (T_s) and furnace gas pressure (P_G). The manipulated variables are fuel flow (w_f), feedwater flow (w_e), attemperator water flow (w_a), and air flow to the furnace (w_A).

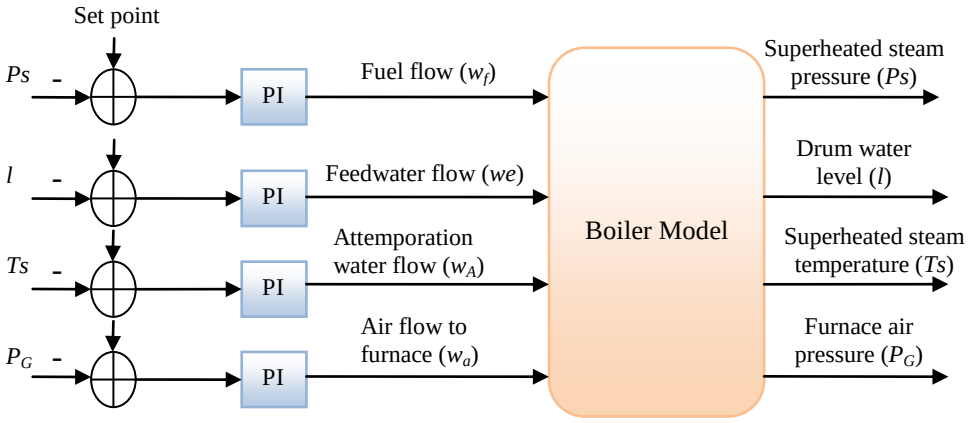


Figure 2.18: Boiler control system

2.5.3.2 Gas Turbine Control Scheme

The gas turbine control scheme is adopted from Rowen (1983), Ordys et al.(1994) and Saez et al. (2007). In this scheme, the main control loop is the speed governor. It detects frequency deviation from the nominal value and determines the controlled variable (fuel demand F_d). The speed governor loop shares its controlled variable F_d with a temperature controller and the power controller. These three different controllers' outputs are compared into a low value select (LVS) function, which selects the minimum of three incoming signals. The fuel demand signal is then fed to the gas turbine model through the fuel dynamic model as shown in Figure 2.19.

The temperature control consists of two branches. The main branch is a proportional-integral (PI) controller, which acts as an air supply control (w_a). The second branch is a proportional controller, which acts as a fuel demand controller through the LVS minimum value function. It is used to control the exhaust temperature when the main controller is not enough to maintain safe temperatures. A proportional-integral (PI) controller is used to control the injected steam flow into the combustion chamber (w_{is}) to control gas turbine NO_x emission rates.

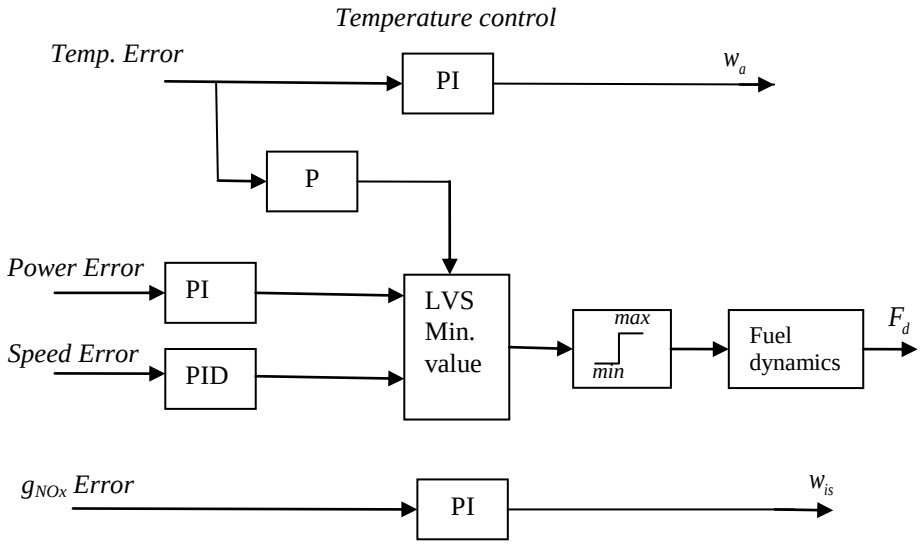


Figure 2.19: Simplified representation of gas turbine control scheme

2.5.3.3 Steam Turbine Control Scheme

In combined cycle power plants (CCPP), the objective of the steam turbine controller is to regulate the steam turbine generator (power demand control) and steam turbine governor valves. Three modes of control are available as follows (Flynn, 2003):

1- Boiler following mode (constant pressure mode): as shown in Figure 2.20, the power-demand signal controls the turbine governor valve to meet the load, while the fuel and air flow signals (firing rate) control the steam pressure in the boiler controller to keep the pressure at a constant level. This is the most common control mode in classical thermal power plants due to its fast responses to power demand.

2- Turbine following mode: in this mode the power-demand signal is fed directly to the boiler control system to regulate the boiler firing rate. The steam pressure is controlled by steam turbine governor valves. This mode achieves very stable responses with minimum pressure and temperature fluctuations and allows the generating units to operate continuously at their maximum capacity rating. However, the response to power load changes is slow. Therefore, this method is preferred for base-load thermal plants.

3- Sliding-pressure mode: this mode of operation allows the pressure to be modified for changes in power load demand. For example, this control mode reduces the boiler pressure set-point at lower loads in order to reduce control valve throttling. This mode combines some of the advantages of the previous modes but at the same time is complex and requires coordinated control.

The steam turbine control system presented in this study is based on boiler follow mode, as in Sáez et al.(2008). As shown in Figure 2.20, the steam flow from the boiler (w_{in}) is regulated by a PID controller to meet the load demand. The steam pressure and temperature are controlled in the boiler system.

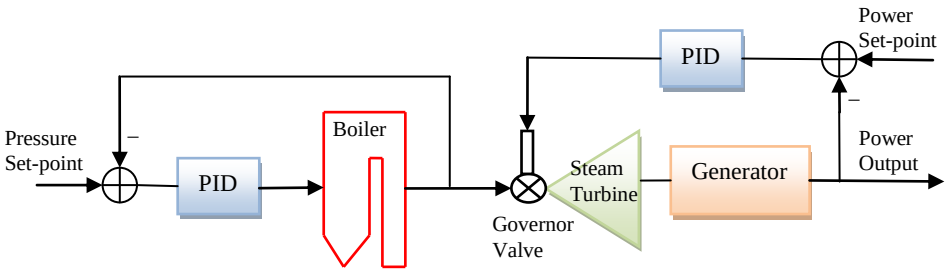


Figure 2.20: Boiler follow mode

2.5.3.4 CCPP Complete Model

Subsystem modules of CCPP have been described and implemented in MATLAB SIMULINK S-functions which provide an efficient algorithm for integration. Figure 2.21 show the hierarchical structure of CCPP. SIMULINK block diagrams of CCPP are listed in Appendix B. It consists of three subsystems: boiler system, gas turbine and generator system, and steam turbine and generator system. As shown in Figure 2.21, an integrated CCPP model is constructed by connecting these three main modules. This model describes the closed loop dynamic relationship between controlled variables (CVs) and controlled variable set-points. The controlled variables of the complete CCPP system have been chosen as follows: superheated boiler pressure (Ps), boiler drum level (L), boiler superheated temperature (Ts), furnace gas pressure (P_G), gas turbine exhaust temperature (T_{Gout}), power of gas turbine (P_{mG}), gas turbine speed (Omega), NOx level in gas turbine exhaust gases (g_{NOx}), and power of steam turbine (P_{ms}).

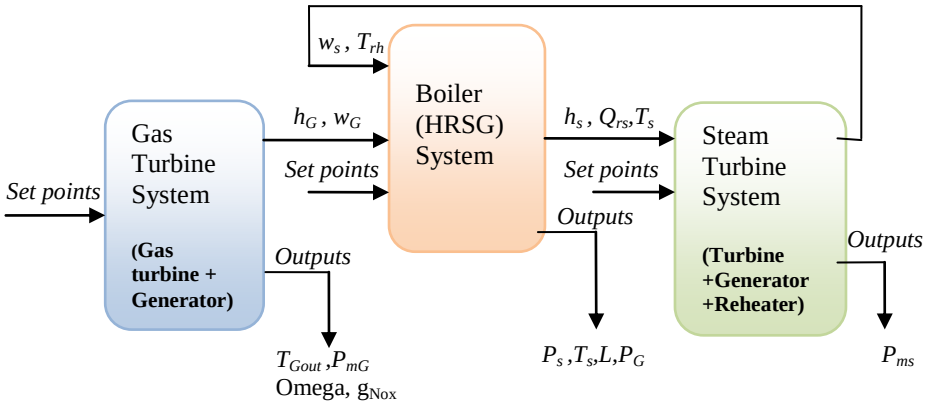


Figure 2.21: Complete CCPP system

2.5.3.5 Simulation and Results

In this simulation, the CCPP PID parameters in Ordys et al. (1994) are first used and then each controller is detuned sequentially using a trial and error method to preserve stability and to achieve good tracking and disturbance rejection performance. The tuning parameters used in this simulation are listed in Table 2.4. Advanced methods for PID tuning can be used to optimize the closed-loop performance. IMC and relay feedback tuning methods will be used in the next section to tune the PID parameters of boiler-turbine unit.

Table 2.4: CCPP tuning parameters

	K_p	K_i	K_d
Exhaust temp.	-4.0	-1.0	___
Gas turbine Power	0.05	0.5	___
Speed	0.05	0.5	0.1
Nox emission	-0.001	-0.01	___
Drum pressure	1.5e-5	1e-6	___
Water level	27.0	0.05	___
Superheated temp.	-0.1	-0.1	___
Furnace pressure	2.25e-4	3.17e-5	___
Gas turbine Power	1.23	1.23	___

To assess the performance of each module and for the complete CCPP, tests have been performed to the CCPP system with PID controllers. These tests are the model response to disturbances, set-points changes and load demand.

Figure 2.22 to Figure 2.26 show the responses of the controlled outputs and the manipulated inputs for the CCPP (boiler, gas turbine and steam turbine) to step changes in boiler superheated pressure set-point at t = 250 (s) plus ramp and step changes in gas and steam turbines' load demand. From these results it is clear that the system has good tracking performance. In addition, these tests prove the stability of the system over the working range.

Dynamic responses of the boiler to step disturbances in boiler inlet fuel flow (w_f) are shown in Figure 2.27. As shown in this figure, the disturbance can be eliminated using the PID controllers.

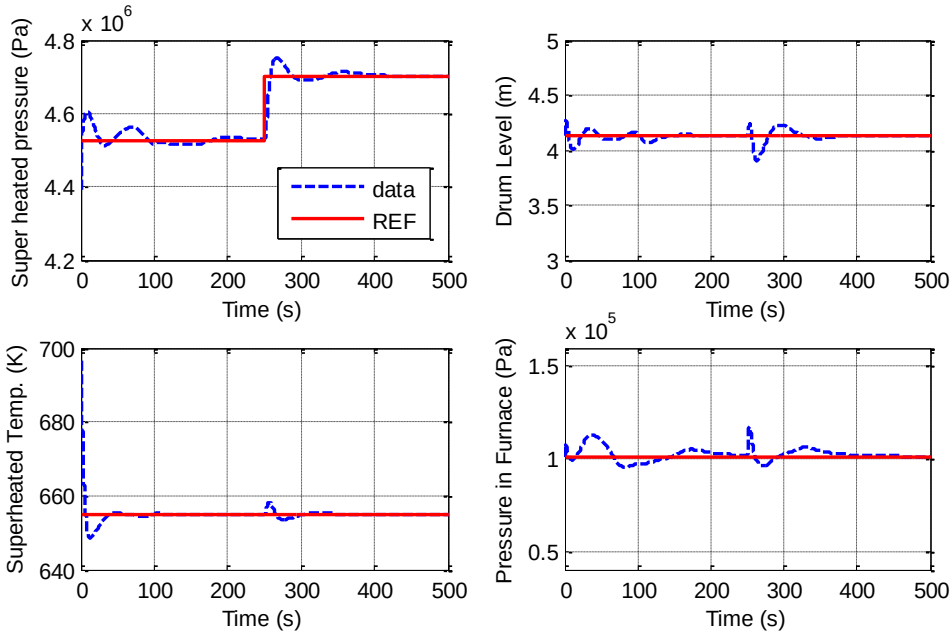


Figure 2.22: Boiler response to step changes in boiler superheated pressure (Controlled variables)

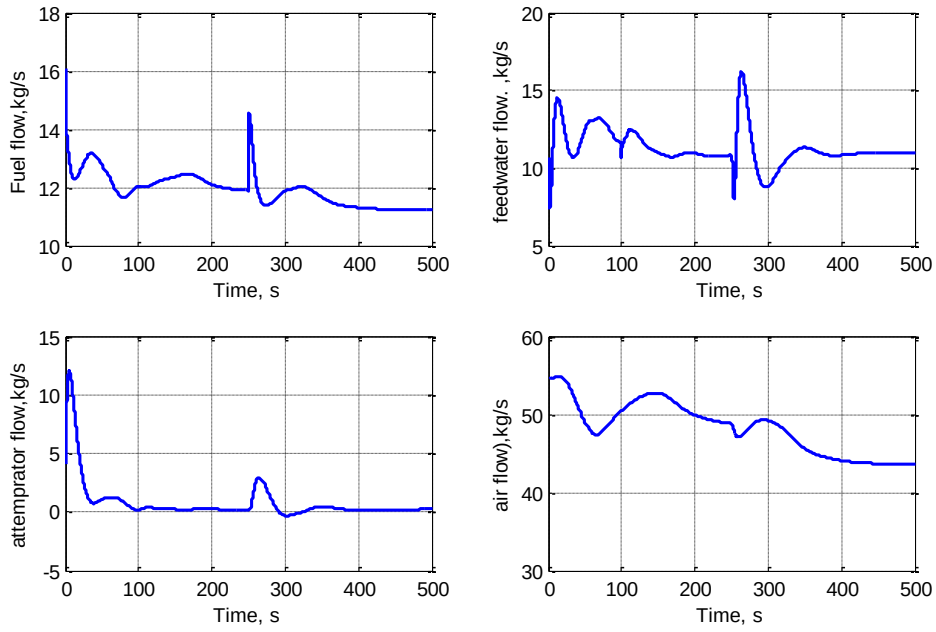


Figure 2.23: Boiler response to step changes in boiler superheated pressure (Manipulated inputs)

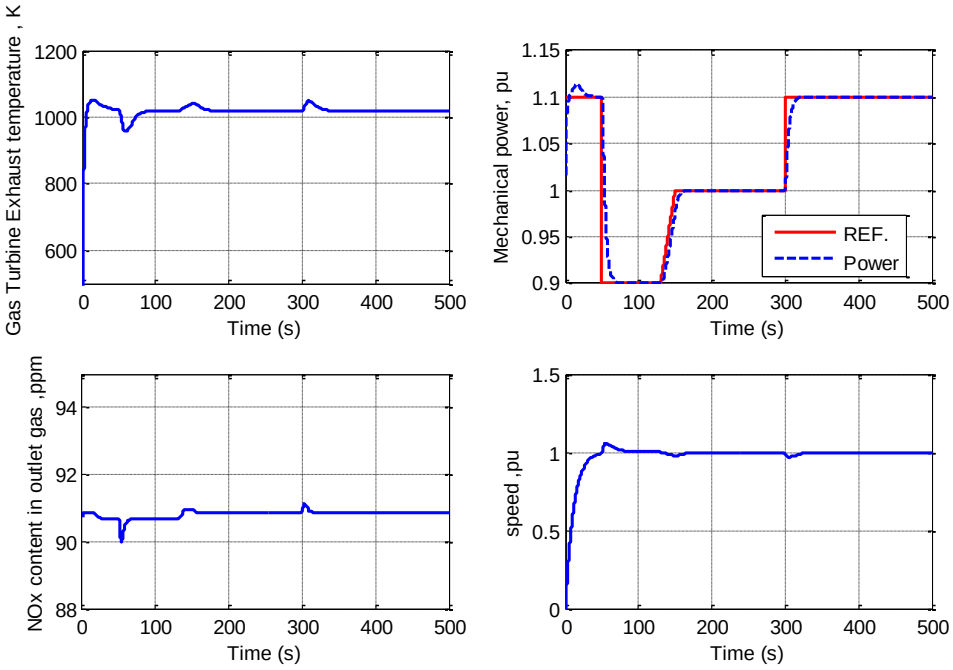


Figure 2.24: Gas turbine response to ramp and step changes in gas turbine power demand (Controlled outputs)

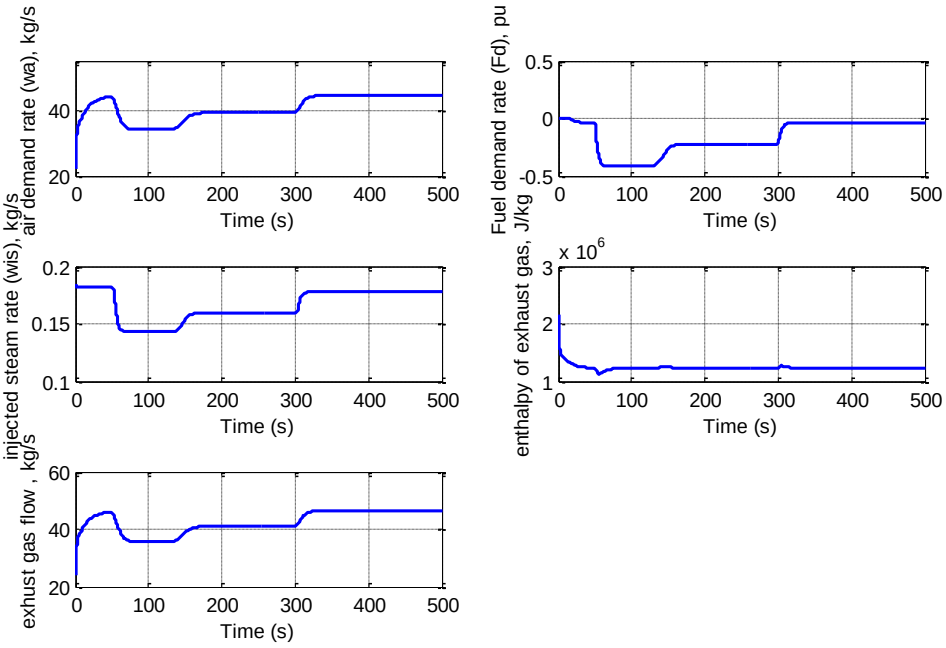


Figure 2.25: Gas turbine response to ramp and step changes in gas turbine power demand (Manipulated inputs)

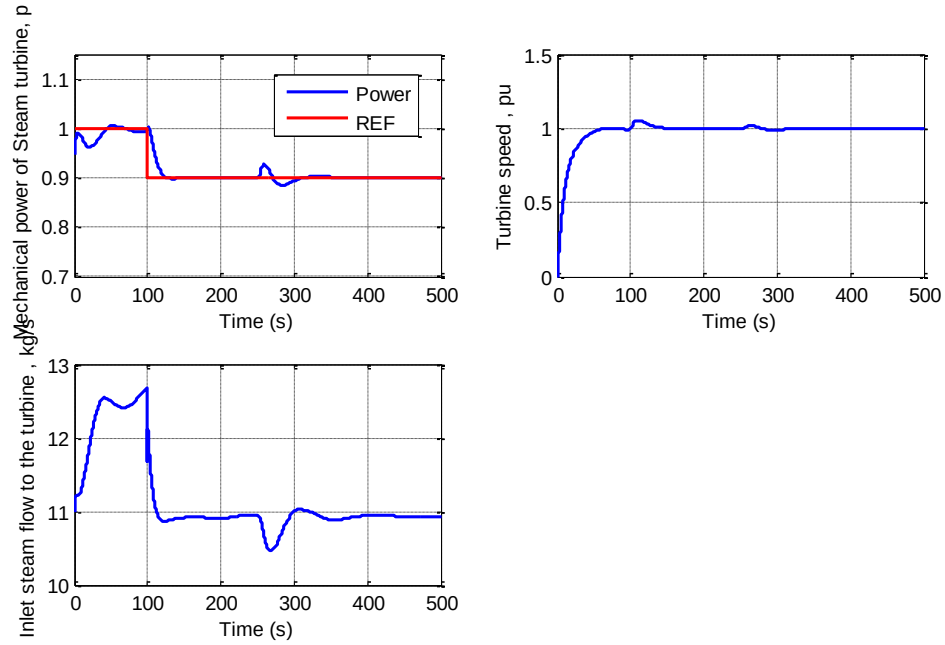


Figure 2.26: Steam turbine response to set-points changes in boiler pressure and steam turbine power demand (Controlled and manipulated variables)

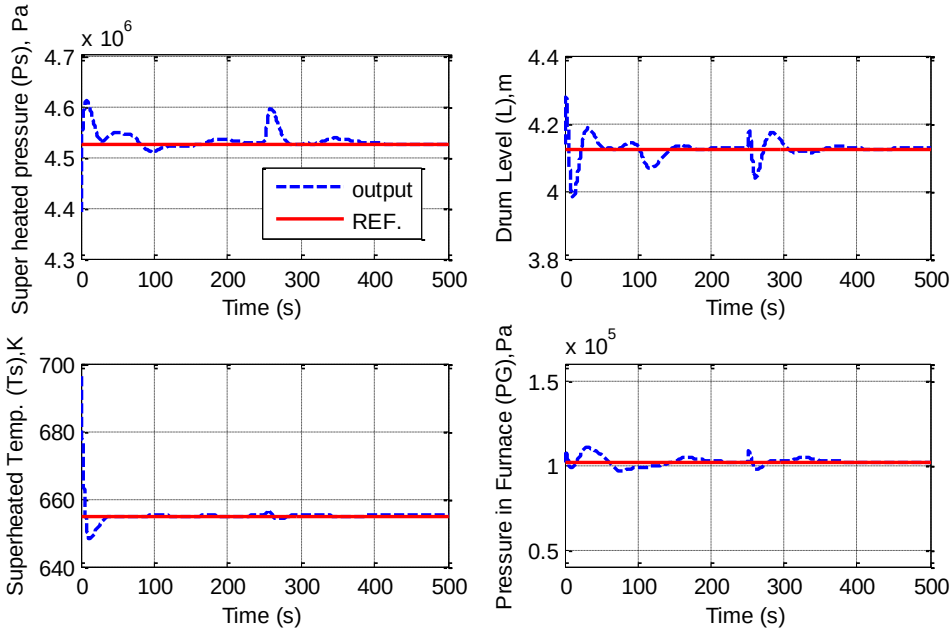


Figure 2.27: Boiler response to +5% step disturbance in boiler inlet fuel flow (w_f) (Controlled outputs)

2.6 Simplified CCPP Model

In this section, the CCPP model is approximated (reduced) by a simpler model in order to simplify the derivation of the nonlinear state-dependent matrices that will be discussed in Chapter 4. The following assumptions are considered:

1- The effect of economizer dynamics is ignored, as it makes an insignificant influence to overall system dynamics. Therefore, the feed water specific enthalpy entering the drum boiler is assumed to be constant.

2- Assume that there is no reheater in the steam turbine.

3- The steam turbine model is approximated by a single stage steam turbine model, where the output power is scaled up to be identical to the original model. Figure 2.28 shows a comparison between the original and reduced model responses to step changes in steam turbine power demand at 100 sec and step disturbance in inlet steam at 200 sec. It is clear that the reduced model can capture well the dynamic properties of the original model.

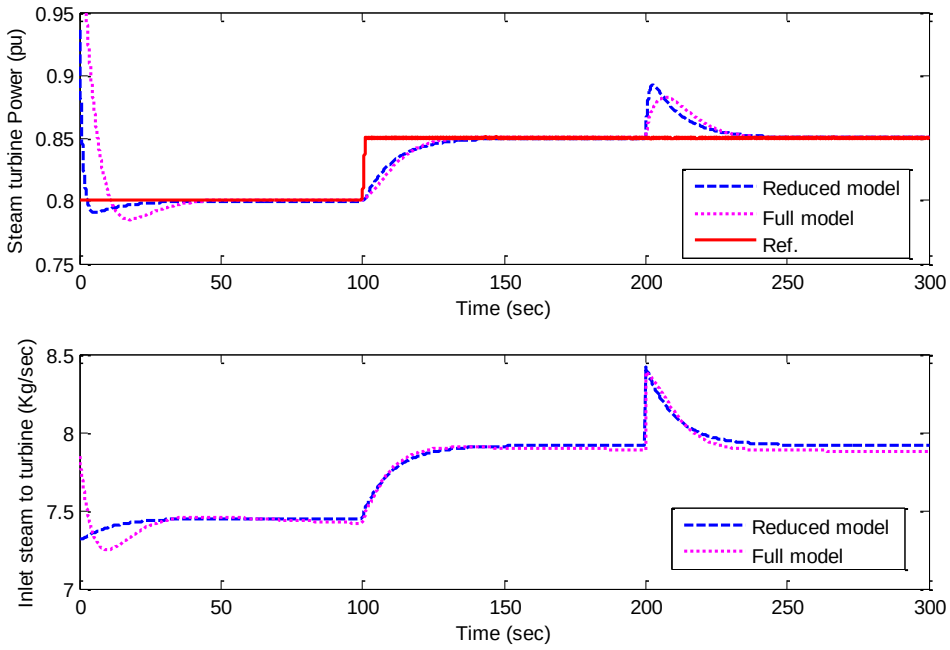


Figure 2.28: Comparison between reduced and complete steam turbine models

Figure 2.29 shows a block diagram of the reduced boiler model that includes the following interconnected subsystems: drum and riser model, furnace model, and superheater model.

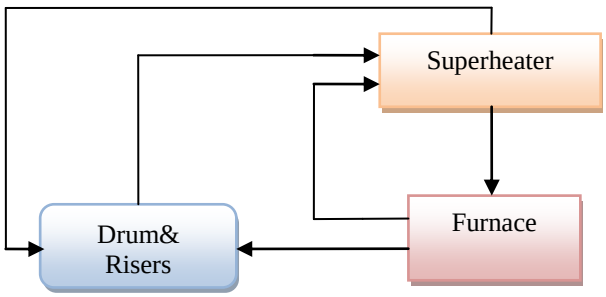


Figure 2.29: Block diagram of reduced boiler model

The boiler model is directly connected to the approximated steam turbine model with a generator model, creating a boiler-turbine model. This model can be connected to the gas turbine linear model to generate the complete CCPP model. The boiler-turbine model is represented by 13th order nonlinear first principle differential

equations. As shown in Figure 2.30, the boiler-turbine control system consists of four decentralized PI control loops, designed to meet the set-point requirements.

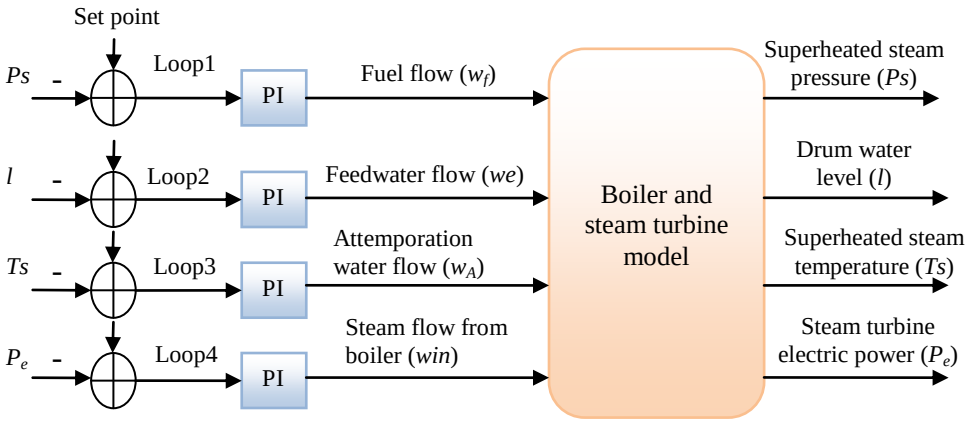


Figure 2.30: Boiler-turbine control system

2.6.1 Decentralized PID Tuning of the Simplified Boiler-turbine System

In this section, relay feedback and internal model control (IMC) tuning methods described in section 2.5.2 will be used to tune the PID parameters of boiler-turbine system.

2.6.1.1 Relay Feedback Tuning Results

The relay feedback method described in section 2.5.2.1 is applied to the MIMO decentralized PI controller for the boiler-turbine system. Starting with the second loop, the controller parameters converge in 3 iterations. Table 2.5 shows the results of the ultimate gain (K_{cu}) and the ultimate frequency (P_u) for the four loops.

Table 2.5: Ultimate gain and ultimate frequency for the four loops

	1-Pressure loop	2- Level loop	3-Temperature loop	4- Power loop
K_{cu}	19.6943	29.4413	1.1683	57.4955
P_u	37.65	126.15	2.25	0.75

The results of the PI tuning parameters using Tyreus-Luyben rules and Ziegler-Nichols rules are compared in Table 2.6.

Table 2.6: PI tuning results

PI settings $G_c = K_c + \frac{K_i}{s}$	Tyreus-Luyben rules				Ziegler-Nichols rules			
	Loop1	Loop2	Loop3	Loop4	Loop1	Loop2	Loop3	Loop4
K_c	6.1545	9.1268	-0.3651	17.9674	8.9520	13.382	-0.5311	26.1343
K_i	0.0743	0.03288	-0.0737	10.889	0.2853	0.127	-0.2832	41.8149

A comparison between these two methods to set-point changes are shown in Figure 2.31 to Figure 2.34. It is clearly shown in these figures that the Ziegler-Nichols tunings result is a very good disturbance rejection but gives an aggressive response compared to the Tyreus-Luyben method. However, in the water level loop, the controller response is very slow by using both tuning rules, as shown in Figure 2.32.

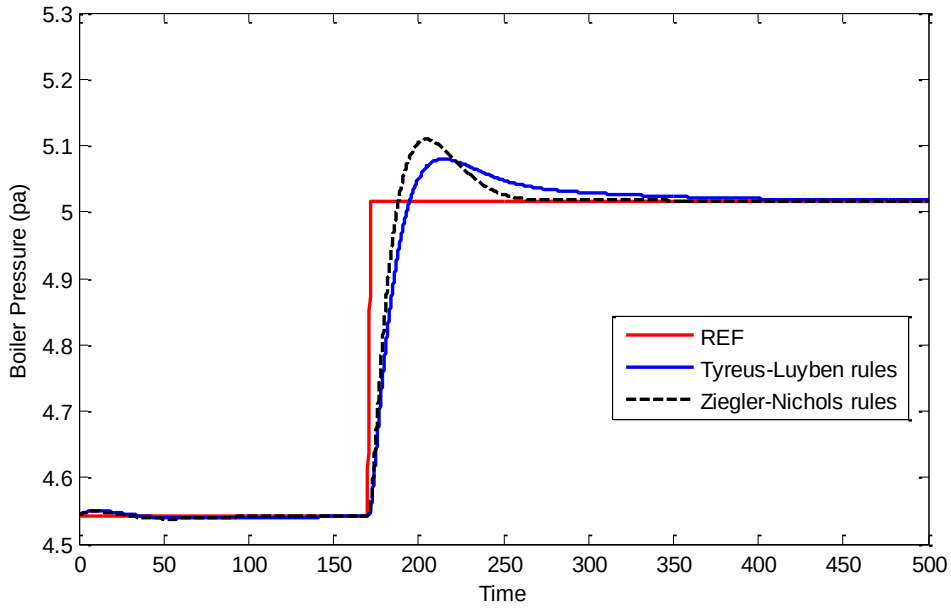


Figure 2.31: Ziegler and Tyreus response comparison (boiler pressure)

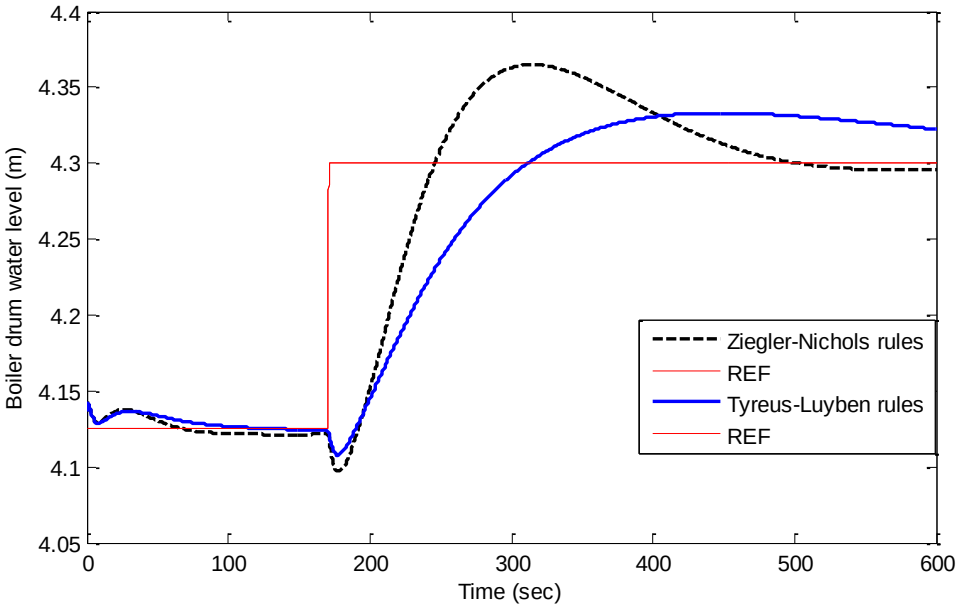


Figure 2.32: Ziegler and Tyreus response comparison (water level)

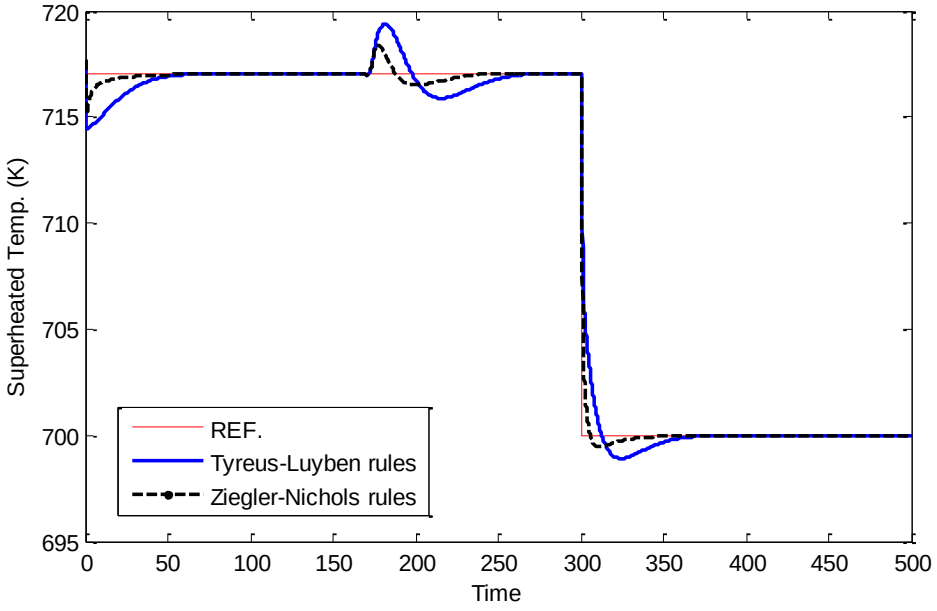


Figure 2.33: Ziegler and Tyreus response comparison (superheated temp.)

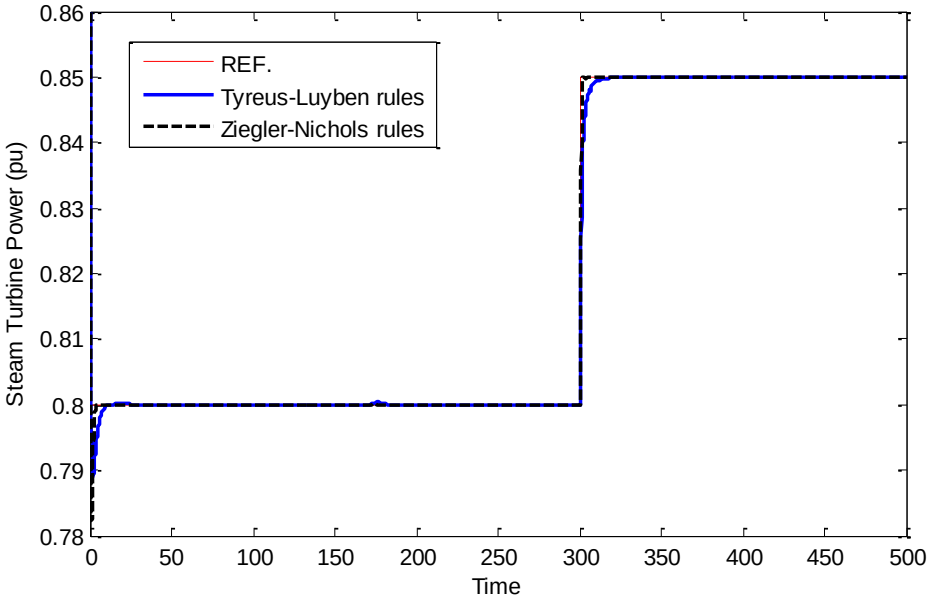


Figure 2.34: Ziegler and Tyreus response comparison (output power)

2.6.1.2 IMC Tuning Results

The Skogestad (2003) tuning method described in section 2.5.2.2 is used to obtain the controller parameters. Firstly, the MIMO transfer function of the process is found by linearization around an operating point. Then, the reduction functions in MATAB (Balreal and Modred) are used to reduce the diagonal elements of the process to a first order system. The process transfer function can be approximated by ignoring the interactions as follows:

Pressure loop:

The transfer function between fuel valve input and output pressure is:

$$G_{11} = \frac{0.000411}{s - 4.76 \times 10^{-5}} \tag{2.32}$$

Water level loop:

The transfer function between feed-water and water level in the boiler drum is:

$$G_{22} = \frac{0.0013}{s - 4.76 \times 10^{-5}} \tag{2.33}$$

Temperature loop:

The transfer function between the attemperator water valve and superheated temperature is:

$$G_{33} = \frac{-0.2964}{s + 0.006941} \tag{2.34}$$

Power loop:

The transfer function between the steam governor valve and output power is:

$$G_{44} = \frac{0.1191}{s + 1} \tag{2.35}$$

The optimal value of the tuning parameter λ is determined by a trade-off between fast response and good disturbance rejection by selecting a small value of λ or a slow response and better stability and robustness when selecting a large λ. In a power plant process, the very fast closed loop response is limited by the actuator saturation and the disturbance produced from the interactions between loops.

Note that it is not always possible to achieve stability using independent IMC_PI tuning design method due to omitting the interactions. Different papers have described methods for detuning the multi-loop controllers designed using the IMC-PI rules to take into account the interactions of the multivariable systems (Skogestad and Morari, 1989; Vu and Lee, 2010). However in this case the interactions have been considered by the careful selection of the tuning parameters for each of the loops based on the closed–loop performance.

The results of IMC and relay feedback (Tyreus rules) tuning parameters and performance are listed in Table 2.7. Comparisons between these two methods are shown in Figure 2.35, Figure 2.36, Figure 2.37, and Figure 2.38. As can be seen in these figures and Table 2.7, the advantage of the IMC method is that it can be tuned to provide good performances, comparable with real power plant (Ordys et al., 1994) and with settling time less than 120s in boiler system and 50s in steam turbine system.

Table 2.7: IMC and relay feedback tuning parameters and performance

	IMC				Relay feedback (Tyreus)			
	Loop1	Loop2	Loop3	Loop4	Loop1	Loop2	Loop3	Loop4
λ	10	65	7	10	-----	-----	-----	------
K_c	7.5131	23.697	-0.48197	0.8396	6.1545	9.1268	-0.36511	17.9674
K_i	0.1878	0.182	-0.0172	0.8396	0.0743	0.0329	-0.07376	10.889
Overshoot (%)	13.2	14	4.65	------	9	19.9	5.88	-----
Rise time (s)	14.7	26.3	16.8	22	20.3	89.9	7.11	1.9
Settling time(s)	108	111	106	39.1	177	748	49.1	5.06
Gm	inf	8.62	inf	inf	inf	16.9	inf	inf
Pm (deg)	76.6	50.6	121	90	82	58.8	127	99.7

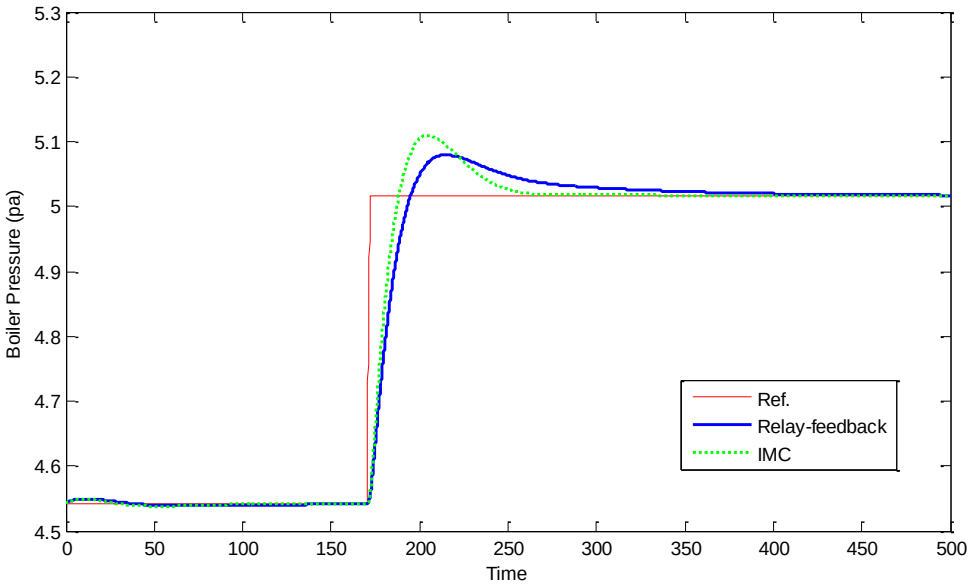


Figure 2.35: IMC and relay step response comparison (loop 1)

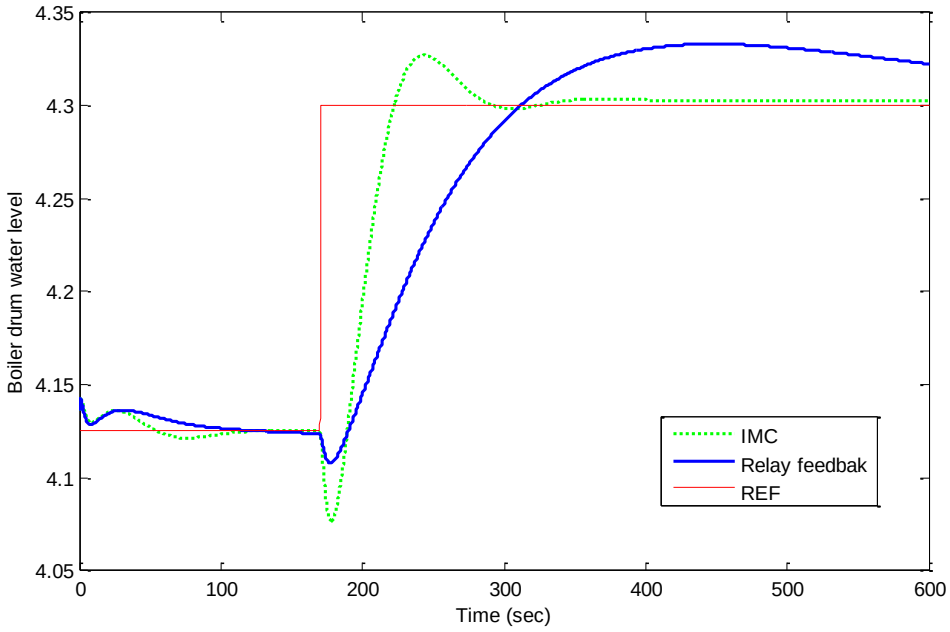


Figure 2.36: IMC and relay step response comparison (loop 2)

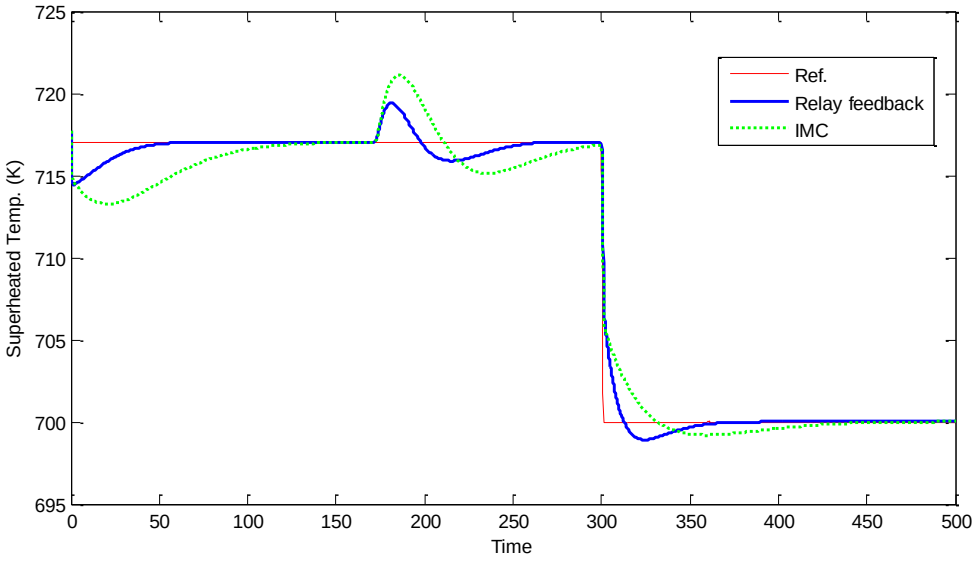


Figure 2.37: IMC and relay step response comparison (loop 3)

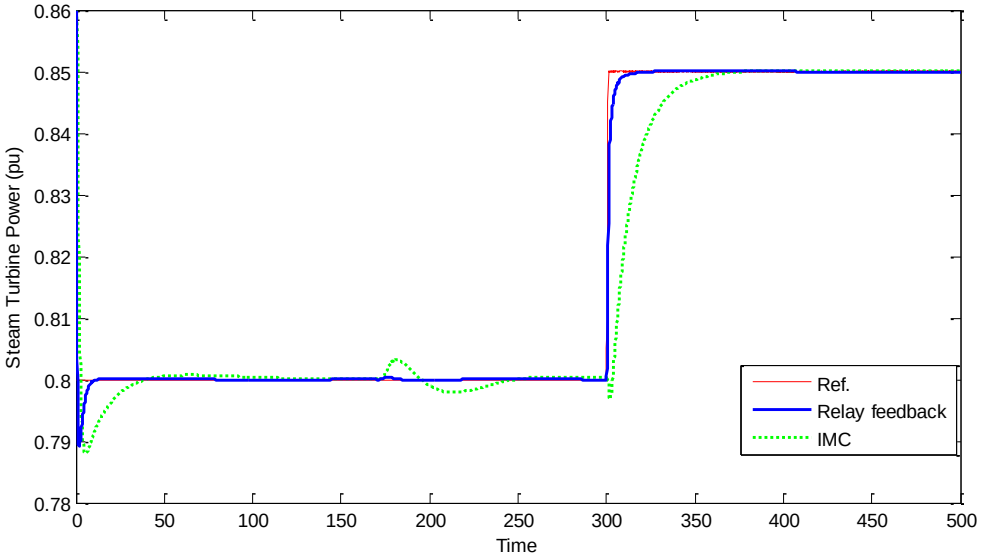


Figure 2.38: IMC and relay step response comparison (loop 4)

2.7 Conclusions

In this chapter, a non-linear CCPP model was developed in S-function MATLAB-SIMULINK environment. This model is based on the principles of mass, energy and momentum conservations that can capture the key dynamical properties over a wide operating range and are suitable for use in designing a model based controller. In this work, a new boiler model that captures much of the system dynamics, such as the shrink and swell phenomenon, was developed. Validation of the model was performed through comparison of the simulation results with the published data. Dynamic responses of the CCPP system with a PID controller to step disturbances, set-point changes and ramp changes in the load have been presented that prove the system stability with acceptable performance. In order to simplify the derivation of the non-linear state-dependent matrices of CCPP, a reduced CCPP model was presented that can capture much of the original model dynamics.

3. Supervisory Model Predictive Control Design for a Combined Cycle Power Plant

3.1 Introduction

The PID controller is traditionally used in power plant control systems. The need for energy efficiency and the tightening of environmental regulations have influenced the need for more advanced control strategies. Model based predictive control (MPC) has demonstrated its performance and gained much popularity in the process industries (Qin and Badgwell, 2003). This is mainly due to its ability to handle constraints.

LMPC refers to a family of MPC schemes in which linear or linearized models are used to predict system dynamics. LMPC is acceptable when the process operates at a single operating point and the controller is used only for disturbance rejection. NMPC techniques involve solving nonlinear differential equations and nonlinear dynamic optimization problems online that require high computational effort.

Since the appearance of the first contribution to model based predictive control in Richalet et al. (1978), many different forms have evolved, such as dynamic matrix control (DCM), generalized predictive control (GPC), and linear quadratic generalized predictive control (LQGPC) (Grimble and Ordys, 2001). The main differences between methods are seen in the type of model used and how the cost function is defined. The key features of MPC are as follows:

1. Multi-variable control
2. Constraints can be easily included
3. Can be used in either supervisory or primary control modes
4. Systems with a large time delay or unstable systems can be controlled

The basic MPC structure and approach are simply presented in Figure 3.1 and Figure 3.2 respectively. The basic concept behind all MPC formulation is to use a model of the system to predict the future output $\hat{y}(k+j|k)$ behaviour of the process over a horizon. The future control inputs $u(k+j)$ are then optimized such that the predicted response of the system has desirable features. This optimum solution is done by solving a quadratic cost function. The first input in the optimal sequence is applied to the system and the process is repeated for the next time step.

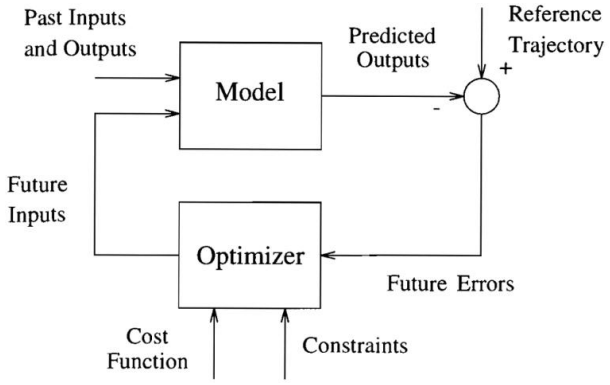


Figure 3.1: MPC basic structure

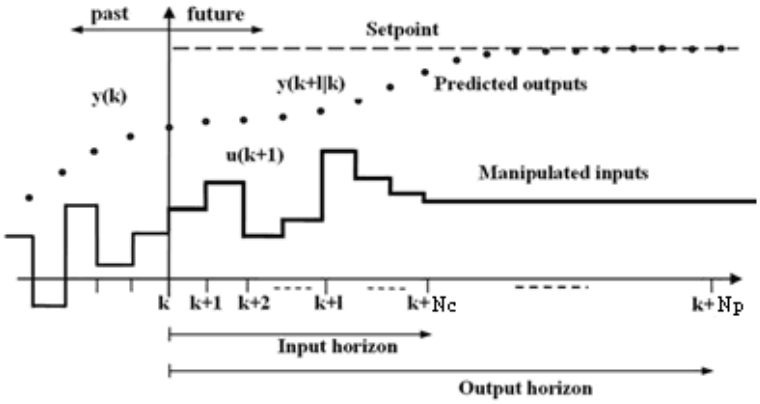


Figure 3.2: MPC approach

In the literature, there are some papers that deal with power plant control using MPC. For example, Katebi and Johnson (1997) described application of a decentralized predictive control scheme based on state space representation of generalized predictive control (GPC). Prasad et al. (2002) presented a non-linear MPC method to control a thermal power plant. This made use of successive linearization to obtain a linear model from a non-linear state space plant model. Extended Kalman filtering

(EKF) was used to estimate the state space model. The linear model was then used to formulate the predictive control routine. It also proposed a hierarchical MPC approach based on supervisory and regulatory levels. In it, the lower level PI loops help stabilize the unstable drum-boiler dynamic. The effectiveness of this method in disturbance rejection was reported. Saez et al. (2005) presented a supervisory model predictive control for a thermal power plant.

In this chapter, a supervisory predictive controller will be used to control the CCPP using a linearized state space model of the CCPP. The supervisory MPC provides the regulatory level set-points based on the objective function dynamic optimization. A Kalman filter is used to estimate the system state.

3.2 Control Strategy

The proposed control strategy consists of two levels: a regulatory PID level and a supervisory MPC optimization level. As discussed in Chapter 2, section 2.5.1, the supervisory level is used to provide optimal set-points for the regulatory level. The regulatory control layer manipulates and tracks the set-points, as shown in Figure 3.3.

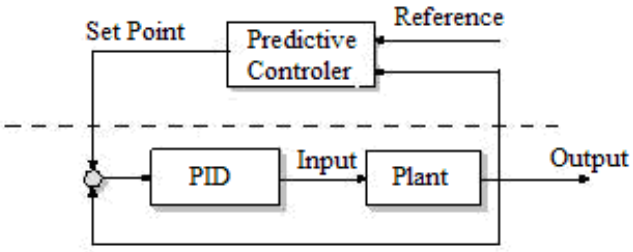


Figure 3.3: Supervisory control structure

3.3 Formulation of Model Predictive Control

In this section, a generalized MPC formulation based on linear discrete-time state space model will be described. The state space MPC multivariable formulation presented below is based on Wills (2004) and Maciejowski (2002).

3.3.1 Linearization

In order to design a model based control strategy, nonlinear systems can be approximated with equivalent linear systems. The linearized model must capture the main system dynamics. The nonlinear model can often be linearized by taking a Taylor's series approximation at the nominal operating point. For simple models this can be done analytically by hand or using symbolic math software. For complex systems, a linear model can be approximated by using the MATLAB\SIMULINK Control Design toolbox.

Given a system of nonlinear differential equations in continuous time:

$$\begin{aligned} \dot{x} &= f(x,u,t) \\ y &= g(x,u,t) \end{aligned} \tag{3.1}$$

where x, y and u represent the states, outputs and inputs respectively.

Making a Taylor expansion around an operating point yields an approximation of the nonlinear system(3.1):

$$\begin{aligned} \delta\dot{x} &= A\delta x + B\delta u \\ \delta y &= C\delta x + D\delta u \end{aligned} \tag{3.2}$$

where *A, B, C* and *D* are constant coefficient matrices. These matrices are defined as the Jacobians of the system, evaluated at the operating point as follows:

$$\begin{aligned} A &= \frac{\partial f(x_0,u_0)}{\partial x}; \qquad B = \frac{\partial f(x_0,u_0)}{\partial u} \\ C &= \frac{\partial g(x_0,u_0)}{\partial x}; \qquad D = \frac{\partial g(x_0,u_0)}{\partial u} \end{aligned} \tag{3.3}$$

where

$$\begin{aligned} \delta x &= x - x_0 \\ \delta u &= u - u_0 \\ \delta y &= y - y_0 \end{aligned} \tag{3.4}$$

x_0, u_0 and y_0 are the nominal operating point values.

In this thesis, in order to design a supervisory predictive controller, the complex non-linear model of CCPP, including the PI controllers, will be linearized by using the Graphic User Interface (GUI) SIMULINK Control Design toolbox.

3.3.2 Plant Model and Prediction

As the name suggests, model predictive control uses a model to make predictions which are then used for control purposes. The CCPP can be modelled using the linearized, discrete-time, state space system of system (3.1) as follows:

$$x(k+1) = Ax(k) + Bu(k) + \xi_k \tag{3.5}$$

$$y(k) = Cx(k) + \eta_k \tag{3.6}$$

where x, u and y are vectors of state variables, inputs and outputs respectively; A, B, and C are constant matrices; and ξ_k and η_k are the state and measurement noise assumed to be Gaussian distributed with zero mean.

Based on the state space model (A, B, C), the future values of the plant states and outputs over the prediction horizon may be obtained sequentially as follows:

$$\begin{aligned}
\hat{x}(k_i+1\,|\,k_i) &= A\hat{x}(k_i) + Bu(k_i) \\
\hat{x}(k_i+2\,|\,k_i) &= A\hat{x}(k_i+1\,|\,k) + Bu(k_i+1) \\
&= A^2x(k_i) + ABu(k_i) + Bu(k_i+1) \\
&\;\;\vdots \\
\hat{x}(k_i+N_p\,|\,k_i) &= A^{N_p-1}\hat{x}(k_i+1\,|\,k_i) + \sum_{j=1}^{N_p-1} A^{N_p-j-1}Bu(k_i+j\,|\,k)
\end{aligned} \tag{3.7}$$

The predicted output variables are calculated as follows:

$$\hat{y}(k_i+N_p\,|\,k_i) = C\hat{x}(k_i+N_p\,|\,k_i) \tag{3.8}$$

$$\hat{y}(k_i + N_p \mid k_i) = CA^{N_p-1}\hat{x}(k_i+1 \mid k_i) + C\sum_{j=1}^{N_p-1} A^{N_p-j-1}Bu(k_i + j \mid k_i) \tag{3.9}$$

For the time horizon from 1 to N_p (prediction horizon), the output predictor can be represented in a vector form as follows:

$$\hat{Y} = \Lambda\hat{x}(k_i) + \Phi U \tag{3.10}$$

where

$$\Lambda = \begin{bmatrix} CA \\ CA^2 \\ . \\ . \\ . \\ CA^{N_p} \end{bmatrix}; \qquad \Phi = \begin{bmatrix} CB & 0 & 0 & . & . & 0 \\ CAB & CB & 0 & . & . & 0 \\ CA^2B & CAB & CB & . & . & 0 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ CA^{N_p-1}B & CA^{N_p-2}B & CA^{N_p-3}B & . & . & CB \end{bmatrix}.$$

$$\hat{Y} = \begin{bmatrix} y(k_i+1 \mid k_i) \\ y(k_i+2 \mid k_i) \\ . \\ . \\ . \\ y(k_i + N_p \mid k_i) \end{bmatrix}; \qquad U = \begin{bmatrix} u(k_i) \\ u(k_i+1) \\ . \\ . \\ . \\ u(k_i + N_p - 1) \end{bmatrix}$$

3.3.3 Cost Function

The general aim of the control law is that the future output ($\hat{y}$) on the considered horizon should follow a determined reference signal (r) and, at the same time, the control effort (U) necessary for doing so should be penalized. The cost function to be minimized to obtain the optimal control input U_k at discrete-time k is of the following quadratic form:

$$J(U_k) = \frac{1}{2}\sum_{j=1}^{N_p} \| \hat{y}_{k+j|k} - r_{k+j} \|^2_Q + \sum_{j=1}^{Nc-1} \| u_{k+j|k} - u_{k+j-1|k} \|^2_S \quad (3.11)$$

where N_p and N_c are the prediction and control horizons respectively, Q is the weighting on the tracking error, and S is the weighting on the control increments. In this study, it is assumed that the control horizon is equal to the prediction horizon.

The summation terms in equation (3.11) can be expanded as follows (Wills, 2004):

$$\phi_z = \frac{1}{2}\sum_{j=1}^{N_p} \| \hat{y}_{k+j|k} - r_{k+j} \|^2_Q = \frac{1}{2} \| Y_k - R_k \|^2_{\bar{Q}} \quad (3.12)$$

Using equation(3.10), the objective function can be written as:

$$\phi_z = \frac{1}{2} \| \Lambda \hat{x}(k_i) + \Phi U_k - R_k \|^2_{\bar{Q}} \quad (3.13)$$

$$\phi_z = \frac{1}{2} \| \Phi U_k - (R_k - \Lambda \hat{x}(k_i)) \|^2_{\bar{Q}} \quad (3.14)$$

$$\phi_z = \frac{1}{2} \| \Phi U_k - b \|^2_{\bar{Q}} \qquad , b = R_k - \Lambda \hat{x}(k_i) \quad (3.15)$$

$$\phi_z = \frac{1}{2} (\Phi U_k - b)^T \bar{Q} (\Phi U_k - b) \quad (3.16)$$

$$\phi_z = \frac{1}{2} U_k^T \Phi^T \bar{Q} \Phi U_k - (\Phi^T \bar{Q} b)^T U_k + C_1 \quad (3.17)$$

This objective function can be expressed as a quadratic programming (QP) problem as follows:

$$\phi_z = \frac{1}{2} U_k^T H_z U_k + g_z^T U_k + C_1 \quad (3.18)$$

where

$$H_z = \Phi^T \bar{Q} \Phi \tag{3.19}$$

$$\begin{aligned} g_z &= -\Phi^T \bar{Q} b \\ &= -\Phi^T \bar{Q}(R_k - \Lambda \hat{x}(k_i)) \end{aligned} \tag{3.20}$$

$$R_k = \begin{bmatrix} r_{k+1} \\ . \\ . \\ r_{k+N_p} \end{bmatrix}, \qquad \bar{Q} = \begin{bmatrix} Q & & & \\ & Q & & \\ & & . & \\ & & & Q \end{bmatrix}, \qquad U_k = \begin{bmatrix} u_{k+1} \\ . \\ . \\ u_{k+N_p} \end{bmatrix}$$

C_1 is a constant term that can be discarded since it does not influence the solution of the problem.

The other term of the cost function is expressed as follows:

$$\phi_{\Delta u} = \frac{1}{2} \sum_{j=1}^{N_p - 1} \| u_{k+j|k} - u_{k+j-1|k} \|^2_S \tag{3.21}$$

$$\phi_{\Delta u} = \frac{1}{2} \sum_{j=1}^{N_p - 1} (u_{k+j|k} - u_{k+j-1|k})^T S (u_{k+j|k} - u_{k+j-1|k}) \tag{3.22}$$

Equation (3.22) can be expressed as follows:

$$\phi_{\Delta u} = \frac{1}{2} U_k^T \bar{S} U_k + (M_{u-1} u_k)^T U_k + C_2 \tag{3.23}$$

where

$$\bar{S} = \begin{bmatrix} 2S & -S & & & \\ -S & 2S & -S & & \\ & . & . & . & \\ & & -S & 2S & -S \\ & & & -S & S \end{bmatrix}, \quad M_{u-1} = -\begin{bmatrix} S \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

C_2 is a constant term that does not depend on U_k and can be discarded.

Combining equation (3.18) and equation (3.23) the objective function can be expressed as:

$$J(U_k) = \frac{1}{2} U_k^T H U_k + U_k^T f + C_3 \tag{3.24}$$

where

C_3 is the combination of C_1 and C_2 and may be safely ignored.

$$H = \Phi^T \bar{Q} \Phi + \bar{S} \tag{3.25}$$

$$f = -\Phi^T \bar{Q}(R_k - \Lambda \hat{x}(k_i)) + M_{u-1} u_k \tag{3.26}$$

This can be expressed as:

$$f = \Gamma \begin{bmatrix} \hat{x}_{k+1|k} \\ R_k \end{bmatrix} - \begin{bmatrix} Su_k \\ 0 \\ . \\ 0 \end{bmatrix} \tag{3.27}$$

$$\Gamma = \begin{bmatrix} \Phi^T \bar{Q} \Lambda & -\Phi^T \bar{Q} \end{bmatrix} \tag{3.28}$$

The optimal control input U_k can be found by solving the QP problem equation (3.24) with MATLAB's "Quadprog" function.

Applying the receding horizon control principle, only the first element of the vector U_k is used to obtain the control signal. The rest of the predicated control variables trajectory is discarded and at the next sample interval the procedure is repeated to find a new optimal solution for the control signal U_k.

3.3.4 MPC Formulation with Input Disturbance

The MPC problem will now be extended by adding measured disturbance. An example of measured disturbance is the feed rate from an upstream process, as will

be discussed in Chapter 5. If a disturbance variable is known or can be measured, it can be included in the state-space model as follows:

$$x(k+1) = Ax(k) + Bu(k) + Ed(k) \tag{3.29}$$

$$y(k) = Cx(k) \tag{3.30}$$

Where $d(k)$ and E are the measured disturbance and disturbance matrix respectively.

The output prediction can be calculated as in equation (3.7) as follows:

$$\begin{aligned}\hat{y}(k_i + N_p \mid k_i) = CA^{N_p-1}\hat{x}(k_i+1 \mid k_i) + C\sum_{j=1}^{N_p-1} A^{N_p-j-1}Bu(k_i + j \mid k_i) \\ + C\sum_{j=1}^{N_p-1} A^{N_p-j-1}Ed(k_i + j \mid k_i)\end{aligned} \tag{3.31}$$

The output predictor can be represented in a vector form as follows:

$$\hat{Y} = \Lambda\hat{x}(k_i) + \Phi U + \Phi_d D \tag{3.32}$$

Therefore, it is easy to see that including the disturbance term yields a new term equivalent to (ΦU). This new term is named ($\Phi_d D$).

where

$$\Phi_d = \begin{bmatrix} CE & 0 & 0 & . & . & 0 \\ CAE & CE & 0 & . & . & 0 \\ CA^2E & CAE & CE & . & . & 0 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ CA^{N_p-1}E & CA^{N_p-2}E & CA^{N_p-3}E & . & . & CE \end{bmatrix}, \quad D = \begin{bmatrix} d_k \\ . \\ . \\ d_{k+N_p-1} \end{bmatrix}.$$

The introduction of the $\Phi_d D$ term in $\hat{Y}$ means that equation (3.26) in the cost function has to be rewritten as follows:

$$f = -\Phi^T\bar{Q}(R_k - \Lambda\hat{x}(k_i) - \Phi_d D) + M_{u-1}u_k \tag{3.33}$$

3.3.5 Output Disturbance and Integral Action

In practical applications there are always modeling errors and disturbances present. The MPC formulation described above contains no explicit mechanism to deal with these complications. Therefore, these problems have to be considered. Usually, the controller is designed so that it contains integral action, which ensures zero steady-state error. There are several ways of including this integration in a state space model. All of them involve augmenting the state vector (Maciejowski, 2002).

In this study, the disturbance rejection will be done by augmenting the process model to include constant step output disturbances (Maciejowski, 2002; Muske and Badgwell, 2002). In this case, the difference between the predicted and the actual plant output is assumed to be caused by an output step disturbance, which remains constant in the future and can be estimated using a Kalman filter. The augmented state space system is given by:

$$\begin{bmatrix} x_{k+1} \\ d_{k+1} \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} x_k \\ d_k \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_k + \begin{bmatrix} \xi_k \\ \xi_k \end{bmatrix} \tag{3.34}$$

$$y_k = \begin{bmatrix} C & I \end{bmatrix} \begin{bmatrix} x_k \\ d_k \end{bmatrix} + \eta_k \tag{3.35}$$

For simplicity of notation, in the remaining part of this chapter the notation of the augmented state space model equations (3.34) and (3.35) is written as follows:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + \xi_k \\ y(k) &= Cx(k) + \eta_k \end{aligned} \tag{3.36}$$

3.3.6 State Estimation

The MPC algorithm for obtaining the optimal signal at each sample assumes that the present states are available. In practice, these states may not be available, so it should be estimated from measured outputs and known inputs. Thus, a state estimator is needed to estimate the states. For this purpose the Kalman filter is used to estimate the states of a discrete-time controlled process that is governed by a linear augmented

state space model as given by equation (3.36). ξ_k and η_k represent the process and measurement noises respectively, which are assumed to be zero mean Gaussian white noise with covariances Q_w and R_ω respectively.

The Kalman filter is used to estimate the state $\hat{x}_{k+1|k}$. The first phase of the Kalman filter is the time update or prediction phase. It is used to produce an estimate of the current state, using the estimate of the previous state:

$$\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1} + Bu_k \tag{3.37}$$

$$P_{k|k-1} = AP_{k-1|k-1}A^T + Q_\omega \tag{3.38}$$

The second phase of the filter is the measurement update or correction phase, where the measurements are used to get a more accurate estimate.

$$K_k = P_{k|k-1}C^T(CP_{k|k-1}C^T + R_\omega)^{-1} \tag{3.39}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - C\hat{x}_{k|k-1}) \tag{3.40}$$

$$P_{k|k} = (I - K_kC)P_{k|k-1}(I - K_kC)^T + K_kR_\omega K_k^T \tag{3.41}$$

The solution for Kalman gain K_k can be solved using the dlqe function in MATLAB.

3.3.7 Constrained MPC

The MPC will be extended such that it contains constraints. Constraints are present in all real-world processes. The process variables should stay within specified boundaries due to design requirements, physical constraints and safety requirements. Physical constraints are the result of physical limitations of the process equipment. For example, actuators limit, tank levels and valve slew rates. Safety considerations often impose limitations on plant outputs, such as temperature, heights in tanks, and pressure. Therefore, the MPC problem has to take the limits of the physical system into consideration. Violations of the constraints must not be allowed while the operation is kept close to these constraints.

One of the major advantages of MPC is its ability to handle constraints. The algorithm does this by optimising predicted performance subject to constraint satisfaction. There are three types of constraints, namely, input constraints, output constraints and state constraints. In this study, the optimization problem is assumed to be subject to constraints on the plant outputs.

3.3.7.1 Output Constraints

The output of a process can be limited by introducing upper and lower output constraints $y_{\max}$ and $y_{\min}$. Then the output constraints are specified as:

$$y_{\min} \le y_{k+j} \le y_{\max} \tag{3.42}$$

where

$$y_{\max} = [y_{1(k+j-1)\max}, ..., y_{m(k+j-1)\max}]^T \tag{3.43}$$

$$y_{\min} = [y_{1(k+j-1)\min}, ..., y_{m(k+j-1)\min}]^T \tag{3.44}$$

m is the number of outputs, j=1: N_p.

The output constraints are expressed in terms of U_k and using vector notations as follows (Maciejowski, 2002):

$$Y_{\min} \le \Lambda \hat{x}(k) + \Phi U_k \le Y_{\max} \tag{3.45}$$

where

$$Y_{\max} = [Y_{1\max}^T, ..., Y_{m\max}^T]^T \tag{3.46}$$

$$Y_{\min} = [Y_{1\min}^T, ..., Y_{m\min}^T]^T \tag{3.47}$$

Finally, the model predictive control in the presence of hard constraints is proposed as finding the parameter vector U_k that minimizes the cost function:

$$J(U_k) = \frac{1}{2} U_k^T H U_k + U_k^T f \tag{3.48}$$

Subject to the inequality constraints:

$$A_c U_k \le B_c \tag{3.49}$$

where

$$A_c = \begin{bmatrix} -\Phi \\ \Phi \end{bmatrix} \tag{3.50}$$

$$B_c = \begin{bmatrix} -Y_{\min} + \Lambda \hat{x}(k) \\ Y_{\max} - \Lambda \hat{x}(k) \end{bmatrix} \tag{3.51}$$

This is a standard QP optimization problem that can be solved using MATLAB's Quadprog function.

3.3.7.2 Soft Output Constraints

Predictive control optimization problems may become non-feasible in the presence of constraints. This is mainly due to model-plant mismatches, disturbances or noise. A simple solution to this problem is to enlarge the horizons. However, this will increase the computational burden and will not solve all sorts of infeasibilities (Afonso and Galvão, 2012). In MPC applications, soft constraints are often used on system states and outputs. Softening constraints is the process of removing the original hard constraints and adding a penalty function to the objective function (de Oliveira and Biegler, 1994; Zheng and Morari, 1995).

The objective function for soft constraints to obtain U_k at any time *k* can be written as follows (Scokaert and Rawlings, 1999):

$$J(U_k, \Psi) = \frac{1}{2}\sum_{j=1}^{N_p} \| \hat{y}_{k+j|k} - r_{k+j} \|^2_Q + \frac{1}{2}\sum_{j=1}^{N_p-1} \| u_{k+j|k} - u_{k+j-1|k} \|^2_S + \frac{1}{2}\|\psi_k\|^2_{S_\psi} + s_\psi^T \psi_k \tag{3.52}$$

subject to:

$$\begin{aligned} -\infty < y_k - \psi_k &\le y_{\max} \\ y_{\min} \le y_k + \psi_k &\le \infty \\ 0 \le \psi_k &< \infty \end{aligned}$$

The added terms in the cost are the square of the maximum violation ψ_k over the horizon, weighted by a constant S_ψ and linear measures of constraints violations $s_\psi^T \psi_k$ term. Tuning S_ψ and s_ψ adjusts the relative importance of the two terms in the cost. Setting S_ψ to zero effectively removes the output constraints and increasing S_ψ leads to smaller violations or ever more hardening constraint.

The new term in the objective function can be written in QP formulation as follows (Sørensen and Kristiansen 2007):

$$\phi_\psi = \frac{1}{2}\|\psi_k\|^2_{S_\psi} + s_\psi^T \psi_k = \frac{1}{2}\Psi^T \bar{S}_\psi \Psi + s_\psi^T \Psi \tag{3.53}$$

where

$$\bar{S}_\psi = \begin{bmatrix} \bar{S}_\psi & & \\ & \ddots & \\ & & \bar{S}_\psi \end{bmatrix}, \bar{s}_\psi = \begin{bmatrix} s_\psi \\ \vdots \\ s_\psi \end{bmatrix}, \Psi = \begin{bmatrix} \psi_1 \\ \vdots \\ \psi_{N_p} \end{bmatrix}$$

The final objective function can be written in QP as follows:

$$J(U_k, \Psi) = \frac{1}{2}\bar{U}_k^T \bar{H} \bar{U}_k + \bar{f}^T \bar{U}_k \tag{3.54}$$

subject to

$$b_{\min} \le A U_k \le b_{\max} \tag{3.55}$$

where

$$\bar{H} = \begin{bmatrix} H & 0 \\ 0 & \bar{S}_\psi \end{bmatrix}, \ \bar{f} = \begin{bmatrix} f \\ \bar{s}_\psi \end{bmatrix}, \ \bar{U}_k = \begin{bmatrix} U_k \\ \Psi \end{bmatrix}$$

$$b_{\min} = \begin{bmatrix} -\infty \\ Y_{\min} \end{bmatrix}, b_{\max} = \begin{bmatrix} Y_{\max} \\ \infty \end{bmatrix}, A = \begin{bmatrix} \Phi & -I \\ \Phi & I \end{bmatrix}$$

3.3.8 MPC Tuning

The tuning parameters of the MPC controller are the cost function weighting matrices Q and S, the control horizon N_c and the prediction horizon N_p. Garriga and Soroush (2010) presented a review of the available tuning guidelines for MPC. Despite some progress made so far, MPC tuning is still performed by running several simulations to check if the chosen tuning parameters are suitable (Shah and Engell, 2011). The choice of these parameters has an effect on the nominal stability, robustness, and controller performance of the MPC algorithms. The effect of tuning parameters on control performance can be summarized as follows (Garriga and Soroush, 2010):

1- The prediction horizon N_p: A long prediction horizon leads to better performance and less aggressive control action and has a stabilizing effect, although it increases the computation burden.

2- The control horizon N_c: A larger control horizon leads to a more optimal and aggressive controller. In this case, the system response is faster and more sensitive to disturbances. In addition, long N_c leads to unnecessary control action and long computation time.

3- The weighting matrices Q and S: The diagonal matrix Q penalises the tracking errors and guides the servo performance of the control system, the most important variables having the largest weights and less deviation from setpoint. The diagonal matrix S penalizes the movement of manipulated variables. Increasing the values of weight S relative to weight Q has the effect of reducing the control activity.

3.3.9 MPC and Stability

In general, the MPC problem is formulated as solving on-line a finite horizon open-loop optimal control problem subject to constraints. However, in this general form, the stability of MPC control is not guaranteed (Bithmead et al., 1991). This is because finite horizon causes deviation between open-loop prediction and the closed-

loop system. Closed-loop stability can be ensured by suitable tuning of MPC parameters and by selecting the prediction horizon sufficiently long (Maciejowski, 2002). This method is still widely used in industry to ensure the stability of classical MPC algorithms (Chen, 2010).

Several ways of guaranteeing nominal stability are investigated (Mayne et al., 2000). The most widely used method is the addition of terminal cost and constraints to explicitly ensure stability. The main idea is to evaluate the cost function over an infinite prediction horizon as follows:

$$J(k) = \sum_{j=0}^{\infty} [x_{k+j|k}^T \, Q \, x_{k+j|k} + u_{k+j|k}^T \, R \, u_{k+j|k}] \tag{3.56}$$

where $Q = Q^T \geq 0, \quad R > 0$ are weighting matrices of state and control input.

The solution to an infinite horizon is computationally intractable, therefore the practical way of solving this problem is through the dual mode prediction scheme, where the infinite horizon is divided into two sub-problems (Michalska and Mayne, 1993).

$$J(x,u) = \sum_{j=0}^{N_p-1} [x_{k+j|k}^T \, Q \, x_{k+j|k} + u_{k+j|k}^T \, R \, u_{k+j|k}] + \sum_{j=N_p}^{\infty} [x_{k+j|k}^T \, Q \, x_{k+j|k} + u_{k+j|k}^T \, R \, u_{k+j|k}] \tag{3.57}$$

The infinite horizon quadratic cost is given by:

$$\sum_{j=N_p}^{\infty} [x_{k+j|k}^T \, Q \, x_{k+j|k} + u_{k+j|k}^T \, R \, u_{k+j|k}] = x_{k+N_p|k}^T \, P \, x_{k+N_p|k} \tag{3.58}$$

Therefore, the cost function can be written as follows:

$$J(x,u) = x_{k+N_p|k}^T \, P \, x_{k+N_p|k} + \sum_{j=0}^{N_p-1} [x_{k+j|k}^T \, Q \, x_{k+j|k} + u_{k+j|k}^T \, R \, u_{k+j|k}] \tag{3.59}$$

where the terminal weight *P* matrix is calculated as a solution to a fake algebraic Riccati equation (Maciejowski, 2002):

$$P = A^T PA - A^T PB(B^T PB + R)^{-1} B^T PA + Q \tag{3.60}$$

In mode 2, for $j \geq N_p$, the optimization solution will yield the unconstrained linear quadratic regulator (LQR), where the optimal control is given by:

$$u(k+j) = -Kx(k+j) \tag{3.61}$$

$$K = (B^T PB + R)^{-1} B^T PA \tag{3.62}$$

Here mode 1 refers to an initial horizon of $j < N_p$, where optimal control is obtained by solving the constrained MPC optimization problem, as defined in the second term of equation (3.59).

Another possibility for proving stability of linear MPC is to add a terminal constraint at the end of the horizon, which forces the state to take a particular value. This will result in stable closed loop when the optimization problem has a feasible solution. However, adding a terminal constraint is quite restrictive (Maciejowski, 2002).

3.3.10 MPC Algorithm

The basic MPC law is described by the following algorithm:

1. Define output constraints.
2. Calculate Kalman filter gain equation (3.39).
3. Get input /output measurements and update states equation (3.40).
4. Compute output prediction equation (3.32).
5. Solve the optimization problem equation (3.54) .
6. Apply first element of control input U_k and shift optimal horizon one step forward.
7. Let k=k+1 and go to 3.

3.4 Simulation and Results

In this simulation, a constraint supervisory MPC controller presented in section 3.3.10 was used to control the CCPP simulator that was developed in Chapter 2. As shown in Figure 3.4, the supervisory MPC was used to provide optimal set-points for the regulatory level using the linearized state space model of the CCPP system. A Kalman filter observer was used to estimate the system states. In this study, the closed -loop stability is achieved by a suitable tuning of MPC design parameters.

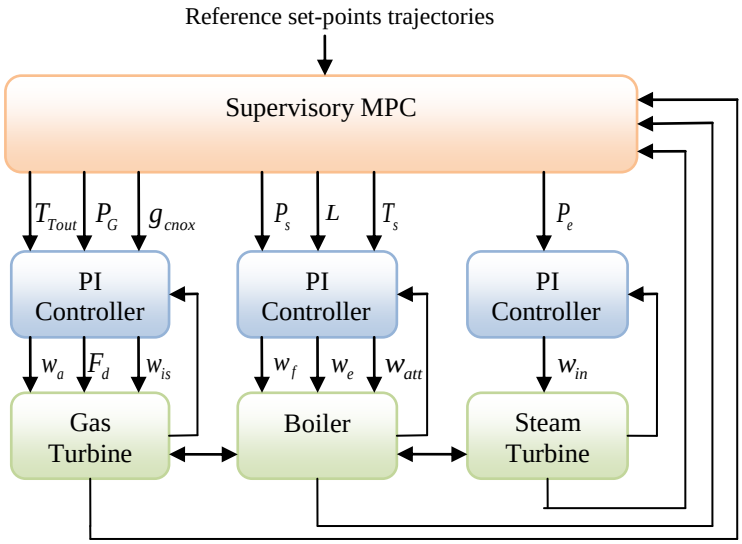


Figure 3.4: Supervisory MPC

3.4.1 Linearization of CCPP System

The nonlinear model of CCPP, including the PI controllers, was linearized into a state space model at a given operating point using the SIMULINK control design program. In this simulation, it was assumed that the load operating points were 80%, 90%, 110% and 120% of the nominal operating (100%), as listed in Table 3.1. These different operating points are used to prove that the linearized model cannot cover wide range of operating points. The linearized models obtained using the above operating points had 37 states, which was reduced to an acceptable 30th order model using BALREAL and MODRED MATLAB functions.

Table 3.1: Operating points

	80%	90%	100%	110%	120%
Boiler pressure (pa)	3.62×10^6	4.072×10^6	4.525×10^6	4.977×10^6	5.43×10^6
Drum level (m)	3.314	3.728	4.142	4.556	5.05
Boiler temp. (K)	524	598.5	655	720	786
Gas turbine temp.(K)	814.4	916.2	1018	1118	1218
Gas turbine power-pu	0.6	0.7	0.8	0.9	1.0
NOx contents (ppm)	72.608	81.684	90.76	99.76	108
Steam turbine power	0.6	0.7	0.8	0.7	0.6

To validate linearization results, sinusoidal input signals were injected in both the nonlinear model at the nominal operating point (100%) and all linear models that linearized using different operating points (80%, 90%, 100%,110% and 120%). This comparison test was made by injection of sinusoidal input signals with +10% amplitude and frequency of $\omega = 0.05\ rad/\sec$ in the boiler superheated pressure set-point, as shown in Figure 3.5. The accuracy of the linearization is clearly shown in this figure.

The linearization results were also validated by comparing the simulation outputs of the linear models and the nonlinear model using the same step input signals. Figure 3.6 shows the boiler superheated pressure responses to +10% step changes in the set-point. As shown in this figure, both the nonlinear and linear models reached steady state at the same time but with very small differences in the transient response due to the nonlinearity of the boiler system.

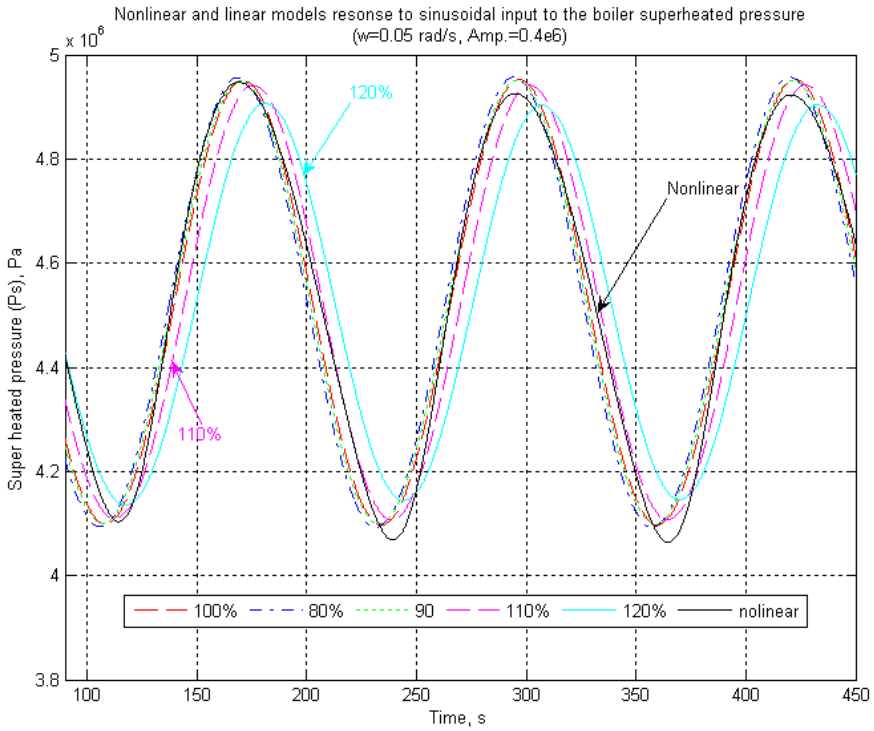


Figure 3.5: Nonlinear and linearized models' responses to sinusoidal inputs

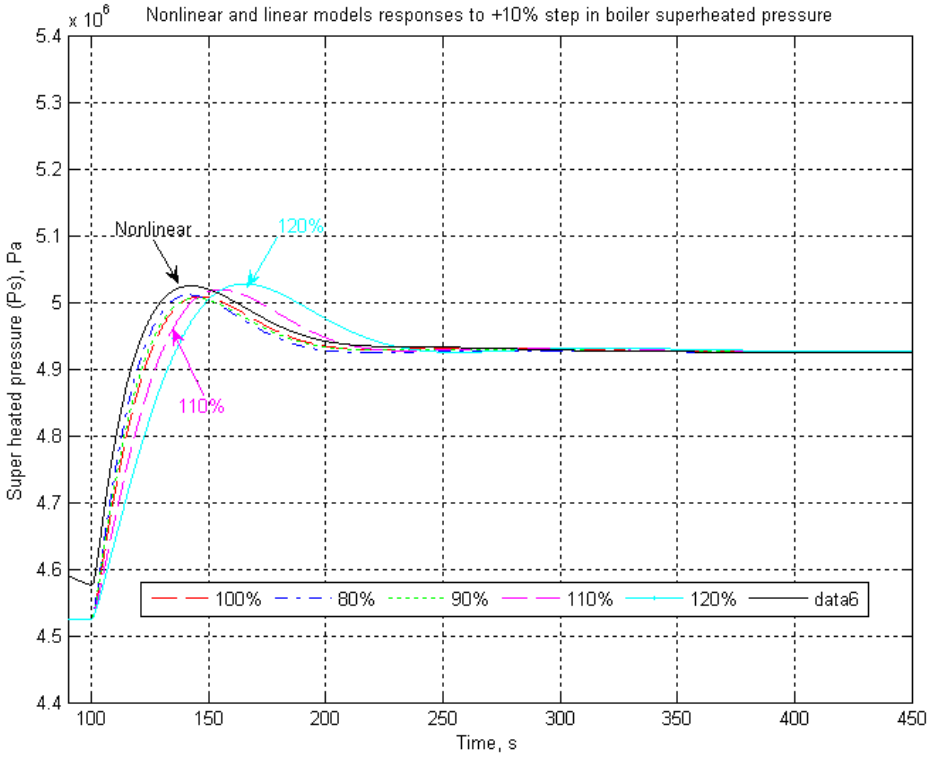


Figure 3.6: Nonlinear and linear models' responses to step inputs

3.4.2 Supervisory MPC Results

In this simulation, the nonlinear CCPP states were estimated using the Kalman filter; the prediction horizon was selected as N_p= 80. The control horizon was assumed to be equal to the prediction horizon. The discretization time was chosen as T=1 sec. The weighting matrices Q and S were chosen first arbitrarily and then the role of each parameter was retuned using simulations. Q and S were chosen as follows:

S= diag [$7.11\text{x}10^{-5}$ 5000 200 0.6517 55800 300 50000],

Q=diag [$1\text{x}10^{-6}$ 5000 150 1.018 89800 250 80000]

The introduced maximum and minimum output constraints are listed in Table .

Table 3.2: Output constraints

Output constraints	Minimum	Maximum
Superheated boiler pressure (Pa)	4.4×10^{6}	4.65×10^{6}
Boiler drum level (m)	3.5	4.5
Steam turbine power (pu)	0.62	1

Figure 3.7 shows the responses of the boiler drum level and steam turbine power to +10% step changes in boiler superheated pressure using constrained and unconstrained MPC. It can be observed that the imposed output constraints are satisfied using the constrained MPC.

Simulation results comparing the performance of supervisory MPC and the classical PID controllers are shown in Figure 3.8 and Figure 3.9. These figures show that MPC is able to reach the setpoint faster than the PID, which continuously oscillates around the setpoint with a larger overshoot. Figure 3.8 also demonstrates how the constrained MPC keeps the output pressure level within the specified bounds, which is violated when using the PID controller.

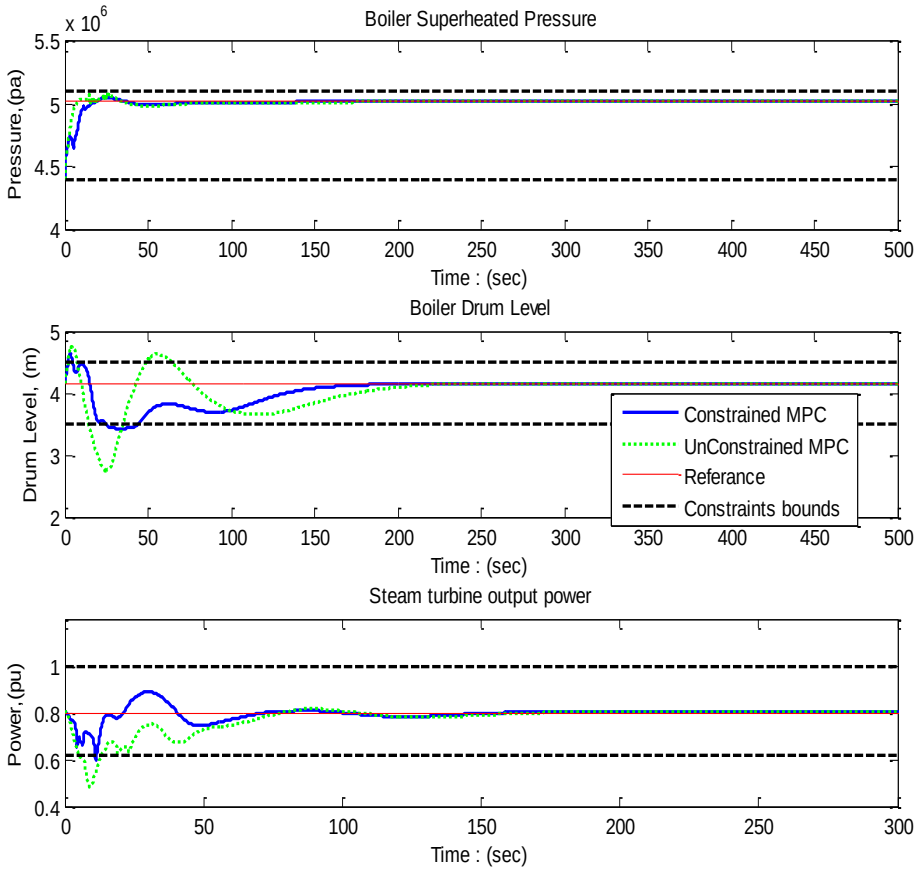


Figure 3.7: Boiler drum level and power of steam turbine response using constrained and unconstrained MPC

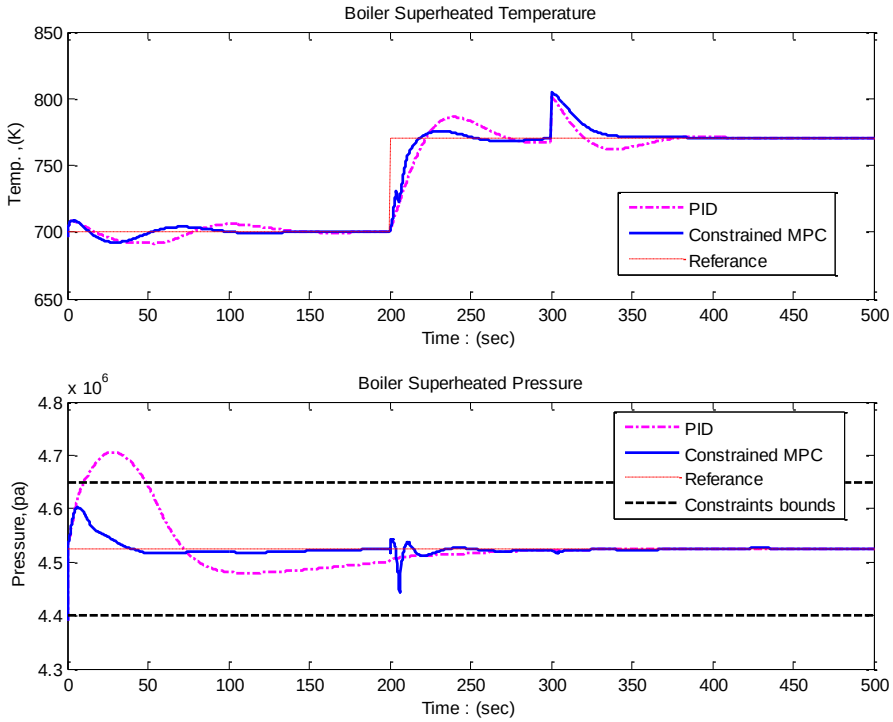


Figure 3.8: Boiler pressure response to +10% setpoint changes and 5% output disturbance in boiler temperature at 300 seconds

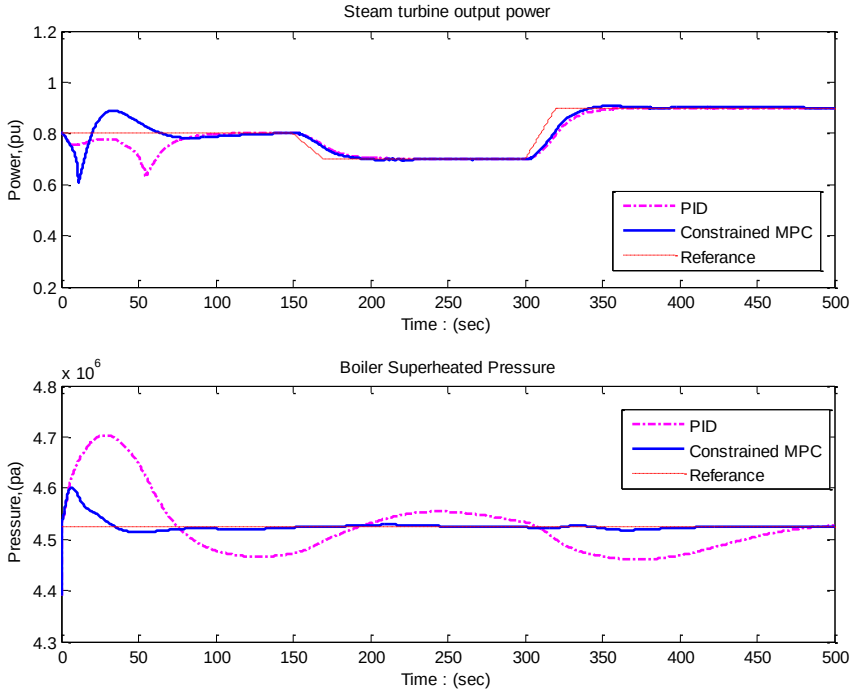


Figure 3.9: Boiler pressure response to steam turbine load change

3.5 Conclusions

This chapter proposed the use of a second layer of control based on MPC to tune the performance of PID controllers of the CCPP. This has the advantage of including output constraints to provide safety limitations and satisfy environmental regulations. Simulation results showed that the supervisory MPC has better performance than classical PID control schemes and allows handling constraints.

In linear MPC, a linear model is used to predict system dynamics, which has the advantage of solving convex-optimization problems. However, if the process is operated over a wide range of conditions, a fixed model linearized at a steady-state is usually not sufficient to have the predictability to cover the operating range. Therefore, in the next chapter, NMPC will be considered to control the CCPP, where this linear MPC algorithm will be used as a basis for the development of LTV state-dependent NMPC.

4. Nonlinear Predictive Control and Estimation for an Industrial Power Plant State-Dependent Model

4.1 Introduction

The major dynamics of power plants include nonlinear behaviour, time delays and uncertainties. Nowadays, power generation processes need to operate under much tighter performance specifications. For example, economic, environmental and safety constraints need to be satisfied. These increasing demands can only be met by using advanced control strategies.

MPC technology has made a significant impact on industrial control engineering, due to its ability to handle constraints. As discussed in Chapter 3, LMPC refers to a family of MPC schemes in which linear or linearized models are used to predict system dynamics. LMPC is acceptable when the process operates at a single operating point and the controller is only used for disturbance rejection.

Nowadays, the demand for rapid changes in power generation requires frequent changes from one operation point to another, and often near the boundary of the admissible region. Under these conditions, the results obtained by using an LMPC are poor in terms of performance, and often not sufficient for coping with the process requirements (Findeisen and Allgöwer, 2002). This suggests the need for nonlinear control strategies, which are based directly on the nonlinear model and which explicitly take account of the nonlinearities during the system synthesis process.

In contrast to linear MPC, where convex quadratic programs are mostly solved exactly at each sampling time, the solution to the NMPC optimization problem requires solving a nonconvex, nonlinear problem online, which is, in general, computationally expensive and increases significantly with the complexity of the

model used in the controller. Moreover, the global solution for optimization cannot be guaranteed (Camacho and Bordons, 2007).

In this chapter, a supervisory constrained NMPC strategy based on the LTV state-dependent model is formulated and applied for the regulation of a nonlinear power plant to reduce the complexity of online optimization problems. The main difference with the previous work in this area is that the complete algorithm, using an optimization procedure, a nonlinear estimator and constraints, is employed to design a 2nd level controller to generate optimal set points for 1st level regulating PID loops. NMPC requires full state information for the prediction.

4.2 State-Dependent Riccati Equation Control Method

Based on the state dependent state-space model, Cloutier (1997) developed a nonlinear regulator technique called the State Dependent Riccati Equation (SDRE). This technique approaches the problem by mimicking the LQR formulation for linear systems. It has been applied successfully to nonlinear process systems, both in theory and in experimental practice (Banks et al., 2007; Çimen, 2012). The quadratic infinite horizon cost function is in the form:

$$Minimize \quad J = \frac{1}{2}\int_{t_0}^{\infty} x^T Q_w x + u^T R_w u \, dt \tag{4.1}$$

with the state $x \in \Re^n$ and control $u \in \Re^m$, subject to the nonlinear system constraints

$$\begin{aligned} \dot{x} &= f(x) + B(x)u \\ y &= C(x)x \end{aligned} \tag{4.2}$$

where $B \in \Re^{n\times m}$ and C are the system input and output matrices. $Q_w \in \Re^{n\times n}$ and $R_w \in \Re^{m\times m}$ are state dependent weighting matrices which satisfy $Q_w \geq 0$ and $R_w > 0$ for $\forall x$.

Direct parameterization is used to transform the nonlinear dynamics to the state-dependent coefficient (SDC) form:

$$\dot{x} = A(x)x + B(x)u \tag{4.3}$$

with $f(x) = A(x)x$, where $A \in \Re^{n \times n}$ is the state matrix.

Based on this linear-like state representation, the SDRE approach is almost the same as the standard LQR formulation for linear systems. However, for the SDRE approach, the system model matrices $A(x)$ and $B(x)$ are updated at every time-step because they depend on the state. Then, the optimal feedback control u is obtained, as follows:

$$u = -K(x)x = -R^{-1}B^{T}(x)P(x)x \tag{4.4}$$

where P(x) is the symmetric, positive-definite solution of the state-dependent Algebraic Riccati Equation (ARE):

$$A^{T}(x)P(x) + P(x)A(x) - P(x)B(x)R_{w}^{-1}B^{T}(x)P(x) + Q_{w} = 0 \tag{4.5}$$

To ensure the desired solution of equation(4.5), the global controllability and observability of state-dependent system factorizations is commonly assumed.

4.3 State-Dependent Coefficient (SDC) Representation

Cloutier et al. (1996) have shown that the SDC form (4.3) can always be obtained for most nonlinear dynamic systems, using simple algebraic manipulations if $f(0) = 0$ and $f(x)$ are continuously differentiable, that is $f(x) \in C^{1}$. In the multivariable case, it is stated in (Cloutier et al., 1996) that the SDC parameterizations (4.3) are not unique. In fact, there are an infinite number of ways to bring the nonlinear system (4.2) to SDC form. This is true, provided that at least two parameterizations exist for (4.2). Let $A_1(x)$ and $A_2(x)$ be two distinct SDC parameterizations, then

$f(x) = A_1(x)x = A_2(x)x$. Consider the SDC matrix $A(x,\alpha) = \alpha A_1(x) + (1-\alpha)A_2(x)$, so:

$$\begin{aligned} A(x,\alpha)x &= \alpha A_1(x)x + (1-\alpha)A_2(x)x \\ &= \alpha f(x) + (1-\alpha) f(x) \\ &= f(x) \end{aligned} \tag{4.6}$$

Therefore, $A(x,\alpha)$ is a valid SDC parameterization for all $\alpha \in \Re$, so that there are an infinite number of parameterizations corresponding to the choice of α. The non uniqueness of the SDC parameterization for multivariable systems creates additional degrees of freedom, which can be utilized to provide great design flexibility (Cloutier, 1997). The SDC parameterization must be chosen in accordance with control system and state estimation design objectives. An important factor for this choice is not violating the controllability and observability of the system. The local stability of the closed loop system resulting from using the SDRE nonlinear regulation technique is given by the following theorem from Cloutier et al. (1996).

Theorem 4.3.1: Cloutier et al. (1996) assumed that the SDC parameterization is chosen, such that $f(x) \in C^1$, $\{C(x), A(x)\}$ is globally observable (or at least detectable) and $\{A(x), B(x)\}$ is globally controllable (stabilizable). Then, the SDRE nonlinear regulation control method has a closed loop solution which is locally asymptotically stable. The proof can be found in (Cloutier et al., 1996).

As in linear theory, the controllability matrix is given by:

$$M(x) = \begin{bmatrix} B(x) & A(x)B(x) & \cdots & A^{(n-1)}(x)B(x) \end{bmatrix} \tag{4.7}$$

The state-dependent observability matrix is given by:

$$O(x) = \begin{bmatrix} C(x) \\ C(x)A(x) \\ \vdots \\ C(x)A^{n-1}(x) \end{bmatrix} \tag{4.8}$$

If $M(x)$ and $O(x)$ have full rank for all x, then the system is controllable and observable. This will only guarantee local stability. The main drawback of this method is that it does not provide assurance of global asymptotic stability, which is a difficult issue in general for non-linear systems. In the literature, the stability analysis of the SDRE method either sacrifices performance or imposes conservative rules (Çimen, 2012). However, most SDRE controllers are simply implemented by satisfying theorem 4.3.1, and then simulation analysis is used to assess stability (Çimen, 2012). The other drawback of the SDRE method is that it does not explicitly handle constraints.

4.4 Nonlinear Model Based Predictive Control

In this section, NMPC methods that are based on LTV systems will be first discussed, and then NMPC strategy based on the LTV state-dependent model will be used to control the nonlinear power plant process.

4.4.1 Successive Linearization

Due to computational limitations, nonlinear MPC technology has not yet been used at a large industrial scale. As an alternative to a fully nonlinear MPC, a different MPC algorithm which uses linear time-varying predictions through local Jacobian linearization has been proposed in a variety of papers (Lee et al., 2002; Kouvaritakis et al., 1999).

In this method, the nonlinear model dynamics are linearized at each iteration, around the predicted trajectories for the state and control variables obtained from the previous iteration. The resulting linear time varying system is used in the receding horizon optimization, using the QP problem. Extended Kalman filtering (EKF) can be used to estimate the current states from the noisy outputs. This strategy was used in controlling industrial nonlinear processes, such as in Prasad et al. (2002).

To approximate the nonlinear model using the LTV system, the nonlinear model (4.2) is linearized around the trajectory vector $X_{k+1,}$ as follows:

$$\delta x_{K+1} = A(\bar{x}_k, \bar{u}_k)\delta x + B(\bar{x}_k, \bar{u}_k)\delta u_k \tag{4.9}$$

$$\delta y_{K+1} = C(\bar{x}_k, \bar{u}_k)\delta x + D(\bar{x}_k, \bar{u}_k)\delta u_k \tag{4.10}$$

where:

$$A(\bar{x}_k, \bar{u}_k) = \frac{\partial f(x_k, u_k)}{\partial x_k}|_{x_k, u_k} \quad , \quad B(\bar{x}_k, \bar{u}_k) = \frac{\partial f(x_k, u_k)}{\partial u_k}|_{x_k, u_k}$$

$$C(\bar{x}_k, \bar{u}_k) = \frac{\partial g(x_k, u_k)}{\partial x_k}|_{x_k, u_k} \quad , \quad D(\bar{x}_k, \bar{u}_k) = \frac{\partial g(x_k, u_k)}{\partial u_k}|_{x_k, u_k}$$

The states, outputs and inputs are calculated from equations (4.9) and (4.10), as follows:

$$\begin{aligned} x_k &= \bar{x}_k + \delta x_k \\ y_k &= \bar{y}_k + \delta y_k \\ u_k &= \bar{u}_k + \delta u_k \end{aligned} \tag{4.11}$$

For highly non-linear systems, the nonlinear model is weakly approximated through the Taylor series expansion. This may lead to model mismatch and instability.

4.4.2 Linear Parameter Varying (LPV)

Shamma and Athans (1991) developed a new modelling framework for gain scheduling control, called Linear Parameter Varying (LPV). The LPV strategy reduces the nonlinear system to a linear time invariant (LTI) system for frozen values of the scheduling parameter variable. The LPV model can be obtained using linearization or parameterization of the state-space model as a nonlinear function of the scheduling parameter. A discrete LPV system is represented in state-space as:

$$x(k+1) = A(\rho(k))x(k) + B(\rho(k))u(k) \tag{4.12}$$

$$y(k) = C(\rho(k))x(k) + D(\rho(k))u(k) \tag{4.13}$$

The parameter variable $\rho(k)$ is assumed a priori unknown. However, it can be measured or estimated upon operation of the system using LPV identification techniques.

The advantage of this approach is that it offers a theoretical framework to ensure performance and robustness of the controlled system via convex optimisation over LMI (Mohammadpour Velni and Grigoriadis, 2008). However, for use in the control of large scale industrial processes, the modelling of an LPV is very complex and conservative, which might be difficult to construct. In addition, the identification problem is very complex and requires high costs for plant testing (Xu et al., 2009).

4.4.3 NMPC with State-Dependent State-Space Models

The linear time-varying models can also be obtained using a state dependent description of the nonlinear system. The state dependent representation of a system model avoids model linearization. The non-linearity is handled by the algebraic re-arranging of the original state-space model into a linear time-varying representation. Based on the state dependent state-space model, Cloutier (1997) developed a nonlinear regulator technique called the State Dependent Riccati Equation (SDRE). This technique has been described as a nonlinear extension of the well-known LQR formulation for linear systems.

The use of state-space models for linear Generalised Predictive Control (GPC) was first proposed by Ordys and Clarke (1993). Inspired by SDRE, NMPC based on the state-dependent model was developed by Ordys and Grimble (2001). This approach has been used in many papers, such as by Dutka et al. (2003), Youssef et al.(2003), and by Shakouri and Ordys (2011). In this section, a supervisory constrained NMPC control strategy based on the LTV state-dependent model is formulated and applied for the regulation of a nonlinear power plant model. The technique presented here is adapted from (Ordys and Grimble, 2001) and (Orlowski, 2011).

4.4.3.1 Model Representation

The proposed control strategy consists of two levels, a conventional PI regulatory level and a supervisory NMPC optimization level, as shown in Figure 4.1. In this two layers architecture, the regulator level is assumed to be the existing plant PI controllers. The supervisory NMPC algorithm is used as a second level controller, to

generate optimal set points to the lower level regulating PI loops. The advantage of this structure is that the NMPC algorithm is sitting on top of the existing PI control structure and does not interfere with the closed loop control system. In addition, the model used in NMPC design is, therefore, open-loop stable.

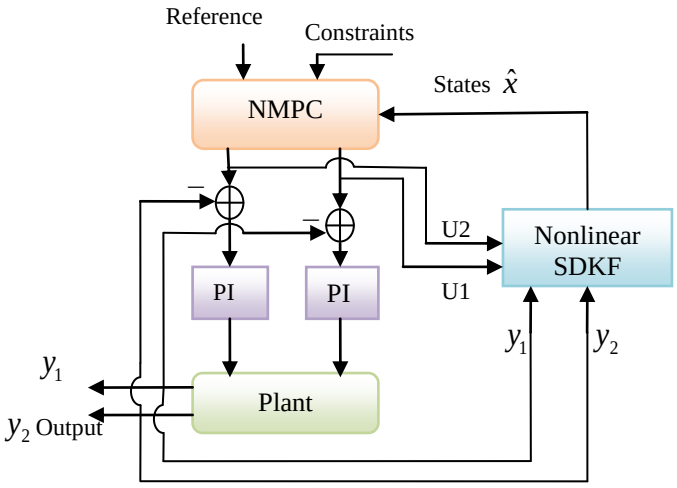


Figure 4.1: Supervisory NMPC control strategy

Let the system under control described by the state-dependent nonlinear model be:

$$\begin{aligned} \dot{x}_p &= A_p(x_p)x_p + B_p(x_p)u_p \\ y_p &= C_p(x_p)x_p \end{aligned} \tag{4.14}$$

where: $A_p(x)$, $B_p(x)$ and $C_p(x)$ are the SDC matrices which are chosen, such that the system is observable and controllable for all *x*.

The state space equation describing the PI controllers can be represented as follows:

$$\begin{aligned} \dot{x}_c &= A_c x_c + B_c e \\ u_p &= C_c x_c + D_c e \end{aligned} \tag{4.15}$$

where: x_c is the state of PI controllers, and e is the error signal and defined as $e = r - y_p$.

After appropriate substitutions, the state space equations for both the boiler model and the PI controller can be written as follows:

$$\begin{aligned} \dot{x} &= A_1(x)x + B_1(x)u \\ y &= C_1(x)x \end{aligned} \tag{4.16}$$

where:

$$x = \begin{bmatrix} x_c \\ x_p \end{bmatrix}, \qquad u = r,$$

$$A_1(x) = \begin{bmatrix} A_c & -B_c C_p(x) \\ B_p(x)C_c & A_p(x) \end{bmatrix}, \quad B_1(x) = \begin{bmatrix} B_c \\ 0 \end{bmatrix}, \quad C_1(x) = \begin{bmatrix} 0 & C_p(x) \end{bmatrix}$$

The state dependent model (4.16) is discretized using the Euler integration method and converted to a state-dependent nonlinear discrete-time model:

$$\begin{aligned} \tilde{x}_{k+1} &= \tilde{A}_k(\tilde{x}_k)\tilde{x}_k + \tilde{B}_k(\tilde{x}_k)u_k \\ \tilde{y}_k &= \tilde{C}_k(\tilde{x}_k)\tilde{x}_k \end{aligned} \tag{4.17}$$

where $\tilde{A}_k(\tilde{x}_k)$, $\tilde{B}_k(\tilde{x}_k)$ and $\tilde{C}_k(\tilde{x}_k)$ are the SDC discrete matrices.

In order to achieve offset-free performance, the process model (4.17) was augmented to include constant step output disturbances. The augmented state-space system is given by:

$$\begin{aligned} x_{k+1} &= A_k(x_k)x_k + B_k(x_k)u_k \\ y_k &= C_k(x_k)x_k \end{aligned} \tag{4.18}$$

where:

$$A(x_k) = \begin{bmatrix} \tilde{A}(\tilde{x}_k) & 0 \\ 0 & I \end{bmatrix}, \; B(x_k) = \begin{bmatrix} \tilde{B}(\tilde{x}_k) \\ 0 \end{bmatrix}, \; C(x_k) = \begin{bmatrix} \tilde{C}(\tilde{x}_k) & I \end{bmatrix}, \; x_k = \begin{bmatrix} \tilde{x}_k \\ d_k \end{bmatrix}$$

For simplicity, the process model in the state-dependent coefficient form (4.18) can be written as follows:

$$x_{k+1} = A_k x_k + B_k u_k + w_k$$
$$y_k = C_k x_k + v_k \tag{4.19}$$

where: $A_k = A_k(x_k)$, $B_k = B_k(x_k)$ and $C_k = C_k(x_k)$. $k = k_0, ...k_0 + N_p - 1$ and N_p form the prediction horizon. w_k and v_k are the process and measurement white noises, respectively.

4.4.3.2 Prediction Calculation

At the time instant k, the current estimated state $\hat{x}_k$ and the past input vector $U_{k-1} = \begin{bmatrix} u_{k-1}^T & u_{k-1+1}^T & \cdots & u_{k-1+Np-1}^T \end{bmatrix}^T$, calculated at previous iteration of the control algorithm are used to predict the future trajectory. The first element of U_{k-1} has already been used, but the remaining part can be used by repeating the last element of the vector once again, i.e. $u_{k-1+N_p-1} = u_{k-1+N_p-2}$.

Therefore, the state prediction vector $X(k+1, N_p \mid k) = \begin{bmatrix} x_{k+1}^T & x_{k+2}^T & \cdots & x_{k+N}^T \end{bmatrix}^T$ and its associated matrices $A(k+j)$, $B(k+j)$ and $C(k+j)$ (where: $j=1,...,N_p$) are calculated iteratively, using the following equation based on the time-varying approximation:

$$\begin{aligned} X_{k+j} &= [A_{k+j-1}A_{k+j-2}...A_k]x_k + [A_{k+j-1}A_{k+j-2}...A_{k+1}]B_k u_k .... \\ &+ [A_{k+j-1}A_{k+j-2}...A_{k+2}]B_k u_{k+1} + [A_{k+j-1}A_{k+j-2}....A_{k+Np}]B_{k-1+j}u_{k-1+j} \end{aligned} \tag{4.20}$$

The state prediction equation can be represented in vector form, as follows:

$$\hat{X} = \hat{\Lambda}\ \hat{x}_0 + \hat{\Phi}\hat{U} \tag{4.21}$$

where:

$$\hat{\Phi} = \hat{L}\hat{B}$$

$$\hat{\Lambda} = \begin{bmatrix} \phi_k^k \\ \vdots \\ \phi_k^{k+Np-1} \end{bmatrix}, \qquad \hat{L} = \begin{bmatrix} I & 0 & \cdots & 0 \\ \phi_{k+1}^{k+1} & I & 0 & \vdots \\ \vdots & \ddots & I & 0 \\ \phi_{k+1}^{k+Np-1} & \cdots & \phi_{k+Np-1}^{k+Np-1} & I \end{bmatrix}$$

$$\hat{B} = \begin{bmatrix} B_k & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & B_{k+Np-1} \end{bmatrix}$$

$$\phi_i^k = A_k\, A_{k-1} \ldots\ldots A_i$$

The output prediction vector $\hat{Y} = \begin{bmatrix} y_{k+1}^T & \cdots & y_{k+Np}^T \end{bmatrix}^T$ can be calculated as follows:

$$\hat{Y} = \begin{bmatrix} C_{k_0} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & C_{k_0+Np-1} \end{bmatrix} \hat{X}$$

4.4.3.3 Cost Function Minimization

To obtain an optimal control sequence, the following performance index must be minimized within a prediction horizon.

$$J(\hat{x}_{k+1|k}, U_k) = \frac{1}{2} \sum_{j=1}^{Np} \| \hat{y}_{k+j|k} - r_{k+j} \|_Q^2 + \| u_{k+j|k} - u_{k+j-1|k} \|_S^2 \tag{4.22}$$

Subject to constraints specified on outputs:

$$y_{\min} \le y_k \le y_{\max} \tag{4.23}$$

where: N_p is the maximum output horizon, and $Q \ge 0$ and $S > 0$ are the weightings on the tracking error and the control increments, respectively. As discussed in Chapter 3, the objective function can be expressed as follows:

$$J(\hat{x}_{k+1|k}, U_k) = \frac{1}{2} U_k^T H U_k + U_k^T f \tag{4.24}$$

where: $H = \Phi^T \bar{Q} \Phi + H_s, \qquad f = \Gamma \begin{bmatrix} \hat{x}_{k+1|k} \\ R_k \end{bmatrix} - \begin{bmatrix} Su_k \\ 0 \\ . \\ 0 \end{bmatrix}$

$$f = \Gamma \begin{bmatrix} \hat{x}_{k+1|k} \\ R_k \end{bmatrix} - \begin{bmatrix} Su_k \\ 0 \\ . \\ 0 \end{bmatrix}, \qquad \Gamma = \begin{bmatrix} \Phi^T \bar{Q} \Lambda & -\Phi^T \bar{Q} \end{bmatrix}$$

$$\bar{Q} = \begin{bmatrix} Q & & & \\ & Q & & \\ & & \ddots & \\ & & & Q \end{bmatrix}, \qquad H_s = \begin{bmatrix} 2S & -S & & & \\ -S & 2S & -S & & \\ & . & . & . & \\ & & -S & 2S & -S \\ & & & -S & S \end{bmatrix}$$

The optimal U_k can be found by solving (4.24) using the MATLAB quadratic programming function "Quadprog". The output of the process can be limited by introducing upper and lower output constraints, $Y_{\max}$ and $Y_{\min}$, as discussed in Chapter 3. The output constraints are specified as $Y_{\min} \le Y_k \le Y_{\max}$. The output constraints are expressed in terms of U_k. Then, the cost function is solved subject to the inequality constraints:

$$A_c U_k \le B_c \tag{4.25}$$

where:

$$A_c = \begin{bmatrix} -\tilde{C}\Phi \\ \tilde{C}\Phi \end{bmatrix}, \qquad B_c = \begin{bmatrix} -Y_{\min} + \tilde{C}\Lambda \hat{x}(k) \\ Y_{\max} - \tilde{C}\Lambda \hat{x}(k) \end{bmatrix}$$

$$\tilde{C} = \begin{bmatrix} C_{k_0} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & C_{k_0+Np-1} \end{bmatrix}$$

4.4.3.4 Stability of State-Dependent NMPC

The stability of NMPC system can either be obtained by tuning the parameters of the controller, or it can be achieved independently of the choice of the performance parameters, which is usually referred to as an NMPC with guaranteed stability. Different possibilities for guaranteeing the closed-loop stability for NMPC have been proposed (Keerthi and Gilbert, 1988; Michalska and Mayne, 1993; Chen and Allgöwer, 1998). Different formulations with guaranteed stability can be found in the survey paper by (Mayne et al., 2000). The stability is usually ensured by adding suitable equality or inequality constraints, and suitable additional penalty terms to the cost functional, as described in Chapter 3. Stability can also be ensured without these constraints, by selecting a sufficiently large optimization horizon, N_p. However, NMPC with stability guarantee issues is still open, especially when integrated with a state observer (Camacho and Bordons, 2007; Huang et al., 2012).

Since the stabilizing properties of MPC with linear models are well known, LTV techniques can be employed, in some cases, to establish stability. These techniques transform the original nonlinear model into a group of linear models. In this thesis, the proposed constrained NMPC based on the state-dependent model can be locally stabilized using the stability formulation for linear MPC. However, guaranteeing global stability without sacrificing performance or imposing conservative rules is still an open issue, as in the case of the SDRE controller which needs to be tackled (Çimen, 2012).

In this study, the stability of the proposed NMPC based on the state-dependent model is ensured by selecting the SDC matrices, in order to fulfil theorem 4.3.1. Then, a simulation analysis is used to assess stability.

4.4.3.5 The NMPC Algorithm

The following steps summarize the NMPC control technique presented in this study:

1- Estimate the current state vector $\hat{x}_k$.

2- At k=0, the initial control trajectory $U_{k,Np}$ can be assumed to be a step control signal with amplitude from the normal operating range for the control.

3- For $k > 0$, given the vector $U_{k,Np}$, which was calculated in the previous iteration after removing the first element, which has already been used in the previous iteration for control and repeating the last element of the vector once again, i.e., $u_{k+Np-1} = u_{k+Np-2}$.

4- Substitute the calculated $U_{k,Np}$ into the state equation and calculating iteratively the state prediction and its associated matrices; $\hat{x}_{k+j}$, A_{k+j} , B_{k+j} and C_{k+j} for n= 1,….., N_p.

5- Calculate the output predictions and the control vector $U_{k,Np}$.

6- Check the differences between the control vectors:

$\left|U_{k,Np}(new) - U_{k,Np}(old)\right| < \sigma$. If this condition is not satisfied, put $U_{k,Np}(old) = U_{k,Np}(new)$ then go to step (3). If the condition is satisfied, then apply the first element of $U_{k,Np}$ to the plant and put k=k+1, then go to step (1).

4.5 Non-linear Estimation

Most NMPC schemes in literature assume the availability of the measurement of complete states, which is not always possible. Therefore, it is important to incorporate a nonlinear state observer into the NMPC scheme to estimate the non-measurable states.

The most commonly used state estimation method is the extended Kalman filter (EKF), due to its simplicity, robustness and suitability for real-time implementations (Lee and Ricker, 1994). The basic idea of EKF is to perform linearization at each time step, to approximate the nonlinear system as a time-varying system to extend the scope of the Kalman filter to systems described by nonlinear functions. However,

for non-linear systems due to approximations made during linearization the state estimates convergence is not guaranteed. In addition, if the initial estimate is far from the actual, or if the process is modelled incorrectly, the filter may quickly diverge (Lo and Rathamarit, 2008).

The state-dependent Riccati equation (SDRE) filtering techniques (Mracek et al., 1996) are rapidly emerging as nonlinear estimators for a variety of nonlinear applications (Çimen, 2012). In contrast to EKF, which is based on Taylor series type linearization, the SDRE technique is based on state-dependent parameterization, which fully captures the nonlinearity by reducing the nonlinear system to a linear structure with state dependent coefficient (SDC) matrices. In the multivariable case, the SDC parameterization is not unique (Cloutier et al., 1996) and can be used to create additional degrees of freedom. This can be used to overcome the limitations in traditional filtering methods, such as singularities and the loss of observability.

There are two commonly used approaches for the SDRE filtering technique. The first approach has the same structure as the steady-state linear Kalman filter (Mracek et al., 1996). In this approach, the Kalman gain is obtained by solving a state dependent algebraic Riccati equation (SDARE). This approach has disadvantages such as high computation costs, and the solution depends significantly on the observability property of the system (Jaganath et al., 2005). The second approach suggested by Haessig and Friedland (2002) uses a State Dependent Differential Riccati Equation (SDDRE) for the generation of filter gains, to avoid the observability shortcomings of the SDARE method.

4.5.1 Extended Kalman Filter

The most commonly used state estimation method in NMPC is the extended Kalman filter (EKF). Let the nonlinear model be expressed as follows:

$$\begin{aligned} x_{k+1} &= f_k\left(x_k, u_k\right) + G_k\left(x_k\right)\xi_k \\ y_k &= h_k\left(x_k\right) + \eta_k \end{aligned} \tag{4.26}$$

where: $f_k(x_k, u_k)$ and $h_k(x_k)$ are assumed to be continuously differentiable, and ξ_k and η_k are the process and measurement noise which are assumed to be zero mean Gaussian process with covariance matrices Q_w and R_w, respectively.

Using the linear Taylor series approximation of the system (4.26), the linear approximation at the current estimated state is derived. A detailed analysis of the method can be found in Chui and Chen (2009). The extended Kalman filter algorithm proceeds as follows:

Initialization:

$$\hat{x}_{k-1/k-1} = \hat{x}_0, P_{k-1/k-1} = P_0 \tag{4.27}$$

Model prediction:

$$x_{k/k-1} = f(x_{k-1/k-1}, u_k) \tag{4.28}$$

$$P_{k/k-1} = F_k P_{k-1/k-1} F_K^T + G_{k-1}(\hat{x}_{k-1/k-1})\, Q_{wk-1}\, G_{k-1}(\hat{x}_{k-1/k-1}) \tag{4.29}$$

Measurement update:

$$K_k = P_{k/k-1} H_k^T (H_k P_{k/k-1} H_k^T + R_w)^{-1} \tag{4.30}$$

$$\hat{x}_{k/k} = \hat{x}_{k/k-1} + K_k (y_k - h(\hat{x}_{k/k-1})) \tag{4.31}$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} (I - K_k H_k)^T + K_k R_w K_k^T \tag{4.32}$$

where:

$$F_k = \frac{\partial f_{k-1}}{\partial x_{k-1}}(\hat{x}_{k-1/k-1}, u_{k-1}), \qquad H_k = \frac{\partial h_k}{\partial x_k}(\hat{x}_{k/k-1})$$

In EKF, the filter gain K_k , $P_{k/k-1}$ and $P_{k/k}$ depend on previous state estimates and on the measurements, thus they cannot be computed off-line as in the linear Kalman filter.

For non-linear systems, due to approximations made on Taylor series expansion during linearization, state estimates' convergence is not guaranteed. The linearized model matrices are functions of the state about which the linearization was carried out. Therefore, model mismatch could happen due to state estimation error. This may lead to improper state updates, and consequently to divergence. Divergence may also occur if the initial state value is far from the actual.

4.5.2 State-Dependent Kalman Filter

An alternative for the extended Kalman filter is presented in this section. In this thesis, the emphasis is placed on state-dependent Kalman filters. The state-dependent Kalman filter (SDKF) was developed by Mracek et al.(1996). It combines the concept of linear time-varying models with the well known Kalman theory for linear systems. It has been used in nonlinear filter development and control designs for some nonlinear benchmark problems for state estimation (Çimen, 2012). The nonlinearities of the system are fully captured by the state-dependent representation, which reduces the nonlinear system to a linear structure with state dependent coefficients.

4.5.2.1 SDARE Kalman Filter

The estimation problem using the state-dependent algebraic Riccati equation is now analyzed. Assume that the nonlinear dynamics model (4.26) can be formulated into SDC form, as described in section 4.3:

$$\begin{aligned} x_{k+1} &= A_k x_k + B_k u_k + \xi_k \\ y_k &= C_k x_k + \eta_k \end{aligned} \tag{4.33}$$

where $A_k = A(x_k),\ B_k = B(x_k),\ C_k = C(x_k)$ are state-dependent matrices. Additionally, it is assumed that $\{C_k, A_k\}$ is point-wise observable for all x.

The one-step recursive update formulation of the SDARE-based observer is given by:

$$\hat{x}_{k+1} = \hat{A}_k \hat{x}_k + \hat{B}_k u_k + K_k (y_k - \hat{C}_k \hat{x}_k) \tag{4.34}$$

$$y_k = C_k x_k \tag{4.35}$$

The observer gain for the one-step formulation is given by:

$$K_k = P_k \hat{C}_k^T (\hat{C}_k P_k \hat{C}_k^T + R_w)^{-1} \tag{4.36}$$

where P_k is updated using the state-dependent algebraic Riccati equation (SDARE):

$$P_k = \hat{A}_k \left[P_k - P_k \hat{C}_k^T (R_w + \hat{C}_k P_k \hat{C}_k^T)^{-1} \hat{C}_k P_k \right] \hat{A}_k^T + Q_w \tag{4.37}$$

4.5.2.2 SDDRE Kalman Filter

In this thesis, the emphasis is placed on the state-dependent differential Riccati equation Kalman (SDDRE) filter proposed by Haessig and Friedland (2002), which avoids the observability shortcomings and computation costs of the SDARE method. Consider the same system dynamics given by (4.26), and also assume that the nonlinear dynamics can be formulated into the state-dependent coefficient form as follows:

$$\begin{aligned} \dot{x} &= A(x)x + B(x)u \\ y &= C(x)x \end{aligned} \tag{4.38}$$

The estimated state $\hat{x}$ is given by:

$$\dot{\hat{x}}_{k+1} = A(\hat{x})\hat{x} + B(\hat{x})u + K(x)(y - C(\hat{x})\hat{x}) \tag{4.39}$$

The observer gain is given by:

$$K(\hat{x}) = P(\hat{x})C^T(\hat{x})R_w^{-1} \tag{4.40}$$

where $P(\hat{x})$ is updated using the state-dependent differential Riccati equation (SDDRE):

$$\dot{P}(\hat{x}) = P(\hat{x})A^T(\hat{x}) + A(\hat{x})P(\hat{x}) - P(\hat{x})C^T(\hat{x})R_w^{-1}C(\hat{x})P(\hat{x}) + Q_w \tag{4.41}$$

The above equations show that the structure of the SDRE filter is similar to EKF. SDRE differs from EKF in that it uses the exact linear model, assuming the state estimates are accurate. In contrast, the linear model of EKF suffers from linearization error.

4.5.2.3 Stability of SDRE Filter

The minimum requirement for the local convergence of the SDRE nonlinear state estimator is that the system model (4.38) ($f(x) = \dot{x}$) has detectable parameterization, and is locally Lipschitz continuous for all *x* in the interested domain (Banks et al., 2007). This can be satisfied if there is a constant $L > 0$, such that:

$$\|f(y) - f(z)\| \le L\|y - z\|, \qquad \forall y, z \in B \tag{4.42}$$

where: the open set $B \subseteq \mathbb{R}^m$.

In this thesis, simulation analysis is used to verify this condition instead of analytical analysis due to the complexity of the system model. For example, Figure 4.2 shows the plot of Lipschitz constant *L* versus the pressure state using boiler model for any $y, z \in (x,\ x + 0.02)$, where the state vector *x* is chosen in the domain of the neighbourhood of the operating point x_0. From this figure it is clearly shown that *L* is bounded which demonstrates that the local Lipschitz condition is satisfied in this system. However, global stability of the SDRE filter is more difficult to demonstrate. This is because getting stable eigenvalues of the SDRE system at each sampling time

does not guarantee asymptotic stability. Although the SDRE filter has been successively implemented in different applications (Çimen, 2012), there are only a few papers in literature which provide theoretical analysis for global stability, such as (Jaganath et al., 2005; Banks et al., 2007; Nemra and Aouf, 2010; Beikzadeh and Taghirad, 2012). These methods either impose conservative conditions which can be violated for many applications, or require simulation to verify certain conditions or to find unknown parameters, especially for large scale or complex systems. So far, in practice, the global stability analysis has been performed using simulation tests (Çimen, 2012).

New robust h-infinity SDRE filters are addressed in (Beikzadeh and Taghirad, 2012; Iratni et al., 2012), in order to overcome the effects of modelling uncertainty, measurement noise and input disturbance in the performance of the standard SDRE filter.

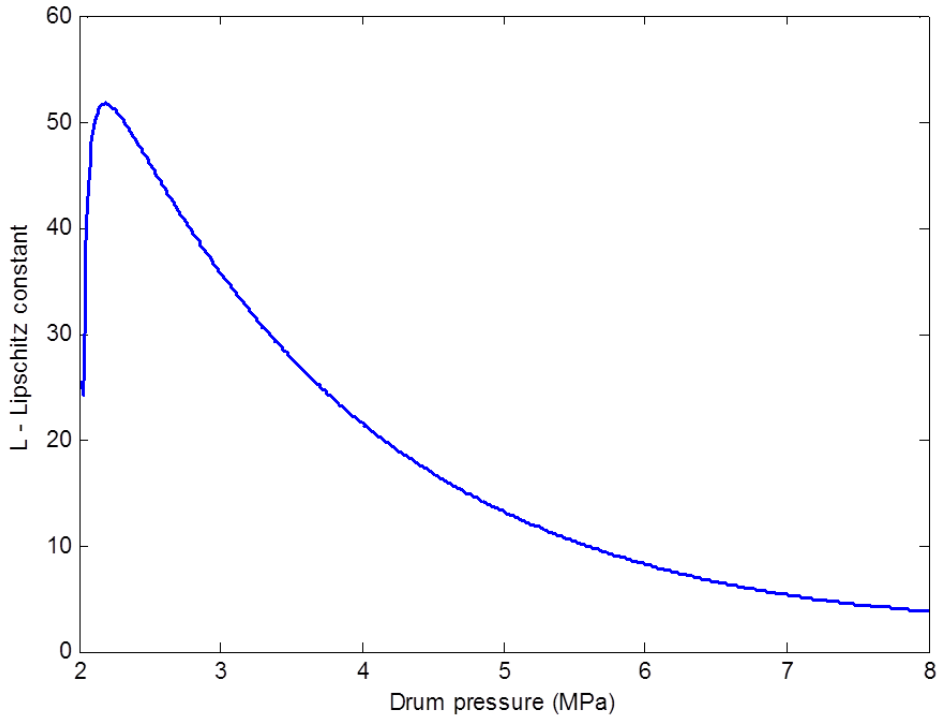


Figure 4.2: Lipschitz constant

4.6 Applications of NMPC to a Nonlinear Industrial Power Plant

4.6.1 NMPC for a Nonlinear Boiler Model

In this section, the NMPC algorithm described in section 4.4.3.5 is applied to control a nonlinear boiler model. This boiler model includes both the fourth order Astrom and Bell (2000) drum-boiler model, and the furnace dynamic model described in Chapter 2. The optimum set-point of the boiler drum pressure P_d^r is calculated using the NMPC to improve tracking and disturbance rejection, as well as to minimise an economic performance index. Then, using this optimum set-point, the PI controller regulates the steam pressure by adjusting the fuel flow to the boiler supplementary furnace as shown in Figure 4.3.

This model will be integrated with other power plant subsystems, to form a complex centralized power plant model. Hence, in order to simplify the control problem, the water level in the drum-boiler is controlled using a PID controller, by adjusting the input water flow to the boiler. A nonlinear state-dependent Kalman filter (SDKF) will be used to estimate the system states.

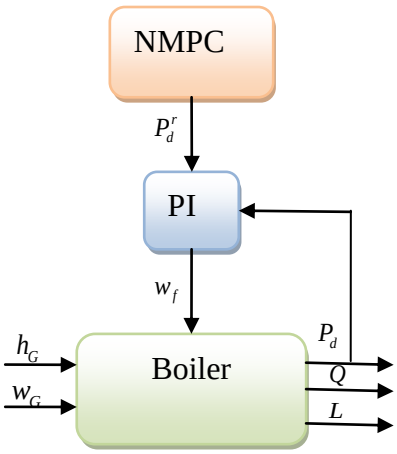


Figure 4.3: Boiler control strategy

4.6.1.1 State-Dependent Coefficients (SDC) Representation of Boiler Model

This section will discuss the SDC representation of the nonlinear boiler model, which is used to predict the boiler outputs in the NMPC calculations. Since the water level is controlled by using a PI controller, the fourth-order boiler-drum model can be reduced to a second-order model as follows:

$$\begin{aligned} e_{11}\frac{dV_{wt}}{dt} + e_{12}\frac{dp}{dt} &= q_f - q_s \\ e_{21}\frac{dV_{wt}}{dt} + e_{22}\frac{dp}{dt} &= Q_{ir} + q_f h_f - q_s h_s \end{aligned} \tag{4.43}$$

This model can be rewritten in state-space form, as follows:

$$\begin{aligned} \frac{dV_{wt}}{dt} &= \frac{e_{22}(q_f - q_s)}{e_{11}e_{22} - e_{12}e_{21}} - \frac{e_{12}(Q_{ir} + q_f h_f - q_s h_s)}{e_{11}e_{22} - e_{12}e_{21}} \\ \frac{dp}{dt} &= \frac{-e_{21}(q_f - q_s)}{e_{11}e_{22} - e_{12}e_{21}} + \frac{e_{11}(Q_{ir} + q_f h_f - q_s h_s)}{e_{11}e_{22} - e_{12}e_{21}} \end{aligned} \tag{4.44}$$

This model is integrated with the furnace model described in Appendix A. The complete model equation is expressed as follows:

$$\begin{aligned} \dot{x}_1(1) &= \frac{1}{V_f}\left[h_A w_A + h_G w_G - Q_{ir} - Q_{is} - k_f R_1 T_g R_s (1 + \frac{y}{100}) x_1(1) \right] \\ \dot{x}_1(2) &= \frac{1}{V_f}\left[w_f + w_A + w_G - k_f R_1 x_1(2) T_g \right] \\ \dot{x}_1(3) &= \frac{e_{22}(q_f - q_s)}{e_{11}e_{22} - e_{12}e_{21}} - \frac{e_{12}(Q_{ir} + q_f h_f - q_s h_s)}{e_{11}e_{22} - e_{12}e_{21}} \\ \dot{x}_1(4) &= \frac{-e_{21}(q_f - q_s)}{e_{11}e_{22} - e_{12}e_{21}} + \frac{e_{11}(Q_{ir} + q_f h_f - q_s h_s)}{e_{11}e_{22} - e_{12}e_{21}} \end{aligned} \tag{4.45}$$

The heat transferred to the boiler drum Q_{ir} can be expressed as a function of $x_1(1)$ and $x_1(2)$, as follows:

$$Q_{ir} = Q_{m1} x_1(1) = Q_{m2} x_1(2) \tag{4.46}$$

where:

$$Q_{m1} = 4\theta k V_f \sigma T_g^3 \left(\frac{1}{c_{pg} x_1(2)} - \frac{h_{ref}}{c_{pg} x_1(1)} + \frac{T_{ref}}{x_1(1)} \right)$$

$$Q_{m2} = 4\theta k V_f \sigma T_g^3 \left(\frac{x_1}{c_{pg} x_1^2(2)} - \frac{h_{ref}}{c_{pg} x_1(2)} + \frac{T_{ref}}{x_1(2)} \right)$$

The state-space, input and output are defined as $x_1 = [x_{f1}, \rho_{EG}, V_{wt}, p]$, $u_1 = w_f$ and $y_1 = p$, respectively. After appropriate substitutions, the SDC model can be represented as follows:

$$\begin{aligned} \dot{x}_1 &= \tilde{A}_{11}(\hat{x}_1) x_1 + \tilde{B}_{11}(\hat{x}_1) u_1 \\ y_1 &= \tilde{C}_{11}(\hat{x}_1) x_1 \end{aligned} \tag{4.47}$$

where:

$$\tilde{A}_{11}(\hat{x}_1, u_1) = \begin{bmatrix} s_{11} & s_{12} & s_{13} & 0 \\ s_{21} & s_{22} & 0 & 0 \\ 0 & s_{32} & s_{33} & 0 \\ s_{41} & 0 & 0 & s_{44} \end{bmatrix}, \quad \tilde{B}_{11}(\hat{x}_1) = \begin{bmatrix} l_1 \\ l_2 \\ 0 \\ 0 \end{bmatrix}, \quad \tilde{C}_{11}(\hat{x}_1) = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}$$

$$s_{11} = -\frac{1}{\nu_f}\left[k_f R_1 T_g R_s \left(1 + \frac{y}{100}\right) \right], \qquad s_{12} = \frac{1}{V_f x_1(2)} \left[h_G w_G - Q_{ir} - Q_{is} \right],$$

$$s_{13} = \frac{h_A w_A}{V_f x_1(2)}, \qquad s_{21} = \frac{w_A + w_G}{V_f x_1(1)},$$

$$s_{22} = \frac{-k_f R_1 T}{V_f}, \qquad s_{32} = -\frac{e_{12} Q_{m1}}{e_{11} e_{22} - e_{12} e_{21}},$$

$$s_{33} = \frac{e_{22}(q_f - q_s) - e_{12}(q_f h_f - q_s h_s)}{(e_{11} e_{22} - e_{12} e_{21}) x_1(3)}, \quad s_{41} = \frac{e_{11} Q_{m1}}{e_{11} e_{22} - e_{12} e_{21}} - \frac{e_{21}(q_f - q_s)}{(e_{11} e_{22} - e_{12} e_{21}) x_1(1)},$$

$$s_{44} = \frac{e_{11}(q_f h_f - q_s h_s)}{(e_{11} e_{22} - e_{12} e_{21}) x_1(4)}, \qquad l_1 = \frac{C_f}{V_f}, \qquad l_2 = \frac{1}{V_f}$$

The SDC model (4.47) is discretized using the Euler method and augmented with the PI controller, as described in section 4.4.3.1, to get the following model:

$$\begin{aligned} x_1(k+1) &= A_{11}(\hat{x}_1) x_1(k) + B_{11}(\hat{x}_1) u_1(k) \\ y_1(k) &= C_{11}(\hat{x}_1) x_1(k) \end{aligned} \tag{4.48}$$

where:

$$A_{11}(\hat{x}_1, u_1) = \begin{bmatrix} 1 & 0 & 0 & 0 & -T \\ Tl_1 k_{i1} & Ts_{11}+1 & Ts_{12} & Ts_{13} & -Tl_1 k_{p1} \\ Tl_2 k_{i1} & Ts_{21} & Ts_{22}+1 & 0 & -Tl_2 k_{p1} \\ 0 & 0 & Ts_{32} & Ts_{33}+1 & 0 \\ 0 & Ts_{41} & 0 & 0 & Ts_{44}+1 \end{bmatrix},$$

$$B_{11}(\hat{x}_1) = \begin{bmatrix} T \\ Tl_1 k_{p1} \\ Tl_2 k_{p1} \\ 0 \\ 0 \end{bmatrix}, \qquad C_{11}(\hat{x}_1) = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

k_{p1} and k_{i1} are the PI controller gains, and T is the sampling time. In this model, the state space vector is defined as $x_1 = [x_{c1}, x_1(1), x_1(2), x_1(3), x_1(4)]$, where x_{c1} is the PI controller state.

4.6.1.2 **Simulation Results**

In this simulation, the best values for the PID gains are found using the Multivariable PID tuning methods described in Chapter 3. The prediction horizon is selected as N_p=10 and the sampling time T=0.02. The controller has been configured by choosing the weight on tracking error Q=450 and weight on control increments S=367. The optimization problem of MPC scheme is solved using the MATLAB Quadprog function provided by the optimization toolbox. Regarding the termination criteria for solving the NMPC problem and the maximum number of iterations, the values used by the simulations are 0.001 and 45, respectively.

The performance of SDKF is demonstrated in Figure 4.4, which shows a comparison between the boiler true states and their estimates of step changes in drum pressure set points when the process and output noises are added to the system. The process noise and sensor noise are $\zeta \sim N(0, 0.001)$ and $\eta \sim N(0, 0.01)$, respectively. The simulation results show a good estimate performance.

Figure 4.5 shows the boiler pressure and water level responses to set-point changes in the boiler pressure at 150 s. It is clearly shown that the controller has a good tracking performance. The controlled variable and manipulated variables are shown in Figure 4.6.

Figure 4.7 and Figure 4.8 show the boiler output, control variable and manipulated variables responses to both set-point changes in boiler pressure at 150 s, and output step disturbance of 0.1 MPa on boiler pressure at 300 s. As shown in these figures, the disturbance can be eliminated efficiently through the proposed NMPC algorithm.

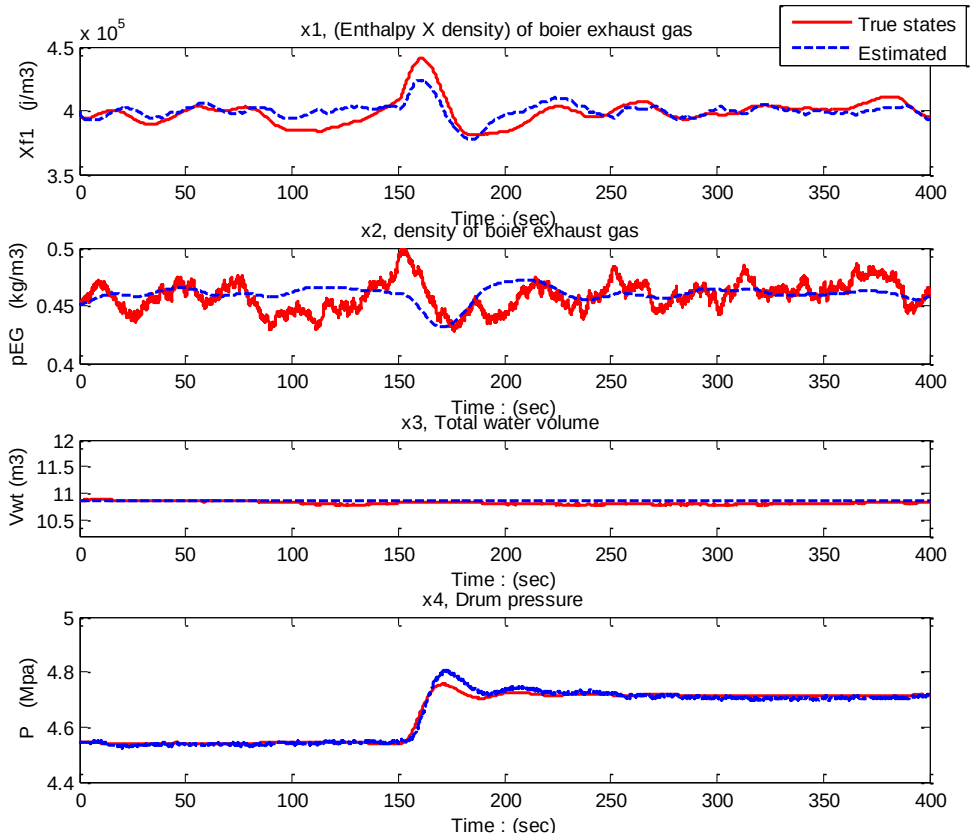


Figure 4.4: Boiler true states and their estimates

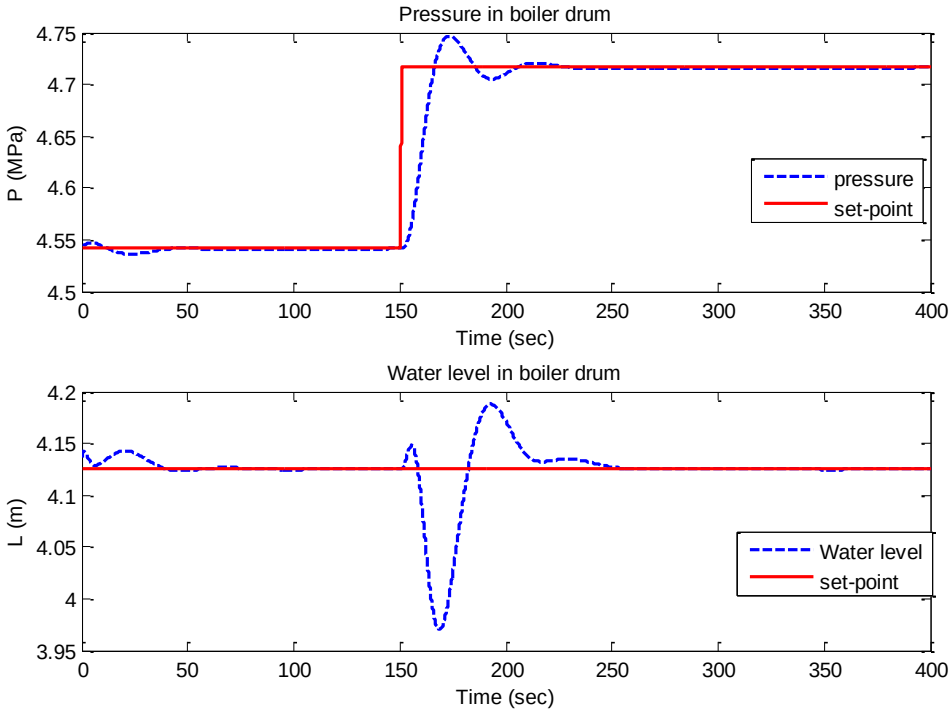


Figure 4.5: Output response to set-point changes at 150 sec

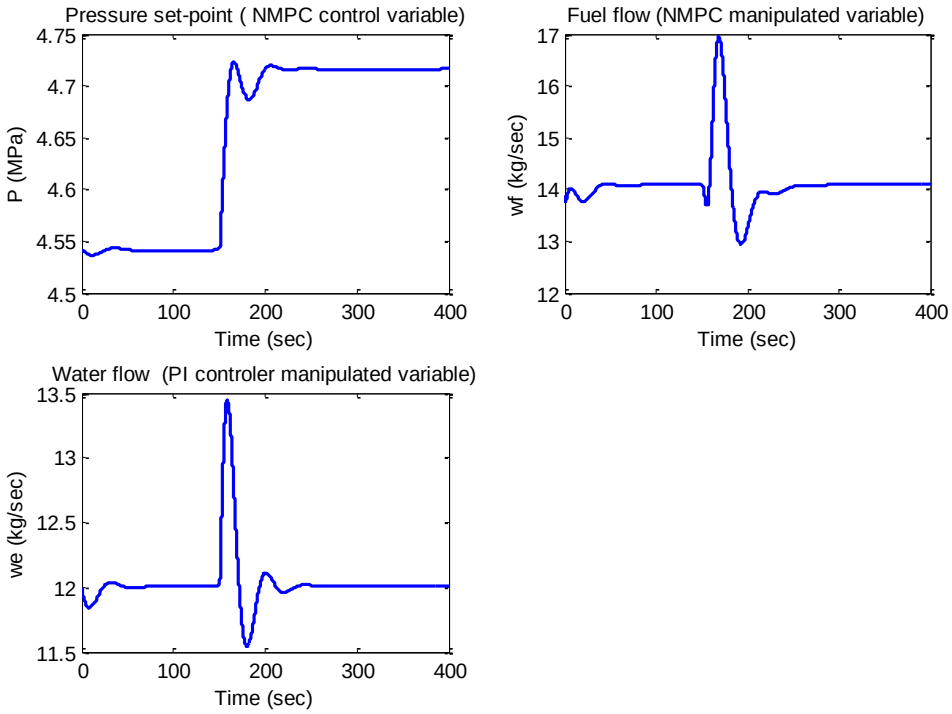


Figure 4.6: Control and manipulated variable response to set-point changes

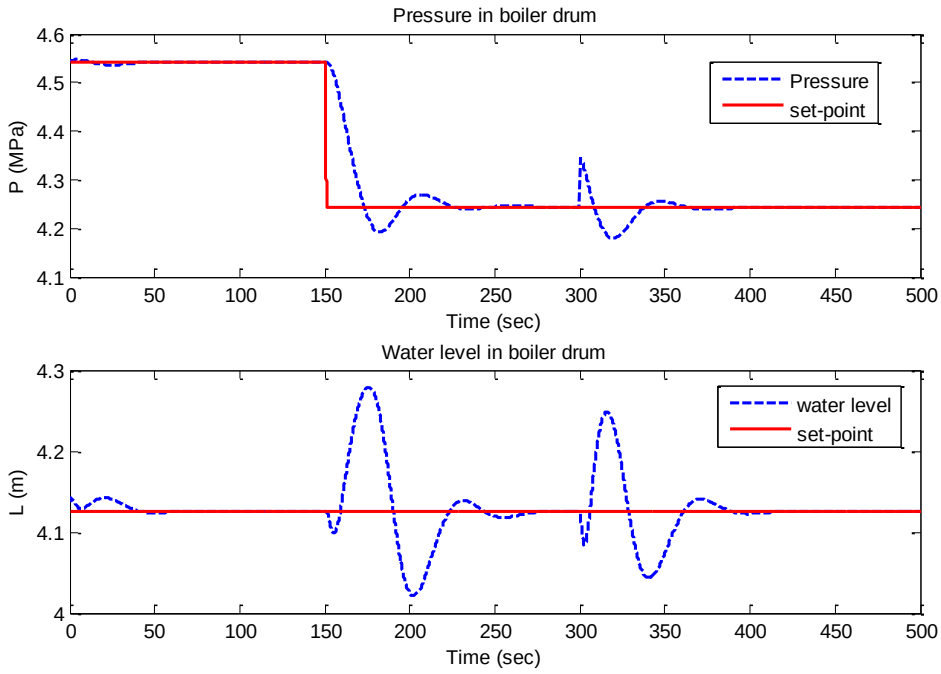


Figure 4.7: Output response to output disturbance on boiler pressure at 300 sec

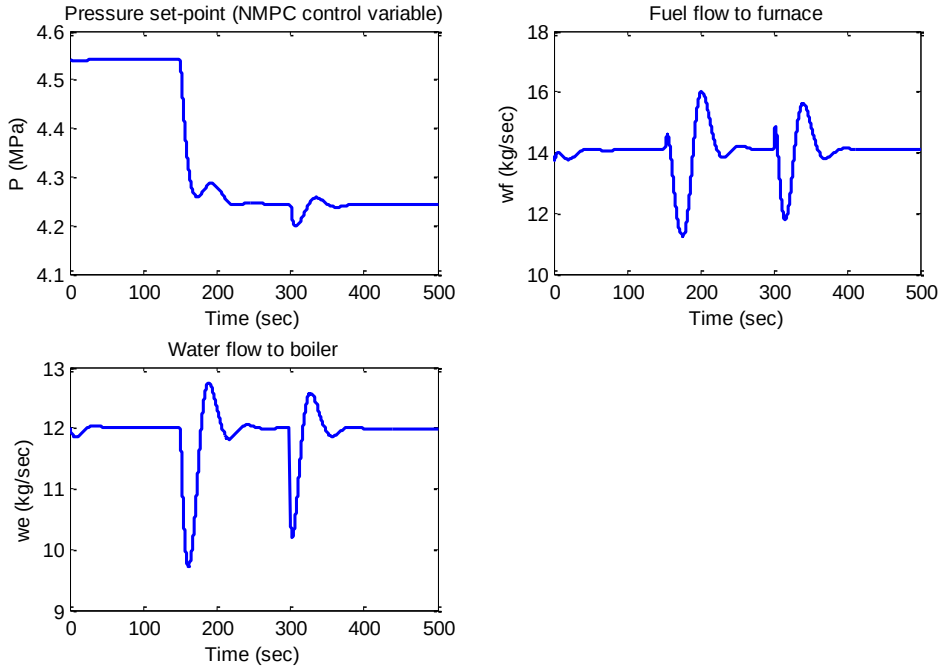


Figure 4.8: Control and manipulated variables response to output disturbance

4.6.2 NMPC for a Nonlinear Power Plant Model

4.6.2.1 State-Dependent Coefficients (SDC) Representation for the Power Plant Model

The hierarchical structure of the considered CCPP model consists of three subsystems: gas turbine with a generator, boiler (HRSG) and steam turbine with a generator. The gas turbine model used in this study is a linear model. Therefore, it is assumed that the gas turbine is controlled separately using a supervisory linear MPC, where the gas turbine outputs h_G and w_G are considered to be constant input variables to the boiler. The boiler is directly connected to a single steam turbine with a generator, to create a complete power plant. This power plant process is represented by an 11th order nonlinear first principles model. As shown in Figure 4.9, the power plant control system consists of three simple proportional integral (PI) control loops, designed to meet the set-points requirements.

In this study, it is assumed that the boiler water level is well controlled using a PID controller, as discussed in the previous section. Therefore, the NMPC will be used only to control the boiler-drum pressure, which has an interaction effect on the subsequent subsystem.

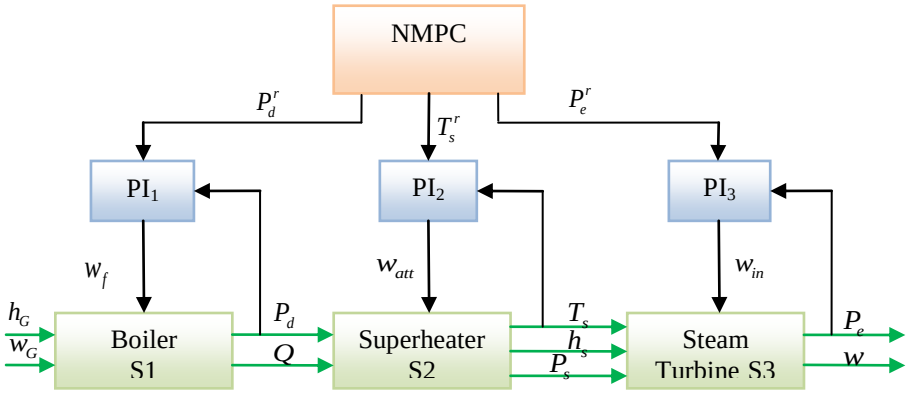


Figure 4.9: Power plant control system

The Furnace and Boiler Drum Pressure Model

As discussed in section 4.6.1, this subsystem includes both the drum pressure dynamics and the supplementary firing furnace dynamics. The boiler's continuous and discrete models given in (4.47) and (4.48) are used in this study.

The Superheater Model

In order to ensure the optimum operating heat rate and to protect the steam turbine, the optimum set-point of the boiler superheater temperature T_s^r is calculated using the supervisory centralized NMPC. Then, using this optimum set-point, the PI controller regulates the superheater outlet temperature by controlling the attemperator spray water flow w_{att}. The inputs of this module are the drum pressure P_d and the heat energy Q_{gs} from the boiler module. The outputs to the steam turbine module are the temperature of the superheater steam T_s, the specific enthalpy of superheater steam h_s, and the pressure of the superheater steam P_s.

In order to represent the superheater differential equations described in Chapter 2 in the SDC model, the differential equations are rewritten as follows:

$$\begin{aligned}
\dot{x}_2(1) &= \frac{w_v - w_s}{V_s} \\
\dot{x}_2(2) &= \frac{1}{M_s C_{st}}[Q_{gs} - k_s w_v^{0.8} x_2(2) + \frac{k_s w_v^{0.8} x_2(3)}{x_2(1) c_{ps}} - \frac{k_s w_v^{0.8} h_{ref}}{c_{ps}} + k_s w_v^{0.8} T_{ref}] \\
\dot{x}_2(3) &= \frac{1}{V_s}[k_s w_v^{0.8} x_2(2) - \frac{k_s w_v^{0.8} x_2(3)}{x_2(1) c_{ps}} + \frac{k_s w_v^{0.8} h_{ref}}{c_{ps}} - k_s w_v^{0.8} T_{ref} + \\
&\qquad w_v h_v - \frac{w_s x_2(3)}{x_2(1)} + (h_a - h_f) w_a]
\end{aligned} \tag{4.49}$$

The superheated steam temperature T_s can be expressed as follows:

$$T_s = \frac{x_2(3)}{c_{ps} x_2(1)} - \frac{h_{ref}}{c_{ps}} + T_{ref} \tag{4.50}$$

The state-space, input and output are defined as $x_2 = [\rho_s, T_{st}, X_{s1}]^T$, $\bar{u}_2 = w_a$ and $y_2 = T_s$ respectively. After appropriate substitutions, the SDC model can be represented as follows:

$$\begin{aligned} \dot{x}_2 &= \tilde{A}_{22}(\hat{x}_1, \hat{x}_2)\, x_2 + \tilde{B}_{22}(\hat{x}_2)\, \bar{u}_2 \\ y_2 &= \tilde{C}_{22}(\hat{x}_2)\, x_2 \end{aligned} \tag{4.51}$$

where:

$$\tilde{A}_{22} = \begin{bmatrix} G_{11} & G_{12} & 0 \\ G_{21} & G_{22} & G_{23} \\ G_{31} & G_{32} & G_{33} \end{bmatrix}, \qquad \tilde{B}_{22} = \begin{bmatrix} 0 \\ 0 \\ b_3 \end{bmatrix}, \qquad \tilde{C}_{22} = \begin{bmatrix} c_1 & 0 & c_2 \end{bmatrix}$$

$$G_{11} = \frac{-w_s}{V_s x_2(1)}, \qquad G_{12} = \frac{w_v}{V_s x_2(2)}, \qquad G_{21} = \frac{Q_{gs} + k_s w_v^{0.8} T_{ref}}{M_s C_{st} x_2(1)} - \frac{k_s w_v^{0.8} h_{ref}}{M_s C_{st} c_{ps} x_2(1)}$$

$$G_{22} = \frac{-k_s w_v^{0.8}}{M_s C_{st}}, \qquad G_{23} = \frac{k_s w_v^{0.8} h_{ref}}{M_s C_{st} c_{ps} x_2(1)}, \qquad G_{31} = \frac{k_s w_v^{0.8} h_{ref}}{V_s c_{ps} x_2(1)} - \frac{k_s w_v^{0.8} T_{ref}}{V_s x_2(1)} + \frac{w_v h_v}{V_s x_2(1)},$$

$$G_{32} = \frac{k_s w_v^{0.8}}{V_s}, \qquad G_{33} = \frac{-k_s w_v^{0.8}}{V_s c_{ps} x_2(1)} - \frac{w_s}{V_s x_2(1)}, \qquad c_1 = -\frac{h_{ref}}{x_2(1) c_{ps}} + \frac{T_{ref}}{x_2(1)},$$

$$c_2 = \frac{1}{x_2(1) c_{ps}}, \qquad b_3 = \frac{T(h_a - h_f)}{V_s}$$

This model is discretized using the Euler method and augmented with a PI controller, to get the following model:

$$\begin{aligned} x_2(k+1) &= A_{22}(\hat{x}_1, \hat{x}_2)\, x_2(k) + B_{22}(\hat{x}_2)\, u_2(k) \\ y_2(k) &= C_{22}(\hat{x}_2)\, x_2(k) \end{aligned} \tag{4.52}$$

where:

$$A_{22}=\begin{bmatrix} 1 & -Tc_1 & 0 & -Tc_2 \\ 0 & TG_{11}+1 & TG_{12} & 0 \\ 0 & TG_{21} & TG_{22}+1 & TG_{23} \\ Tb_3k_{i2} & T(G_{31}-b_3c_1k_{p2}) & TG_{32} & T(G_{33}-b_3c_2k_{p2})+1 \end{bmatrix}$$

$$B_{22}=\begin{bmatrix} T \\ 0 \\ 0 \\ Tb_3k_{p2} \end{bmatrix}, \qquad C_{22}=\begin{bmatrix} 0 & c_1 & 0 & c_2 \end{bmatrix}$$

k_{p2} and k_{i2} are the PI controller gains, and T is the sampling time. In this model, the state space vector is defined as $x_2=[x_{c2}, x_2(1), x_2(2), x_2(3)]^T$, where x_{c2} is the PI controller state. The input is defined as the temperature set-point $u_2=T_s$.

Steam Turbine and Generator Model

Once the future electrical power P_e^r set-point trajectory is available from the centralized NMPC, the PI controller regulates the steam flow w_{in} from the boiler to produce the desired electrical power P_e, while the steam pressure is regulated in the boiler.

In order to represent the steam turbine and generator differential equations described in Chapter 2 in the SDC model, the differential equations are rewritten as follows:

$$\begin{aligned} \dot{x}_3(1) &= \frac{(w_{in}+7.44)}{V}-\frac{x_3(2)}{V} \\ \dot{x}_3(2) &= \frac{(w_{in}+7.44)}{\tau_s}-\frac{x_3(2)}{\tau_s} \\ \dot{x}_3(3) &= \frac{(w_{in}+7.44)h_{in}}{V}-\frac{x_3(2)x_3(2)}{x_3(1)V} \\ \dot{x}_3(4) &= \frac{\omega_0}{2H_{eq}}\left[c_1x_3(2)-P_e-D_{eq}(x_3(4)-\omega_0)\right] \end{aligned} \tag{4.53}$$

As discussed in Chapter 2, the state-space, input and output are defined as $x_3 = [\rho_o, w_{ou}, x_o, \omega_g]^T$, $u_3 = w_{in}(kg/\sec)$ and $y_3 = P_m\ (W)$, respectively. After appropriate substitutions, the SDC model can be represented as follows:

$$\begin{aligned} \dot{x}_3 &= \tilde{A}_{33}(\hat{x}_2, \hat{x}_3)\, x_3 + \tilde{B}_{33}(\hat{x}_3)\bar{u}_3 \\ y_3 &= \tilde{C}_{33}(\hat{x}_3)\, x_3 \end{aligned} \tag{4.54}$$

where:

$$\tilde{A}_{33} = \begin{bmatrix} s_{11} & s_{12} & s_{13} & 0 \\ s_{21} & s_{22} & s_{23} & 0 \\ s_{31} & 0 & 0 & 0 \\ s_{41} & s_{42} & 0 & s_{44} \end{bmatrix}, \quad \tilde{B}_{33} = \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ 0 \end{bmatrix}, \quad \tilde{C}_{33} = \begin{bmatrix} 0 & C_1 & 0 & 0 \end{bmatrix}$$

$$s_{11} = \frac{\alpha_1 7.44}{V x_3(1)}, \qquad s_{12} = \frac{-1}{V}, \qquad s_{13} = \frac{(1-\alpha_1)7.44}{V x_3(3)}, \qquad s_{21} = \frac{\alpha_2 7.44}{\tau_{aus} x_3(1)},$$

$$s_{22} = \frac{-1}{\tau_{aus}}, \qquad s_{23} = \frac{(1-\alpha_2)7.44}{\tau_{aus} x_3(3)}, \qquad s_{31} = \frac{7.44 h_{in} x_3(1) - x_3(2) x_3(3)}{V x_3(1)^2},$$

$$s_{41} = \frac{-z_f (P_e - D_{eq})}{x_3(1)}, \qquad s_{42} = z_f C_1, \qquad s_{44} = -z_f D_{eq}, \qquad C_1 = \frac{2.2\left|n\Delta hi\right|}{1.144 \times 10^7},$$

$$z_f = \frac{\omega_o}{2H_{eq}}, \qquad B_1 = \frac{1}{V}, \qquad B_2 = \frac{1}{\tau_s}, \qquad B_3 = \frac{h_{in}}{V}$$

α_1 and α_2 are real numbers used to provide design flexibility as discussed in section 4.3, and chosen using simulation to enhance the observability as 0.4328 and 0.5672, respectively.

This model is discretized using the Euler method and augmented with a PI controller, to get the following model:

$$\begin{aligned} x_3(k+1) &= A_{33}(\hat{x}_2, \hat{x}_3)\, x_3(k) + B_{33}(\hat{x}_3)\, u_3(k) \\ y_3(k) &= C_{33}(\hat{x}_3)\, x_3(k) \end{aligned} \tag{4.55}$$

where:

$$A_{33}=\begin{bmatrix} 1 & 0 & -TC_1 & 0 & 0 \\ TB_1k_{i3} & Ts_{11}+1 & T(s_{12}-B_1C_1k_{p3}) & Ts_{13} & 0 \\ TB_2k_{i3} & Ts_{21} & T(s_{22}-B_2C_1k_{p3}) & Ts_{23} & 0 \\ TB_3k_{i3} & Ts_{31} & -TB_3C_1k_{p3} & 1 & 0 \\ 0 & Ts_{41} & Ts_{42} & 0 & Ts_{44}+1 \end{bmatrix},$$

$$B_{33}=\begin{bmatrix} T \\ TB_1k_{p3} \\ TB_2k_{p3} \\ TB_3k_{p3} \\ 0 \end{bmatrix}, \quad C_{33}=\begin{bmatrix} 0 & 0 & C_1 & 0 & 0 \end{bmatrix}$$

k_{p3} and k_{i3} are the PI controller gains, and T is the sampling time. In this model, the state space vector is defined as $x_3=[x_{c3}, x_3(1), x_3(2), x_3(3), x_3(4)]^T$, where x_{c3} is the PI controller state. The input is defined as the electrical power demand $u_3=P_e$.

The centralized complete discrete model can be expressed as follows:

$$\begin{aligned} x(k+1) &= A(\hat{x})\,x(k)+B(\hat{x})\,u(k) \\ y(k) &= C(\hat{x})\,x(k) \end{aligned} \tag{4.56}$$

where:

$$x=\begin{bmatrix} x_1^T & x_2^T & x_3^T \end{bmatrix}^T, \qquad y=\begin{bmatrix} y_1 & y_2 & y_3 \end{bmatrix}^T, \qquad u=\begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix}^T,$$

$$A(x)=\begin{bmatrix} A_{11} & 0 & 0 \\ 0 & A_{22} & 0 \\ 0 & 0 & A_{33} \end{bmatrix}, \; B(x)=\begin{bmatrix} B_{11} & 0 & 0 \\ 0 & B_{22} & 0 \\ 0 & 0 & B_{33} \end{bmatrix}, \; C(x)=\begin{bmatrix} C_{11} & 0 & 0 \\ 0 & C_{22} & 0 \\ 0 & 0 & C_{33} \end{bmatrix}$$

Simulation Results

In this simulation, the PID set points are manipulated using NMPC control to achieve a better performance. Hence, PIDs are mainly used to regulate the systems, and NMPC is used to improve tracking and disturbance rejection, as well as to minimise the economic performance index. In this simulation, the SDDRE Kalman filter

described in section 4.5.2.2 is used to estimate the system states. The initial condition of states used in the simulation is as follows:

$$x(0) = [0, 3.9984\times10^{5}, 0.4505, 10.87, 4.5417, 0, 13.6614, 7.3707\times10^{2}, 4.5243\times10^{7}, 0, 1.5982, 10.459, 4.9079\times10^{6}, 1]$$

The best values for the PI gains are found using the Multivariable PID tuning methods described in Chapter 3. The PI tuning gains for the three loops are K_p=[7.5131,-0.48197,0.8396] and K_i=[0.1878,-0.0172,0.8396]. The NMPC controller has been configured by choosing the weights on the tracking error Q= [45, 0.01, 0.5] and the weights on the control increments S= [43, 10, 50]. The prediction horizon for the optimization problem is N_p= 10, with a time step of T=0.02 sec. In this simulation, the process noise is assumed to be $\zeta \sim N(0, 0.001)$ and the sensor noise is $\eta \sim N(0, 0.01)$.

Implementing the NMPC iterative algorithm described in section 4.4.3.5, where the required error norm for the control input to be satisfied is chosen as $\sigma = 0.001$, the maximum iteration limit is defined as $i_{\max} = 120$. Figure 4.10 shows the response of the boiler superheated temperature to set-point changes. It shows also the number of iterations required for convergence. It is clear that the norm of the error is satisfied during the whole trajectory. It can also be seen that, during the transition region, more iterations are required for convergence due to changes in the system states.

Figure 4.11 depicts the closed-loop performance of the NMPC algorithm when process and output noises are added to the system. The simulation results show the strong robustness of performance against noises. The evolution of the real states and their estimates obtained by SDDRE estimators for the selected states are presented in Figure 4.12. The simulation results show that the SDDRE filter can attain a good performance.

In order to maintain a high level of system safety, output constraints are introduced in this simulation. The minimum and maximum output constraints introduced are Y_{min}= [3.0MPa, 715.8^{o}K, 0.78Pu-W] and Y_{max}=[6.0MPa, 718^{o}K, 0.82Pu-W],

respectively. Figure 4.13 shows the steam turbine output power response to set point changes in boiler superheated temperature using constrained and unconstrained NMPC. The constrained and unconstrained superheated temperature responses to boiler pressure set point changes are shown in Figure 4.14 . From these figures, it can be clearly observed that the imposed output constraints are satisfied using the proposed constrained DMPC algorithms.

Figure 4.15 shows the boiler pressure and superheated temperature response to the output step disturbance of 0.1 MPa on the boiler pressure at 150sec. As shown in this figure, the disturbance can be eliminated efficiently through the proposed NMPC algorithm.

Simulation result comparing the performance of supervisory NMPC and the classical PI controllers is shown in Figure 4.16. It can be seen that the NMPC controllers provide better response than PI controllers. NMPC has smaller overshoot and smoother controller action than PID controllers.

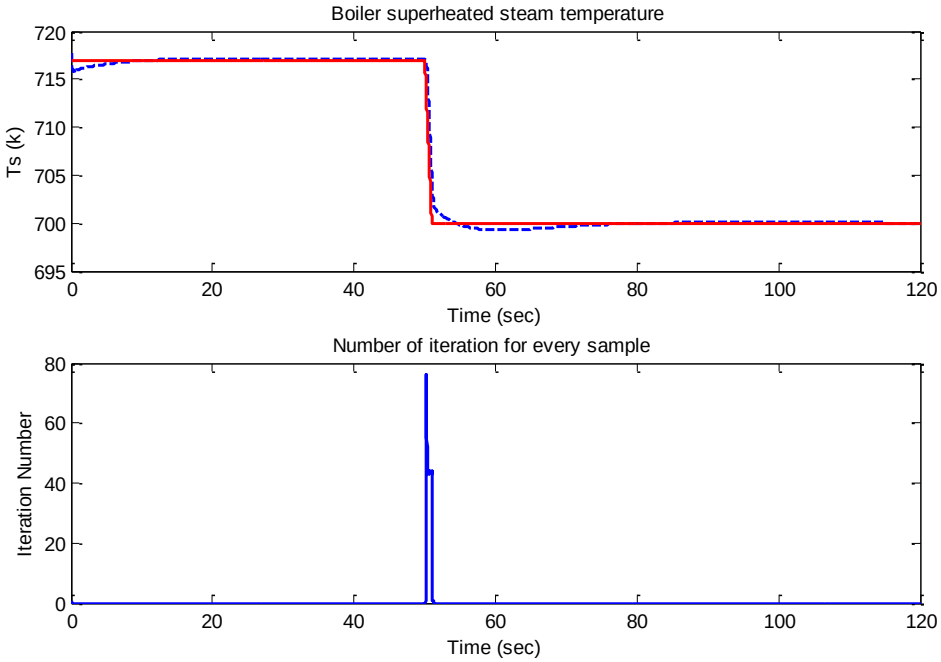


Figure 4.10: Number of iterations required for convergence

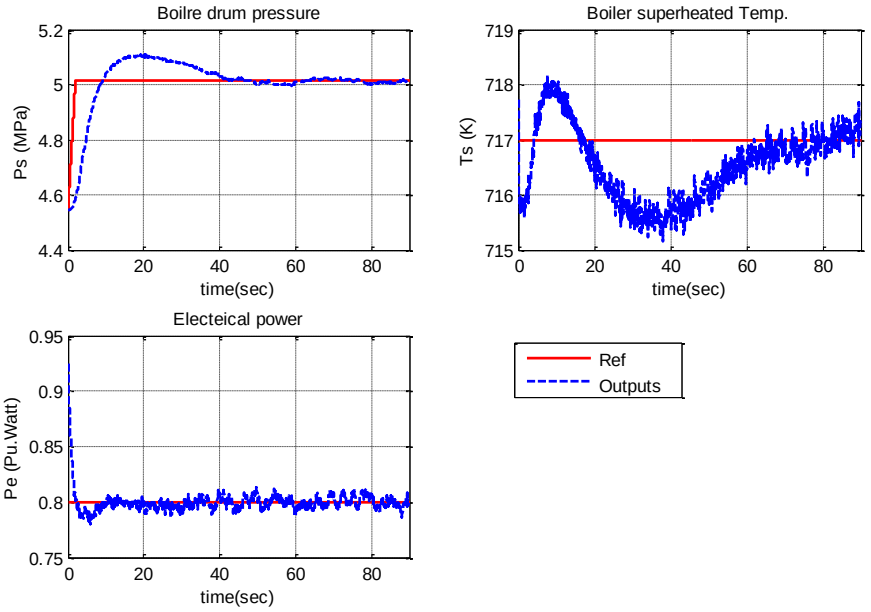


Figure 4.11: Outputs response to set-point change with states and outputs noises

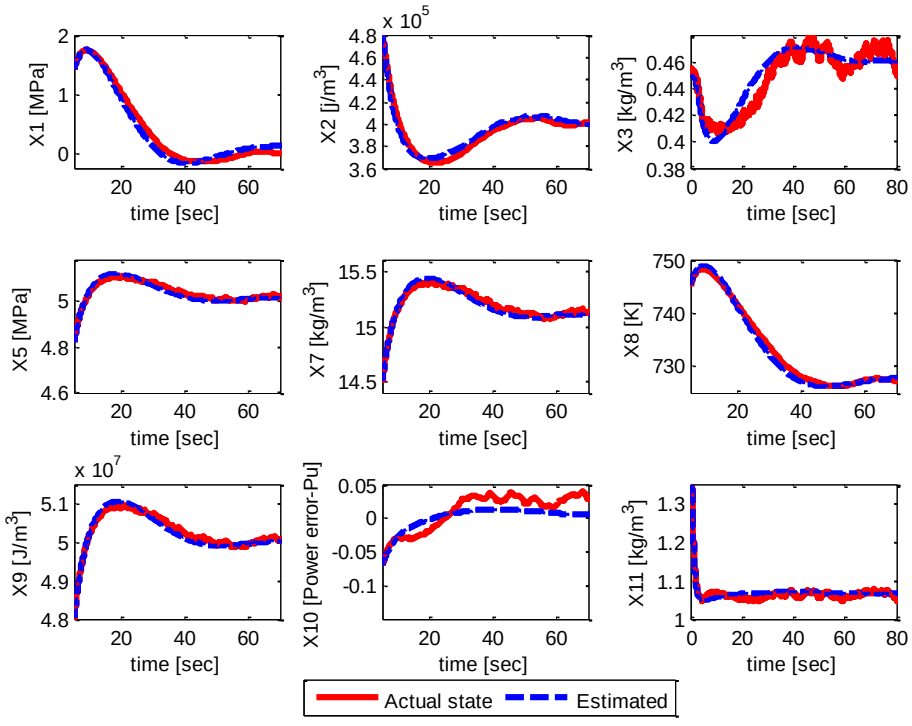


Figure 4.12: System true states and their estimates

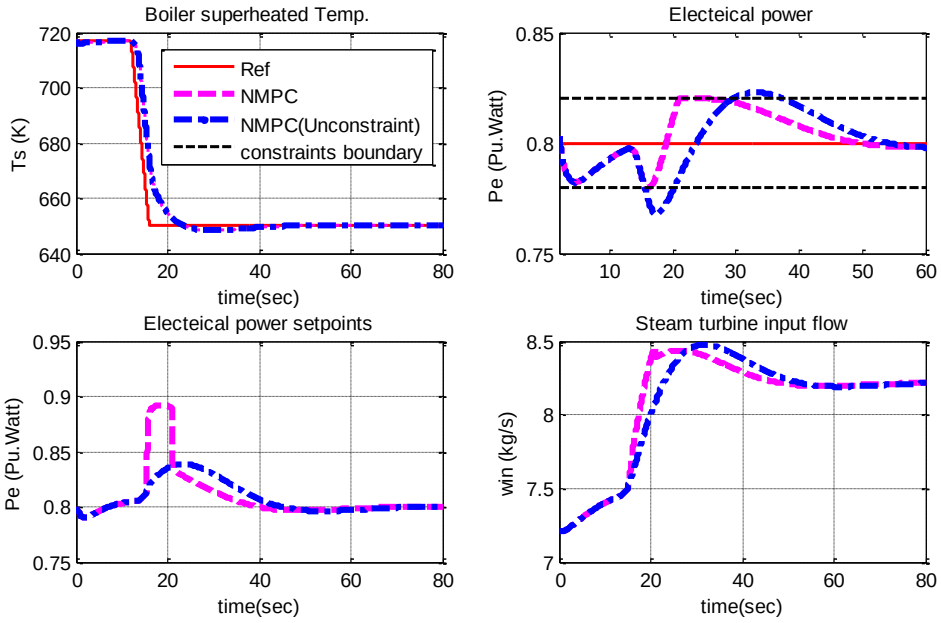


Figure 4.13: Steam turbine responses to set-point changes in superheated temperatures, using constrained and unconstrained NMPC

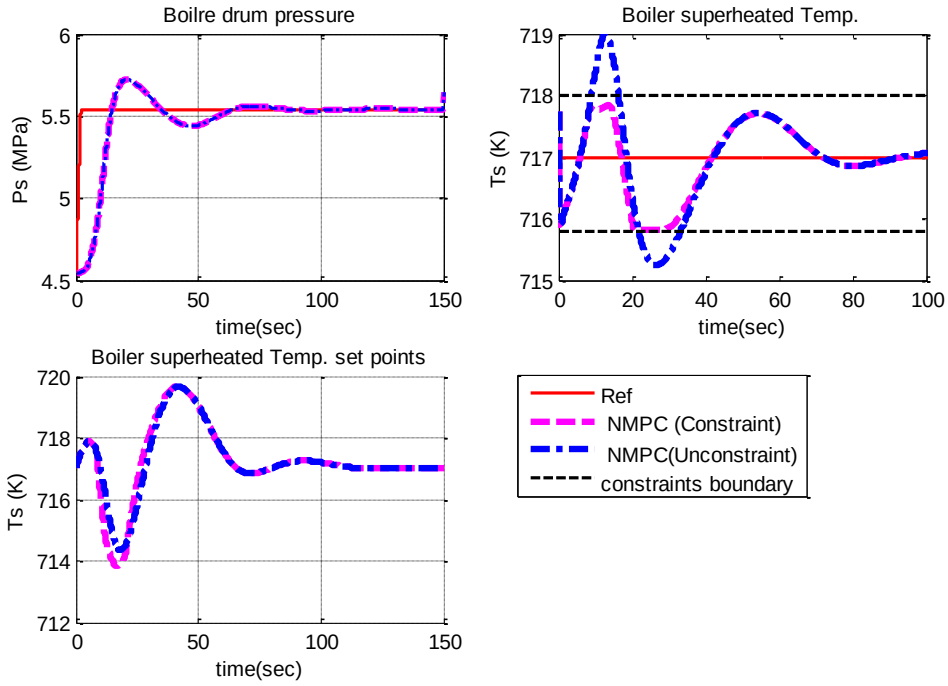


Figure 4.14: Superheated temperature responses to pressure set-point changes, using constrained and unconstrained NMPC

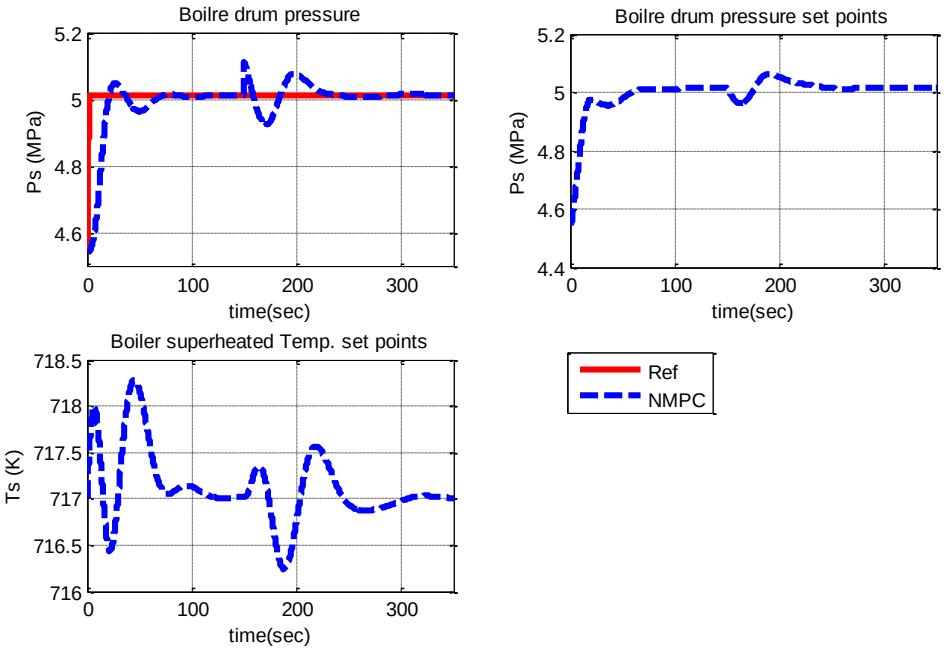


Figure 4.15: Boiler pressure and superheated temperature responses to output disturbance on boiler pressure at 150sec

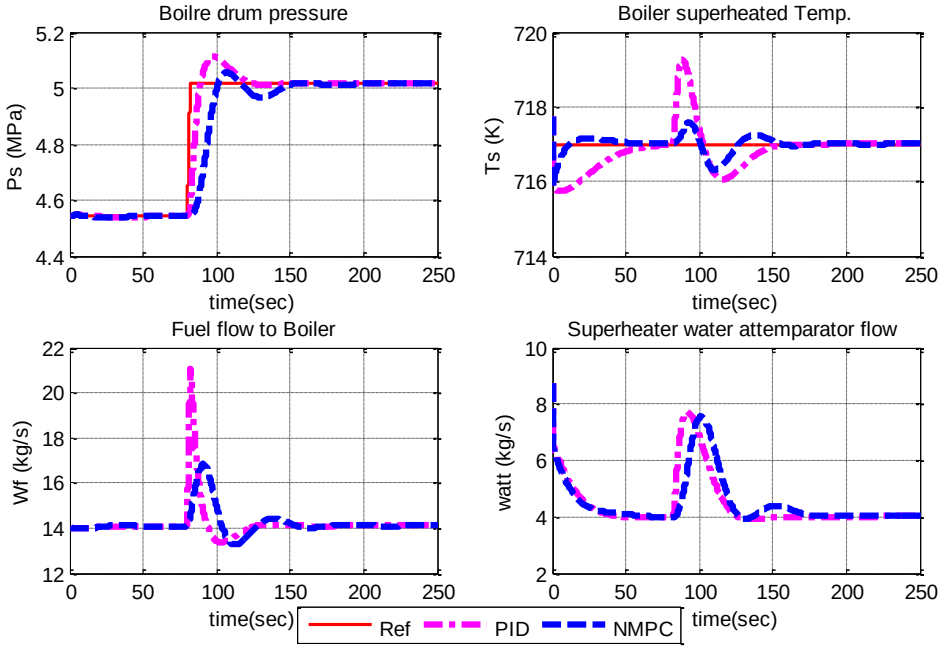


Figure 4.16: Boiler pressure and superheater temperature responses to set-point changes in pressure using NMPC and PI controllers

4.7 Conclusions

This chapter has presented an efficient supervisory NMPC algorithm based on the state-dependent approach for the industrial power plant process. The non-linear power plant system was represented in a state-dependent form to provide global nonlinear behaviour. The nonlinearity of the model and the lack of measurement were handled through nonlinear state estimation using an SDDRE Kalman filter. The NMPC was used in the second layer, to tune the performance of the PID controllers. The simulation results showed that the NMPC controller had a good tracking and disturbance rejection performance, and allowed the inclusion of output constraints.

5. Sequential Nonlinear Distributed Model Predictive Control (DMPC) and Estimation

5.1 Introduction

Modern industrial processes are generally composed of different subsystems, which are interconnected and characterized by significant interactions. At the same time, due to ever tightening environmental regulations coupled with high performance requirements, modern control systems are becoming more and more complex. For these processes, different control solutions can be developed, such as centralized, decentralized and distributed control techniques.

As discussed previously in Chapter 4, the centralized MPC is applied to control the large-scale CCPP, where its formulation is based on a centralized single agent to solve a single optimization problem for the entire system model, which requires a significant computation burden. Another disadvantage of using centralized MPC for large-scale systems is the non-scalability. This is due to the size of control model and the need to retune the centralized controller and rebuild the control model on every change in the system configuration as a result, for example to maintenance or malfunctions. For all these reasons, distributed and decentralized MPC control schemes have been developed for industrial processes, for example (Katebi and Johnson, 1997).

Distributed model predictive control (DMPC) strategies for large-scale industrial process described in literature can be classified into two groups, namely decentralized and distributed MPC, as shown in Figure 5.1. A comprehensive literature review of these methods can be found in survey papers (Rawlings and Stewart, 2008; Negenborn et al., 2009; Scattolini, 2009; Al-Gherwi et al., 2011; Christofides et al., 2012).

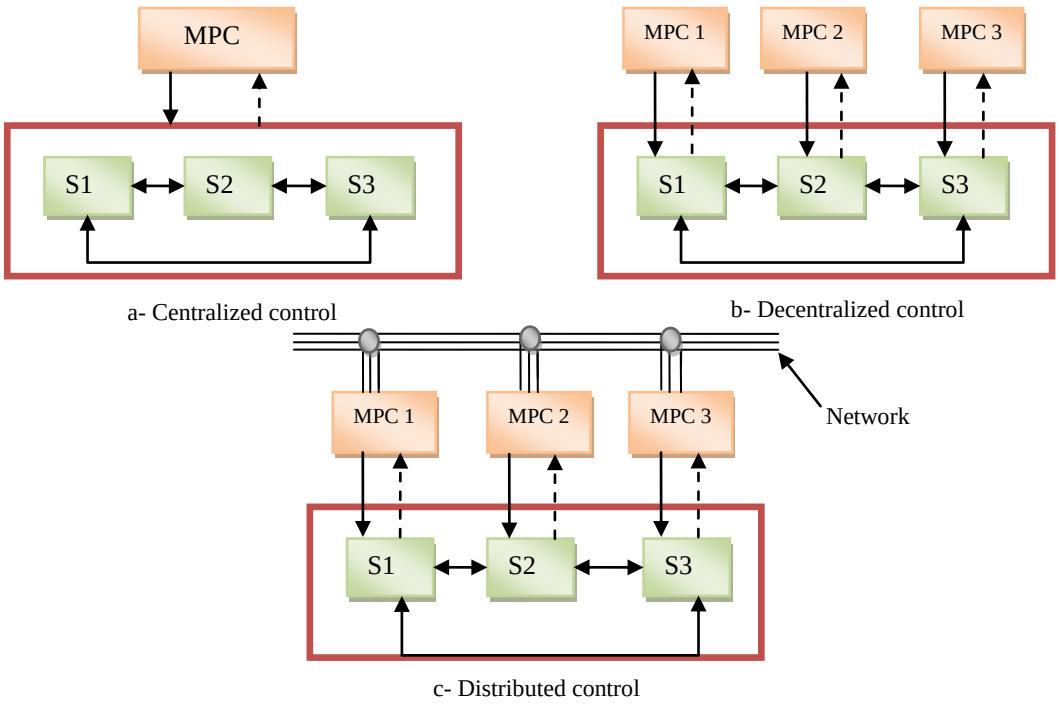


Figure 5.1: Comparison between different control strategies

Most of the distributed strategies reported in literature employ linear models to predict future behavior of the process in order to achieve optimal or sub-optimal closed-loop performance (Scheu and Marquardt, 2011). This chapter presents a novel methodology for nonlinear distributed DMPC for a large-scale power plant process by using sequential DMPC algorithms. The proposed distributed control design requires one direction communication between agents. This control design is based on supervisory NMPC, as discussed in Chapter 4, which uses a state-dependent nonlinear model in order to reduce the complexity of the online optimization problem and avoid linearization. In addition, this chapter presents a distributed nonlinear estimation algorithm based on the state-dependent Riccati equation (SDRE) estimation method discussed in Chapter 4.

Decentralized and distributed control strategies are summarized in the following sections.

5.1.1 Decentralized model predictive control

In the decentralized MPC structure, the overall system under control is composed of a number of subsystems, each one locally controlled with an MPC algorithm, where there is no communication between controllers and all interactions between subsystems are neglected, as shown in Figure 5.1b. In this case, each MPC solves a local cost function that includes only the controlled variables assigned for the specific subsystem without considering the solutions of the other controllers. When the interactions are strong, the fully decentralized controller can lead to instability and performance deterioration (Cui and Jacobsen, 2002). Magni and Scattolini (2006) propose a fully decentralized MPC algorithm for nonlinear systems. In their approach the closed-loop stability is ensured by the inclusion of a contractive constraint in the formulation of the optimization problem. However, the fully decentralized approach requires a conservative solution due to the absence of information exchange between controllers (Alessio et al., 2011).

5.1.2 Distributed model predictive control (DMPC)

In distributed control structure, the original large scale system is replaced by a number of interconnected subsystems, where the different MPC controllers exchange information through a network to coordinate their actions, as shown in Figure 5.1c. In previous work on distributed MPC, Katebi and Johnson (1997) proposed a decentralized filtering and control scheme for generalised predictive control in which a high level global coordinator was used to iteratively find an optimal solution. Jia and Krogh (2001) developed a DMPC strategy in which the controllers exchange their predictions by communication to coordinate their actions and improve performance. Recently, Rawlings and Stewart (2008), Scattolini (2009) and Stewart et al.(2011) have classified distributed MPC strategies found in literature into two types: communication-based DMPC and cooperation-based DMPC.

5.1.2.1 Communication-based (non-cooperative) DMPC

The first type of DMPC strategy is communication-based DMPC strategy, in which each local MPC solves its local cost function and exchanges predicted state and input trajectory information between MPCs by communication (Venkat et al., 2005). In a

communication-based DMPC architecture, all distributed controllers can be evaluated in parallel as in Vaccarini et al. (2009) for example, or in sequential order as in Richards and How (2007), or based on neighbourhood optimization as in Zhang and Li (2007).

In literature, there are many types of distributed MPC strategies that take into account interactions between subsystems. For example, in Camponogara et al.(2002) the system under control is composed of a number of unconstrained linear discrete-time subsystems, where the dynamical coupling between neighbouring states is modelled in prediction through a disturbance signal. In this scheme, the closed loop stability is proved by introducing a contractive constraint on the state prediction norm in each local MPC problem.

In Richards and How (2007) a sequential non-cooperative DMPC is proposed for a class of decoupled systems. In this strategy, controllers are evaluated in sequence, once at each sampling time, and one-directional communication occurs between consecutive distributed controllers. In this architecture, as shown in Figure 5.2 each MPC controller only sends its future input trajectory and the future input trajectories received are sent to the next MPC controller, which reduces computation time. The advantage of this scheme is that it reduces numerical and communication effort compared to a centralized solution to the optimization problem.

An alternative approach to sequential DMPC architecture is to evaluate all distributed controllers in parallel. In this architecture, each distributed controller communicates with all other controllers to exchange future input trajectories (Liu et al., 2010). Al-Gherwi et al. (2011) proposed a robust linear DMPC evaluated in parallel and based on an LMI approach that explicitly accounts for parametric uncertainty in the model. In order to enhance global control performance while reducing the communication burden among subsystems, Zhang and Li (2007) developed a new strategy based on neighbourhood optimization, in which the optimization objective of each local subsystem considers the performance of its neighbours.

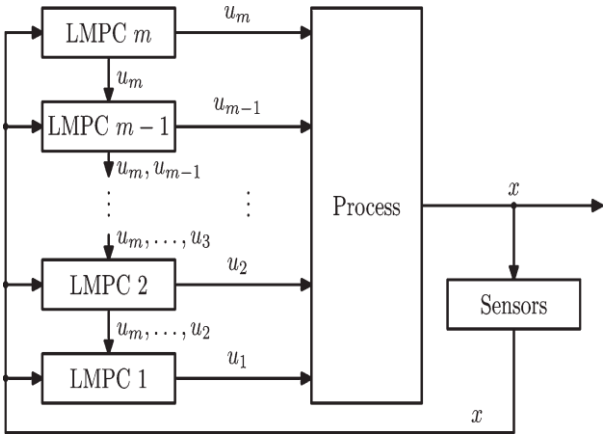


Figure 5.2: Sequential DMPC scheme (Chen et al., 2010)

The advantage of communication-based DMPC control over fully decentralized control is that the control agents have accurate knowledge of the effects of all other agents on their local objectives. However, if the overall system is composed of strongly interacting subsystems, non-cooperative DMPC control can destabilize the plant and the performance may be worse than with decentralized control (Rawlings and Stewart, 2008).

5.1.2.2 Cooperation-based DMPC

The second type of DMPC strategy is cooperation-based distributed MPC strategy, which was developed by Venkat et al. (2005). In this strategy, each controller optimizes a weighted sum of all local cost functions. It is based on negotiations among DMPC agents, where at each time step a sequence of iterations is taken before computing and implementing the input vector. The solution can achieve a global (Pareto) optimal control decision similar to that obtained by centralized MPC if convergence is satisfied. In addition, closed-loop stability and feasibility can still be ensured if the procedure is stopped at any intermediate iterate (Stewart et al., 2010). Stewart et al. (2010) proposed a co-operative linear DMPC strategy in which the subsystem controllers optimize the same objective function in parallel. In this strategy, the closed-loop performance converges to the corresponding centralized control system as the iteration number increases. It can also satisfy hard input constraint and provide nominal stability for plants with even strongly interacting subsystems.

In Liu et al (2009; 2010) a Lyapunov-based iterative DMPC for nonlinear systems subject to asynchronous and delayed measurements was presented. In this strategy each agent has access to a full system model, which is sometimes not possible to obtain or needs a high computation effort in large scale systems (Maestre et al., 2011).

A cooperation-based DMPC strategy could achieve good closed-loop performance compared to communication-based DMPC. However, it requires much more communication resources and computational complexity (Zheng et al., 2011).

5.2 Sequential Supervisory DMPC Architecture

One way of simplifying the MPC solution for large-scale systems is to exploit the architecture or structure of the process. Many industrial processes have a serial or sequential structure, where the upstream sub processes affect only the downstream ones. This is schematically shown in Figure 5.3. For example, the combined cycle power plant process can be considered as a sequential process by assuming that the temperature of water coming from the condenser is constant and there is a small effect on boiler pressure due to the change in the steam turbine steam flow. In this process, a gas turbine generates electricity and the waste exhaust gas is recovered in a recovery boiler that produces steam, which is then expanded in a condensing turbine to generate electricity, as shown in Figure 5.4. This process plant can naturally be decomposed into three interconnected subsystems: gas turbine, HRSG boiler, and steam turbine. Furthermore, each subsystem can be decomposed into other subsystems. For example, the recovery boiler can be decomposed into drum pressure and superheater subsystems.

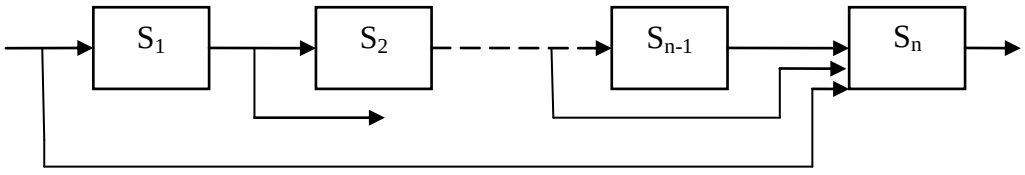


Figure 5.3: A sequential system

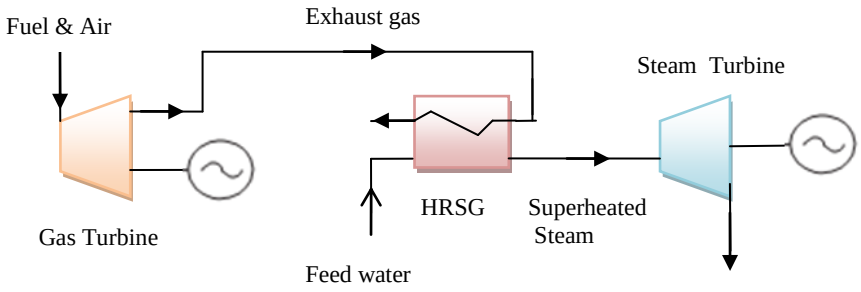


Figure 5.4: Combined cycle power plant process

After decomposing the large-scale system into interconnected subsystems, the MPC controllers exchange information in one directional communication through a network to coordinate their actions. The goal of the decomposition is to reduce the complexity of the optimization problem. In order to reach global optimality, the interactions between the subsystems are included and solved using the Nash optimality concept (Nash, 1951). Since information on interactions is exchanged through the network, each agent can solve its local optimal problem provided that the previous agent's optimal solution is known. The optimal solution will converge if all agents' terminal conditions are satisfied and then the whole system will arrive at the Nash equilibrium.

The proposed hierarchical structure control strategy consists of two levels: an existing plant PID classical control solution, and above this, a supervisory nonlinear predictive control (NMPC) optimization level, as shown in Figure 5.5. The supervisory DMPC controllers are responsible for providing optimal reference trajectories to the local controllers of each corresponding subsystem. The advantage of this structure is that the NMPC algorithm is an add-on to the existing PID control structure. Therefore, it does not interfere with an existing (well-proven) control system. Furthermore, the model used in the DMPC design is therefore open-loop stable. A decentralized nonlinear Kalman filter (SDKF) is used to estimate subsystem states.

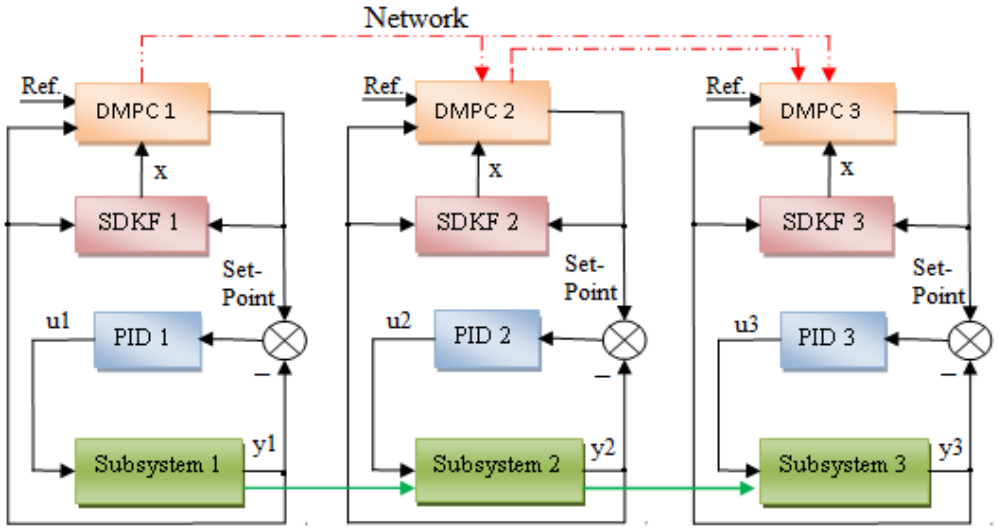


Figure 5.5: Supervisory DMPC structure

5.2.1 Nonlinear state-dependent model representation for a nonlinear sequential distributed system

Consider a nonlinear process system described by the following state-space model

$$x_{K+1} = f(x_k, u_k, \xi_k) \tag{5.1}$$

$$y_k = g(x_k, u_k, \eta_k) \tag{5.2}$$

where x_k is the process state vector, u_k is the control input vector, y_k is the process output, ξ_k is the process noise, and η_k is the measurement noise. Let the system under control be composed of the interconnection of *n* local subsystems described by the state-dependent nonlinear discrete-time model as:

$$x_i(k+1) = A_{ii}(x_i)x_i(k) + B_{ii}(x_i)u_i(k) + \omega_i(k) + \xi_i(k) \tag{5.3}$$

$$y_i(k) = C_{ii}(x_i)x_i(k) + v_i(k) + \eta_i(k) \tag{5.4}$$

where k is discrete time, $x_i \in R^{n_i}, u_i \in R^{m_i}, y_i \in R^{r_i}$ for *i*=1,...,*n*; $x_i(k)$ is the subsystem state vector; and $y_i(k)$ is the output vector for the *i*-th subsystem. $u_i(k)$ is the control input for subsystem *i* (**§**$_i$). $\xi_i(k)$ is a zero mean process noise, and $\eta_i(k)$ is a zero mean

measurement noise for subsystem $\S_i$. Subsystem state-dependent matrices $A_{ii}(x_i)$, $B_{ii}(x_i)$ and $C_{ii}(x_i)$ are formulated such that the LTV system is locally observable and controllable. State and output interactions vectors ω_i and v_i are given by

$$\omega_i(k) = \sum_{j=1(j\neq i)}^{n} A_{ij}(x_j)x_j(k) + \sum_{j=1(j\neq i)}^{n} B_{ij}(x_j)u_j(k) \tag{5.5}$$

$$v_i(k) = \sum_{j=1(j\neq i)}^{n} C_{ij}(x_j)x_j(k) \tag{5.6}$$

The total sequential system has the following lower triangular system matrices:

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{bmatrix}_{k+1} = \begin{bmatrix} A_{11}(\hat{x}_1) & 0 & 0 & 0 & 0 \\ A_{21}(\hat{x}_1) & A_{22}(\hat{x}_2) & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & 0 \\ A_{n\times 1}(\hat{x}_1) & \cdots & \cdots & \cdots & A_{n\times n}(\hat{x}_n) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix}_k +$$

$$\begin{bmatrix} B_{11}(\hat{x}_1) & 0 & 0 & 0 & 0 \\ B_{21}(\hat{x}_1) & B_{22}(\hat{x}_2) & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & 0 \\ B_{m\times 1}(\hat{x}_1) & \cdots & \cdots & \cdots & B_{m\times m}(\hat{x}_n) \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{m-1} \\ u_m \end{bmatrix}_k + \begin{bmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_{n-1} \\ \xi_n \end{bmatrix}_k \tag{5.7}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ \vdots \\ y_r \end{bmatrix}_{k+1} = \begin{bmatrix} C_{11}(\hat{x}_1) & 0 & 0 & 0 & 0 \\ C_{21}(\hat{x}_1) & C_{22}(\hat{x}_2) & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & 0 \\ C_{r\times 1}(\hat{x}_1) & \cdots & \cdots & \cdots & C_{r\times r}(\hat{x}_n) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix}_k + \begin{bmatrix} \eta_1 \\ \eta_2 \\ \vdots \\ \eta_{r-1} \\ \eta_r \end{bmatrix}_k \tag{5.8}$$

5.3 Sequential Distributed Algorithms

The sequential DMPC problem is the minimizing of the local cost function for each subsystem subject to the model constraints, including the interactions from previous subsystems, where the output and current state of a subsystem $\S_i$ (i=2…n) and the

output and states for all previous subsystems, obtained via a communication network, are used to calculate the control input trajectory of S_i. Therefore, the solution at each subsystem comprises a feedback and a feed forward component.

Two sequential algorithms are proposed in this thesis. The difference between these algorithms is related to how the interactions between subsystems in the DMPC solution are considered.

5.3.1 DMPC-Algorithm 1: Interactions from previous subsystems are considered as input disturbances

In this algorithm, the nonlinear model for each subsystem S_i is represented as in equations (5.3) and (5.4). Then, the DMPC optimization problem is solved for any S_i, while considering the interactions from all previous subsystems as disturbances.

Subsystem 1:

In the sequential structure, the optimization of the subsystem process 1 is independent of the rest of the system, where the state-dependent model can be expressed as:

$$\begin{aligned} x_1(k+1) &= A_{11}(\hat{x}_1)\,x_1(k) + B_{11}(\hat{x}_1)\,u_1(k) \\ y_1(k) &= C_{11}(\hat{x}_1)\,x_1(k) \end{aligned} \tag{5.9}$$

Therefore, the state-dependent NMPC method presented in the previous chapter can be used to calculate the control input U_1^*. Then, the control input trajectory and the state prediction trajectory are transmitted through a network to the next subsystem.

Subsystem i (S_i):

For any subsystem S_i, the cost function J_i is minimised subject to the dynamics of subsystem S_i and using all of the previous control law $U_i^*(k, N_p - 1 \mid k)$ and the state prediction $\hat{X}_i(k+1, N_p \mid k)$ of the previous control agents which were received from the network. The system model for any subsystem S_i can be represented as follows:

$$x_i(k+1) = A_{ii}(x_i(k))x_i(k) + B_{ii}(x_i(k))u_i(k) + w_i(k) \tag{5.10}$$

$$y_i(k) = C_{ii}(x_i(k))x_i(k) + v_i(k) \tag{5.11}$$

where:

$$w_i(k) = \sum_{j=1}^{i-1} A_{ij}(\hat{x}_j^*(k))\hat{x}_j^*(k) + \sum_{j=1}^{i-1} B_{ij}(\hat{x}_j^*(k))u_j^*(k) \tag{5.12}$$

$$v_i(k) = \sum_{j=1}^{i-1} C_{ij}(\hat{x}_j^*(k))x_j^*(k) \tag{5.13}$$

5.3.1.1 Integral action:

As discussed in Chapter 3, the discrete model equations (5.10) and (5.11) are augmented by a constant output disturbance model in order to guarantee asymptotic rejection of output disturbances and to achieve an offset-free performance. The states and the additional integrating disturbance are estimated from the plant measurement using a Kalman filter. The augmented state-space system for the distributed model can be represented as follows:

$$\begin{bmatrix} x_i \\ d_i \end{bmatrix}_{k+1} = \begin{bmatrix} A_{ii}(x_i) & 0 \\ 0 & I \end{bmatrix}_k \begin{bmatrix} x_i \\ d_i \end{bmatrix}_k + \begin{bmatrix} B_{ii}(x_i) \\ 0 \end{bmatrix}_k u_i(k) + \begin{bmatrix} w_i \\ 0 \end{bmatrix}_k \tag{5.14}$$

$$y_i(k) = \begin{bmatrix} C_{ii}(x_i(k)) & I \end{bmatrix} \begin{bmatrix} x_i \\ d_i \end{bmatrix}_k + v_i(k) \tag{5.15}$$

5.3.1.2 Interactions prediction calculations:

The following procedure is employed:

1. It is assumed that for any agent *i* at time *k*, all the previous interaction states vectors $X_j^*(k+1, N_p \mid k-1)$ are given.
2. Calculate the interactions matrices $A_{ij}(k+l/k-1)$, $B_{ij}(k+l/k-1)$ and $C_{ij}(k+l/k-1)$ (where l = 1,…N_p and j = 1,…, i-1)

3. Use the control inputs $U_j^*(k, N_p - 1 | k-1)$ to calculate $W_i(k, N_p - 1 | k-1)$ and $V_i(k, N_p - 1 | k-1)$ as follows:

$$W_i = \sum_{j=1(\neq i)}^{i-1} diag\{\bar{A}_{ij}(k+1, N_p | k-1)\}\, \hat{X}_j^*(k+1, N_p | k-1) + \sum_{j=1(\neq i)}^{i-1} diag\{\bar{B}_{ij}(k+1, N_p | k-1)\} U_j^*(k, N_p - 1 | k-1) \tag{5.16}$$

$$V_i = \sum_{j=1(\neq i)}^{i-1} diag\{\bar{C}_{ij}(k+1, N_p | k-1)\}\, \hat{X}_j^*(k+1, N_p | k-1) \tag{5.17}$$

where:

The interaction prediction state $X_j^*(k+1, N_p | k-1)$ and control inputs $U_j^*(k, N_p - 1 | k-1)$ are calculated and transmitted at time *k-1* from previous agents and defined as follows:

$$X_j(k+1, N_p | k-1) = \left[x_j^T(k+1 | k-1)\; x_j^T(k+2 | k-1) \cdots x_j^T(k+N_p | k-1)\right]$$
$$U_j(k, N_p - 1 | k-1) = \left[u_i^T(k | k-1)\; u_i^T(k+1 | k-1) \cdots u_i^T(k+N_p - 1 | k-1)\right]$$
$$W_i(k, N_p - 1 | k-1) = \left[w_i^T(k | k-1)\; w_i^T(k+1 | k-1) \cdots w_i^T(k+N_p - 1 | k-1)\right]$$
$$V_i(k, N_p - 1 | k-1) = \left[v_i^T(k | k-1)\; v_i^T(k+1 | k-1) \cdots v_i^T(k+N_p - 1 | k-1)\right]$$

For example, the state and output interactions prediction vectors for agent 2 can be calculated as follows:

$$W_2(k, N_p - 1 | k-1) = diag\{\bar{A}_{21}(k+1, N_p | k-1)\}\, \hat{X}_1^*(k+1, N_p | k-1) + diag\{\bar{B}_{21}(k+1, N_p | k-1)\} U_1^*(k, N_p - 1 | k-1) \tag{5.18}$$

$$V_2(k, N_p | k-1) = diag\{\bar{C}_{21}(k+1, N_p | k-1)\} \hat{X}_1^*(k+1, N_p | k-1) \tag{5.19}$$

or in matrix form:

$$\begin{bmatrix} \hat{w}_2(k \mid k-1) \\ \hat{w}_2(k+1 \mid k-1) \\ \vdots \\ \hat{w}_2(k+N_p-1 \mid k-1) \end{bmatrix} = \begin{bmatrix} A_{21}(k+1) & 0 & 0 & 0 \\ 0 & A_{21}(k+2) & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & A_{21}(k+N_p) \end{bmatrix} \begin{bmatrix} \hat{x}_1^*(k+1 \mid k-1) \\ \hat{x}_1^*(k+2 \mid k-1) \\ \vdots \\ \hat{x}_1^*(k+N_p \mid k-1) \end{bmatrix}$$

$$+ \begin{bmatrix} B_{21}(k+1) & 0 & 0 & 0 \\ 0 & B_{21}(k+2) & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & B_{21}(k+N_p) \end{bmatrix} \begin{bmatrix} u_1^*(k \mid k-1) \\ u_1^*(k+1 \mid k-1) \\ \vdots \\ u_1^*(k+N_p-1 \mid k-1) \end{bmatrix}$$

$$\begin{bmatrix} v_2(k) \\ v_2(k+1) \\ \vdots \\ v_2(k+N_p-1) \end{bmatrix} = \begin{bmatrix} C_{21}(k+1) & 0 & 0 & 0 \\ 0 & C_{21}(k+2) & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & C_{21}(k+N_p) \end{bmatrix} \begin{bmatrix} \hat{x}_1^*(k+1 \mid k-1) \\ \hat{x}_1^*(k+2 \mid k-1) \\ \vdots \\ \hat{x}_1^*(k+N_p \mid k-1) \end{bmatrix}$$

5.3.1.3 State and output prediction calculation:

Based on the system model in equations (5.10) and (5.11), $\hat{x}_i(k)$ and $U_i(k, N_p-1 \mid k)$, the future values of the plant states over the prediction horizon N_p can be represented in a compact form as follows:

$$\hat{X}_i(k+1, N_p \mid k) = \hat{N}_i \hat{x}_i(k) + \hat{L}_i \bar{B}_{ii} U_i(k, N_p-1 \mid k) + \hat{L}_i \hat{W}_i(k, N_p-1 \mid k-1) \quad (5.20)$$

where:

$$\hat{N}_i = \begin{bmatrix} \phi_k^k \\ \phi_k^{k+1} \\ \vdots \\ \phi_k^{k+N_p-1} \end{bmatrix}, \qquad \hat{L}_i = \begin{bmatrix} I & 0 & \cdots & 0 \\ \phi_{k+1}^{k+1} & I & 0 & 0 \\ \vdots & \ddots & I & \vdots \\ \phi_{k+1}^{k+N_p-1} & \cdots & \phi_{k+N_p-1}^{k+N_p-1} & I \end{bmatrix},$$

$$\bar{B}_{ii} = \begin{bmatrix} B_{ii}(k) & 0 & \cdots & 0 \\ 0 & B_{ii}(k+1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & B_{ii}(k+N_p-1) \end{bmatrix}$$

$$\hat{W}_i(k, N_p-1 \mid k-1) = \begin{bmatrix} \hat{w}_i(k \mid k-1) \\ \hat{w}_i(k+1 \mid k-1) \\ \vdots \\ \hat{w}_i(k+N_p-1 \mid k-1) \end{bmatrix}$$

$$\phi_j^k = A_{ii}(k)\, A_{ii}(k-1) \ldots\ldots A_{ii}(j)$$

An iterative solution is required to calculate the state prediction $\hat{X}_i(k+1, N_p \mid k)$ and the associated matrices, $A_{ii}(k+l/k))$, $B_{ii}(k+l/k))$ and $C_{ii}(k+l/k))$, for l=1,...,N_p . The output prediction $\hat{Y}_i(k+1, N_p \mid k)$ can be calculated using the states prediction as follows:

$$\begin{bmatrix} \hat{y}_i(k+1 \mid k) \\ \hat{y}_i(k+2 \mid k) \\ \vdots \\ \hat{y}_i(k+N_p \mid k) \end{bmatrix} = \begin{bmatrix} C_{ii}(k+1) & 0 & 0 & 0 \\ 0 & C_{ii}(k+2) & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & C_{ii}(k+N_p) \end{bmatrix} \begin{bmatrix} \hat{x}_i(k+1 \mid k) \\ \hat{x}_i(k+2 \mid k) \\ \vdots \\ \hat{x}_i(k+N_p \mid k) \end{bmatrix} + \begin{bmatrix} \hat{v}_i(k+1 \mid k-1) \\ \vdots \\ \hat{v}_i(k+N_p-1 \mid k-1) \\ \hat{v}_i(k+N_p-1 \mid k-1) \end{bmatrix} \tag{5.21}$$

Using equations (5.20) and (5.21), the output prediction can be expressed as follows:

$$\begin{aligned} \hat{Y}_i(k+1, N_p \mid k) = \hat{\Lambda}_i\, \hat{x}_i(k) + \hat{\Phi}_i U_i(k, N_p-1 \mid k) + \hat{\Psi}_i \hat{W}_i(k, N_p-1 \mid k-1) \\ + T_i \hat{V}_i(k, N_p-1 \mid k-1) \end{aligned} \tag{5.22}$$

where:

$$\hat{\Lambda}_i = \bar{C}_{ii}\hat{N}_i\,,\quad \hat{\Phi}_i = \bar{C}_{ii}\hat{L}_i\bar{B}_{ii}\,,\quad \hat{\Psi}_i = \bar{C}_{ii}\hat{L}\ ,$$

$$\hat{C}_{ii} = \begin{bmatrix} C_{ii}(k+1) & 0 & 0 & 0 \\ 0 & C_{ii}(k+2) & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & C_{ii}(k+N_p) \end{bmatrix},$$

$$T_i = \left[\begin{array}{c|ccc} 0_{nyi} & & & \\ 0_{nyi} & & I_{(N_p-1)nyi} & \\ \vdots & & & \\ 0_{nyi} & 0_{nyi} & \cdots & 0_{nyi} \quad I_{nyi} \end{array}\right]$$

5.3.1.4 Cost function minimization:

The DMPC problem is that each agent minimizes the local cost function as follows:

$$J_i = \frac{1}{2}\sum_{j=1}^{N_p} \| \hat{y}_i(k+j\,|\,k) - y_i^d(k+j\,|\,k) \|^2_{Q_i} + \| u_i(k+j\,|\,k) - u_i(k+j-1\,|\,k) \|^2_{S_i} \qquad (5.23)$$

subject to inequality constraints of the form:

$$A_c U_k \le B_c \qquad (5.24)$$

The output of a process can be limited by introducing upper and lower output constraints $y_{\max}$ and $y_{\min}$. Then, the output inequality constraints can be expressed using equation (5.22) as follows:

$$A_c = \begin{bmatrix} -\hat{\Phi} \\ \hat{\Phi} \end{bmatrix} \qquad (5.25)$$

$$B_c = \begin{bmatrix} y_{\min} + \hat{\Lambda}_i\,\hat{x}_i(k) + \hat{\Psi}_i \hat{W}_i(k-1) + T_i\hat{V}(k-1) \\ -y_{\max} - \hat{\Lambda}_i\,\hat{x}_i(k) - \hat{\Psi}_i \hat{W}_i(k-1) - T_i\hat{V}(k-1) \end{bmatrix} \qquad (5.26)$$

where N_p is the maximum output horizon, and Q_i and S_i are the weighting on the tracking error and the control increments respectively. The cost function to be

minimised for any subsystem S_i can be formulated in quadratic programming problem as follows:

$$min\, J_i = \frac{1}{2} U_i^T H_i U_i + G_i^T U \tag{5.27}$$

where:

$$H_i = (\hat{\Phi}_i^T \tilde{Q}_i \hat{\Phi}_i + H_s) \tag{5.28}$$

$$\begin{aligned} G_i = -\hat{\Phi}_i^T \tilde{Q}_i [Y_i^d (k+1, N_p \mid k) - \hat{\Lambda}_i \hat{x}_i (k) - \hat{\Psi}_i \hat{W}_i (k, N_p - 1 \mid k-1) \\ - T_i \hat{V}_i (k, N_p - 1 \mid k-1)] \end{aligned} \tag{5.29}$$

$$\bar{Q}_i = \begin{bmatrix} Q_i & & & \\ & Q_i & & \\ & & \ddots & \\ & & & Q_i \end{bmatrix}, \quad H_s = \begin{bmatrix} 2S_i & -S_i & & & \\ -S_i & 2S_i & -S_i & & \\ & \ddots & \ddots & \ddots & \\ & & -S_i & 2S_i & -S_i \\ & & & -S_i & S_i \end{bmatrix}$$

The implementation algorithm can be summarised as follows:

Step 1: At sampling time k=1 and given the initial control input $u_i(0)$ and initial states $x_i(0)$, each agent estimates the current state vector $\hat{x}_i(k)$ using a nonlinear state-dependent Kalman filter (SDKF). Then, each agent sends their input output measurements to all subsequent SDKF filters through a one directional communication channel.

Step 2: For any subsystem S_i except subsystem 1 which is independent from other subsystems, calculate the interaction prediction using equations (5.16) and (5.17) . It is calculated using all of the previous control law $U_i^*(k, N_p - 1 \mid k)$ and the state prediction $\hat{X}_i(k+1, N_p \mid k)$ of the previous subsystems which were received from network.

Step 3: Each agent calculates iteratively the state prediction $\hat{X}_i(k+1, N_p \mid k)$ using equation (5.20) and announces it to all subsequent agents through a one directional communication network.

Step 4: All agents solve their optimal problem simultaneously to obtain their solution $U_i^*(k, N_p \mid h)$ $(i = 1, ..., n)$ using equation(5.27), where the interactions predictions are considered as disturbance.

Step 6: Each agent checks if its terminal iteration condition is satisfied, which is the difference between the control vectors: $\left|U_i\left(new\right) - U_i\left(old\right)\right| < \sigma$. If all terminal conditions are satisfied, then end iteration and go to step 7; otherwise, take the latest solution to step 3.

Step 7: Each agent takes the first element of the control input as the controller output $u_i(k) = [1 \cdots 0] U_i^*(k)$ $(i = 1, ..., n)$ and applies it to the actuator.

Step 8: Let k=k+1, and go to step 2.

5.3.2 DMPC-Algorithm 2: Interactions from previous subsystems are formulated in the current subsystem state-space model

In this algorithm, instead of separating the interaction terms from each subsystem model and considering them as disturbances, as in algorithm 1, the interaction terms from previous subsystems are formulated as functions of their state variables. Then, the state-dependent model for each subsystem $\S_i$ is represented as a function of its current state, the interaction states of previous subsystems, and control inputs of previous subsystems. In general, there are an infinite number of such re-arrangements. However, the state-dependent matrices of the subsystem model A_{ii}, B_{ii} and C_{ii} should be formulated such that the resulting LTV system is locally observable and controllable. Therefore, for any subsystem $\S_i$, the model can be represented as follows:

$$x_i(k+1) = A_{ii}(x_1^* \cdots x_i, u_1^* \cdots u_{i-1}^*)\ x_i(k) + B_{ii}(x_1^* \cdots x_i, u_1^* \cdots u_{i-1}^*)\ u_i(k) \tag{5.30}$$

$$y_i(k) = C_{ii}(x_1^*, x_2^* \cdots x_i(k), u_1^*, u_2^* \cdots u_{i-1}^*)\ x_i(k) \tag{5.31}$$

where $x_i(k)$ is the subsystem $\S_i$ current state, and $x_1^* \cdots x_{i-1}^*$ and $u_1^* \cdots u_{i-1}^*$ are respectively the states and the control inputs of the previous subsystems, which were estimated and calculated at a previous time step and transmitted through the network. For example, the state-dependent model of subsystem 2 can be represented as follows:

Algorithm 1:

$$\begin{aligned} x_2(k+1) = {} & A_{22}(x_2(k))\, x_2(k) + B_{22}(x_2(k))\, u_2(k) \\ & + A_{21}(x_1(k-1))\, x_1(k-1) + B_{21}(x_1(k-1))\, u_1(k-1) \end{aligned} \tag{5.32}$$

$$y_2(k) = C_{22}(x_2(k))\, x_2(k) + C_{21}(x_1(k-1))\, x_1(k-1) \tag{5.33}$$

Algorithm 2:

$$\begin{aligned} x_2(k+1) = {} & \tilde{A}_{22}(x_1(k-1), x_2(k), u_1(k-1))\, x_2(k) \\ & + \tilde{B}_{22}(x_1(k-1), x_2(k), u_1(k-1))\, u_2(k) \end{aligned} \tag{5.34}$$

$$y_2(k) = \tilde{C}_{22}(x_1(k-1), x_2(k))\, x_2(k) \tag{5.35}$$

The above example shows that the state-dependent model for subsystem 2 in algorithm 2 is of the same dimension and form as the case where there is no interaction. The effect of the subsystem 1 is included inside the state-dependent matrices $\tilde{A}_{22}$, $\tilde{B}_{22}$ and $\tilde{C}_{22}$. Therefore, in algorithm 2, the NMPC control design algorithm for the centralized control structure presented in Chapter 3 can be used to solve the distributed NMPC for any subsystem $\S_i$.

In this algorithm, the nonlinear dynamic model for each subsystem $\S_i$ plus all the interactions from previous subsystems are included in the control system of the i-th unit. This allows all previous subsystem control units to send “Feed-forward” their states to the i-th unit at discrete time instants to provide it with the evolution of the

interaction states. Then, a local nonlinear state-dependent Kalman filter (SDKF) is used to estimate the local subsystem states.

This algorithm has the advantage of less communication burden because only the state estimates and control input are transmitted through the network, whereas in algorithm I, the state prediction and control input trajectories are transmitted.

The implementation algorithm can be summarised as follows:

Step 1: At sampling time k=1 and given the initial control input $u_i(0)$ and initial states $x_i(0)$, each agent estimates the current state vector $\hat{x}_i(k)$ using a nonlinear state-dependent Kalman filter and announces it to all subsequent agents through a one directional communication network.

Step 2: Each agent calculates iteratively the state prediction $\hat{X}_i(k+1, N_p \mid k)$.

Step 3: All agents solve their optimal problem simultaneously to obtain their solution $U_i^*(k, N_p \mid h)$ $(i = 1, ..., n)$.

Step 4: Each agent checks if its terminal iteration condition is satisfied, which is the difference between the control vectors: $\left|U_i\left(new\right) - U_i\left(old\right)\right| < \sigma$. If all terminal conditions are satisfied, then end iteration and go to step 6; otherwise, take the latest solution to step 2.

Step 5: Each agent takes the first element of the control input as the controller output $u_i(k) = [1 \cdots 0] U_i^*(k)$ $(i = 1, ..., n)$ and applies it to the actuator.

Step 6: Let k=k+1, and go to step 2.

5.4 Sequential Distributed Nonlinear State Estimation

Most of the DMPC schemes mentioned in literature depend on the assumption of availability of the measurement of complete states (Christofides et al., 2013). However, in a large-scale control system it is possible that measurements of all states are not available. The conventional centralized Kalman filter observer can be used to estimate states of all distributed controllers. However, the computational effort required to implement this estimator for large-scale systems can be unreasonable for many online applications. Therefore, it is important to study distributed state estimation schemes for a large scale-system in order to integrate them with DMPC control algorithms to achieve the desired level of performance and robustness.

In (Mutambara, 1998) the concept of scalable decentralized estimation is introduced. The basic ideas are found in (Rao and Durrant-Whyte, 1991; Rao et al., 1993). In a distributed Kalman filter, as shown in Figure 5.6, the whole system is decomposed into dimensional subsystems and the estimation algorithm is replaced with a low order Kalman filter implemented at each of these subsystems. Each local filter shares information with other filters through a network and computes a local state estimate. Compared with a traditional centralized estimation scheme, several advantages emerge such as scalability, low communication load, fast implementation and more robustness to sensor failures. Several decentralized and distributed estimation schemes for large-scale systems have been proposed (Vadigepalli and Doyle Iii, 2003; Olfati-Saber, 2007; Menighed et al., 2009; Roshany-Yamchi et al., 2011) to make the estimation problem computationally efficient.

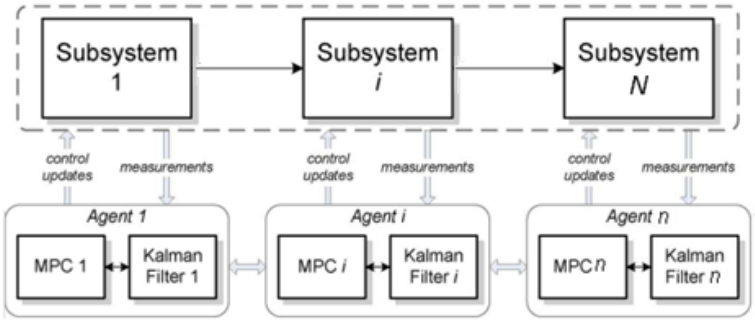


Figure 5.6: Distributed system (Roshany-Yamchi et al., 2011)

Despite the need for nonlinear distributed estimation in numerous practical situations, most studies in the field of distributed estimation have only focussed on linear dynamic systems (Simonetto et al., 2010). A distributed extended Kalman filter (EKF) was proposed in Di Rocco and Pascucci (2007). The performance of this EKF might be degraded when nonlinearity is severe or discontinuous.

This chapter proposes fully distributed state estimation algorithms for large-scale power plant system, where the previous work on the state-dependent Riccati equation (SDRE) for nonlinear estimation is extended to the distributed case. Two sequential nonlinear state estimation algorithms are proposed in this thesis.

5.4.1 Algorithm I: sharing measurements

Let the system under control be composed of the interconnection of *i* local subsystems described by the state-dependent nonlinear discrete-time model as given in equations (5.3) and (5.4). The state-dependent coefficient (SDC) matrices $A_{ii}(\hat{x}_i)$, $B_{ii}(\hat{x}_i)$ and $C_{ii}(\hat{x}_i)$ should be formulated such that the resulting LTV system is observable for all x.

This sequential algorithm requires every subsystem node to send its measurements to all subsequent subsystem nodes in one direction communication and once at each time step. Let $Y_i(k)$ and $U_i(k)$ denote the output and input information available at subsystem node i at time k. For example, node 1 has access to only $Y_1(k) = y_1(k)$, $U_1(k) = u_1(k)$, whereas node 2 has access to $Y_2(k) = \left[y_1(k-1), y_2(k)\right]$, $U_2(k) = \left[u_1(k-1), u_2(k)\right]$. For any node i >1, the available information are $Y_i(k) = \left[y_1(k-1), y_2(k-1), \cdots, y_i(k)\right]$ and $U_i(k) = \left[u_1(k-1), u_2(k-1), \cdots, u_i(k)\right]$. In the sequential structure, the subsystem process 1 is independent of the rest of the system and there is no interaction affecting this subsystem. The states of subsystem 1 are first estimated using a SDRE Kalman filter:

$$\hat{x}_1(k+1) = A_{11}x_1(k) + B_{11}u_1(k) + K_1^f\left[y_1(k) - C_{11}x_1(k)\right] \tag{5.36}$$

Therefore, the SDRE estimation method described in Chapter 4 can be used to estimate the subsystem 1 states.

For subsystem 2, the Kalman filter may be designed by formulating the model as follows:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_{k+1} = \begin{bmatrix} A_{11}(\hat{x}_1^*) & 0 \\ A_{21}(\hat{x}_1^*) & A_{22}(\hat{x}_2(k)) \end{bmatrix}_k \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_k + \begin{bmatrix} B_{11}(\hat{x}_1^*) & 0 \\ B_{21}(\hat{x}_1^*) & B_{22}(\hat{x}_2) \end{bmatrix}_k \begin{bmatrix} u_1^*(k-1) \\ u_2(k) \end{bmatrix} + \begin{bmatrix} \zeta_1 \\ \zeta_2 \end{bmatrix}_k$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}_k = \begin{bmatrix} C_{11}(\hat{x}_1^*) & 0 \\ C_{21}(\hat{x}_1^*) & C_{22}(\hat{x}_2) \end{bmatrix}_k \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_k + \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix}_k \tag{5.37}$$

The Kalman filter designed using this model has the following structure:

$$\begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix}_{k+1} = \begin{bmatrix} A_{11}(\hat{x}_1^*) & 0 \\ A_{21}(\hat{x}_1^*) & A_{22}(\hat{x}_2) \end{bmatrix}_k \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix}_k + \begin{bmatrix} B_{11}(\hat{x}_1^*) & 0 \\ B_{21}(\hat{x}_1^*) & B_{22}(\hat{x}_2) \end{bmatrix}_k \begin{bmatrix} u_1^*(k-1) \\ u_2(k) \end{bmatrix}$$

$$+ K_2^f \left\{ \begin{bmatrix} y_1(k-1) \\ y_2(k) \end{bmatrix} - \begin{bmatrix} C_{11}(\hat{x}_1^*) & 0 \\ C_{21}(\hat{x}_1^*) & C_{22}(\hat{x}_2) \end{bmatrix}_k \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix}_k \right\} \tag{5.38}$$

where $u_2(k)$ is the closed-loop control signal obtained from the MPC controller for subsystem 2, and $u_1^*(k-1)$ is from the optimization of the first system and received from subsystem 1 through the network. Equation (5.38) can be written in simplified vector form as follows:

$$\hat{X}_2(k+1) = \tilde{A}_2(x_1^*, x_2)\,\hat{X}_2(k) + \tilde{B}_2(x_1^*, x_2)\tilde{U}_2(k) + K_2^f[\tilde{Y}_2(k) - \hat{C}_2\hat{X}_2(k)] \tag{5.39}$$

Therefore, the standard SDRE Kalman filter can be used to estimate the subsystem 2 states.

For any subsystem $\S_i$ the model can be represented as follows:

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_i \end{bmatrix}_{k+1} = \begin{bmatrix} A_{11}(\hat{x}_1) & 0 & 0 & 0 & 0 \\ A_{21}(\hat{x}_1^*) & A_{22}(\hat{x}_2) & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & 0 \\ A_{i\times 1}(\hat{x}_1^*) & \cdots & \cdots & \cdots & A_{i\times i}(\hat{x}_i) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{i-1} \\ x_i \end{bmatrix}_k$$
$$+ \begin{bmatrix} B_{11}(\hat{x}_1) & 0 & 0 & 0 & 0 \\ B_{21}(\hat{x}_1^*) & B_{22}(\hat{x}_2) & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & 0 \\ B_{i\times 1}(\hat{x}_1^*) & \cdots & \cdots & \cdots & B_{i\times i}(\hat{x}_i) \end{bmatrix} \begin{bmatrix} u_1^* \\ u_2^* \\ u_3^* \\ \vdots \\ u_i \end{bmatrix}_k + \begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \vdots \\ \zeta_{i-1} \\ \zeta_i \end{bmatrix}_k \tag{5.40}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ \vdots \\ y_i \end{bmatrix}_k = \begin{bmatrix} C_{11}(\hat{x}_1) & 0 & 0 & 0 & 0 \\ C_{21}(\hat{x}_1^*) & C_{22}(\hat{x}_2) & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & 0 \\ C_{i\times 1}(\hat{x}_1^*) & \cdots & \cdots & \cdots & C_{i\times i}(\hat{x}_i) \end{bmatrix} \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \vdots \\ \hat{x}_{i-1} \\ \hat{x}_i \end{bmatrix}_k + \begin{bmatrix} \eta_1 \\ \eta_2 \\ \vdots \\ \vdots \\ \eta_i \end{bmatrix}_k \tag{5.41}$$

The Kalman filter can be designed using the following structure:

$$\begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \vdots \\ \vdots \\ \hat{x}_i \end{bmatrix}_{k+1} = \begin{bmatrix} A_{11}(\hat{x}_1^*) & 0 & 0 & 0 & 0 \\ A_{21}(\hat{x}_1^*) & A_{22}(\hat{x}_2^*) & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & 0 \\ A_{i\times 1}(\hat{x}_1^*) & \cdots & \cdots & \cdots & A_{i\times i}(\hat{x}_i) \end{bmatrix} \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \vdots \\ \hat{x}_{i-1} \\ \hat{x}_i \end{bmatrix}_k$$
$$+ \begin{bmatrix} B_{11}(\hat{x}_1^*) & 0 & 0 & 0 & 0 \\ B_{21}(\hat{x}_1^*) & B_{22}(\hat{x}_2^*) & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & 0 \\ B_{i\times 1}(\hat{x}_1^*) & \cdots & \cdots & \cdots & B_{i\times i}(\hat{x}_i) \end{bmatrix} \begin{bmatrix} u_1^* \\ u_2^* \\ u_3^* \\ \vdots \\ u_i \end{bmatrix}_k \tag{5.42}$$
$$+ K_i^f \left\{ \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ \vdots \\ y_i \end{bmatrix} - \begin{bmatrix} C_{11}(\hat{x}_1^*) & 0 & 0 & 0 & 0 \\ C_{21}(\hat{x}_1^*) & C_{22}(\hat{x}_2^*) & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & 0 \\ C_{i\times 1}(\hat{x}_1^*) & \cdots & \cdots & \cdots & C_{i\times i}(\hat{x}_i) \end{bmatrix} \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \vdots \\ \hat{x}_{i-1} \\ \hat{x}_i \end{bmatrix}_k \right\}$$

This can be written in simplified form as in the following equation, and then the standard SDRE Kalman filter can be used to estimate the states for any subsystem *i*:

$$\begin{aligned}\hat{X}_i(k+1) = \tilde{A}_i(x_1^*,\ldots,x_i)\,\hat{X}_i(k) + \tilde{B}_i(x_1^*,\ldots x_i)\tilde{U}_i(k) \\ + K_i^f[\tilde{Y}_i(k) - \hat{C}_i(x_1^*,\ldots x_i)\hat{X}_i(k)]\end{aligned} \tag{5.43}$$

5.4.2 Algorithm II: sharing states and control inputs

In this algorithm, the state-dependent model for each subsystem $\S_i$ is represented as in equations (5.30) and (5.31), where the current state, the interaction states of previous subsystems, and control inputs of previous subsystems are all included in the model. Therefore, this sequential algorithm requires every subsystem node to send its state estimates and control input to all subsequent subsystem nodes in one direction communication and once at each time step. The advantage of this estimation algorithm compared to algorithm I is that it has less of a computation burden. This is because in algorithm I each subsystem model should include all models of previous subsystems. Another advantage, if this algorithm is used with distributed control algorithm II, is that no additional information is required to communicate through the network.

Let $X_i(k)$ and $U_i(k)$ denote the states and input information available at subsystem node i at time k. For example, for any node i >1, the available information is $X_i(k) = [x_1(k-1),\cdots x_{i-1}(k-1), x_i(k)]$, $U_i(k) = [u_1(k-1),\cdots u_{i-1}(k-1), u_i(k)]$. Figure **5.7** shows the sequential interconnections and exchange of estimation of information.

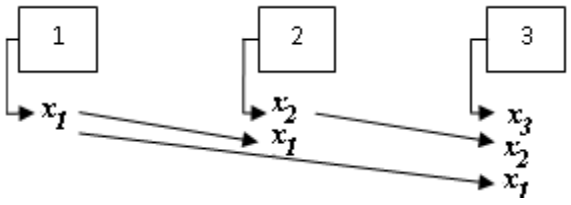


Figure 5.7: Sequential exchange of estimation information

Using the model equation given in equations (5.30) and (5.31), the SDRE Kalman filter described in Chapter 4 can be applied for any subsystem $\S_i$ using the following structure:

$$\hat{x}_i(k+1) = A_{ii}(x_1^* \cdots x_i, u_1^* \cdots u_{i-1}^*)\ x_i(k) + B_{ii}(x_1^* \cdots x_i, u_1^* \cdots u_{i-1}^*)\ u_i(k) + K_i^f \left[y_i(k) - C_{ii}(x_1^* \cdots x_i, u_1^* \cdots u_{i-1}^*)\, x_i(k) \right] \tag{5.44}$$

5.5 Combined Cycle Power Plant Distributed Models

The power plant nonlinear model described in Chapter 4 section 4.6.2.1 was partitioned or decomposed according to the physical plant structure into three subsystems: Drum-boiler with supplementary firing furnace, Boiler superheater and steam turbine with electric generator. Figure 5.8 illustrates a schematic structure for the power plant system including interactions.

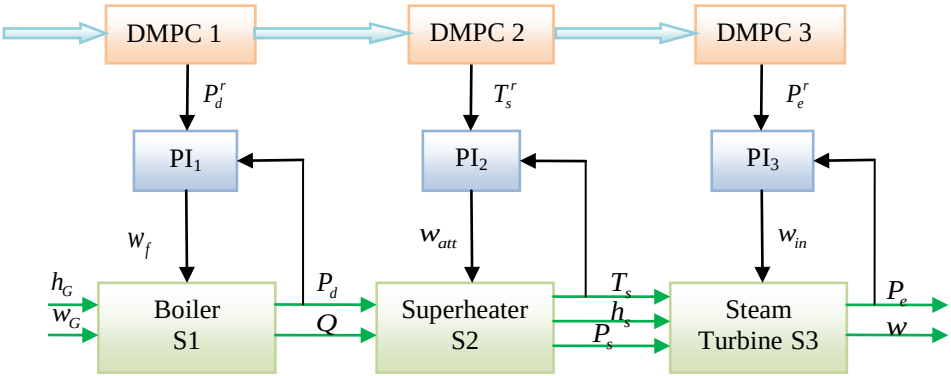


Figure 5.8: Boiler turbine system structure

5.5.1 Subsystems models

The dynamic models of boiler, superheater and steam turbine given in Chapter 4 equations (4.48) , (4.52) and (4.55) are used in the simulation of DMPC-algorithm 2. However, for DMPC-algorithm 1, in order to separate the interactions to consider them as disturbances, these equations are modified as follows:

5.5.1.1 Subsystem 1: Boiler model

The boiler subsystem 1 model is independent from the rest of the system. Therefore, the SDC models given in equation (4.47) for continuous model and in equation (4.48) for discrete model are used in SDDRE filter and DMPC respectively.

5.5.1.2 Subsystem 2: Superheater model

To simulate DMPC-algorithm 1, the superheater nonlinear dynamic model given in Chapter 4, equation (4.52), is decomposed into two terms: the SDC matrix $A_{22}(\hat{x}_2)$, and the SDC interaction matrix $A_{21}(\hat{x}_1)$. Therefore, the superheater SDC discrete model including the PI controller can be expressed as follows:

$$\begin{aligned} x_2(k+1) &= A_{22}(\hat{x}_2)\,x_2(k) + B_{22}(\hat{x}_2)\,u_2(k) + A_{21}(\hat{x}_1)\,x_1(k) \\ y_2(k) &= C_{22}(\hat{x}_2)\,x_2(k) \end{aligned} \tag{5.45}$$

where:

$$A_{22}(\hat{x}_2) = \begin{bmatrix} 1 & -Tc_1 & 0 & -Tc_2 \\ 0 & TG_{11}+1 & TG_{12} & 0 \\ 0 & 0 & TG_{22}+1 & TG_{23} \\ Tb_3k_{i2} & T(G_{31}-b_3c_1k_{p2}) & TG_{32} & T(G_{33}-b_3c_2k_{p2})+1 \end{bmatrix}$$

$$B_{22}(\hat{x}_2) = \begin{bmatrix} T \\ 0 \\ 0 \\ Tb_3k_{p2} \end{bmatrix}, \qquad C_{22}(\hat{x}_2) = \begin{bmatrix} 0 & c_1 & 0 & c_2 \end{bmatrix}$$

$$A_{21}(\hat{x}_1) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & Tm_2 & 0 & Tm_3 & Tm_4 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

where m_1 , m_2 and m_3 are the interaction terms from subsystem 1, related to the energy supplied from the boiler and boiler drum pressure, and defined using the following equations:

The steam mass flow from the drum to the superheater w_v can be formulated as a function of pressure drum pressure states $x_1(5)$:

$$w_v = x_1(5)\sqrt{\frac{\rho_v - \frac{p_s \rho_v}{x_1(5)}}{x_1(5) f_s}} \tag{5.46}$$

The energy Q_{gs} is formulated as a function of subsystem 1' states as follows:

$$Q_{gs} = 0.7(Fd + Td)x_1(2) \tag{5.47}$$

$$Fd = 4(1-\theta)kV_f \sigma T_g^3 \left(\frac{1}{x_1(3)c_{pg}} - \frac{h_{ref}}{c_{pg}x_1(2)} + \frac{Tref}{x_1(2)} \right) \tag{5.48}$$

$$Td = k_{gs} w_{Eg}^{0.6} \left(\frac{1}{x_1(3)c_{pg}} - \frac{h_{ref}}{c_{pg}x_1(2)} + \frac{T_{ref}}{x_1(2)} - \frac{T_{st}}{x_1(2)} \right) \tag{5.49}$$

$$m_1 = \frac{0.7(Fd + Td)}{M_s C_{st}} \tag{5.50}$$

$$m_2 = \frac{k_s w_v^{0.8} T_{ref}}{M_s C_{st} x_1(4)} \tag{5.51}$$

$$m_3 = \frac{-1\times10^6 k_s m_0 w_v^{-0.2} h_{ref}}{M_s C_{st} cps} \tag{5.52}$$

$$m_0 = \frac{w_v}{x_1(5)} \tag{5.53}$$

All coefficients in the previous equations are given in Chapter 2 and Chapter 4.

5.5.1.3 Subsystem 3: Steam turbine model

The steam turbine and generator nonlinear dynamic model given in Chapter 3, equation is decomposed into two terms: the SDC matrix $A_{33}(\hat{x}_3)$, and the SDC interaction matrix $A_{32}(\hat{x}_2)$. Therefore, the SDC discrete model including the PI controller is expressed as follows:

$$\begin{aligned} x_3(k+1) &= A_{33}(\hat{x}_3)x_3 + B_{33}(\hat{x}_3)u_3(k) + A_{32}(\hat{x}_2)x_2 \\ y_{33}(k) &= C_{33}(\hat{x}_3)x_3(k) \end{aligned} \tag{5.54}$$

where:

$$A_{33}(x_3) = \begin{bmatrix} 1 & 0 & -TC_1 & 0 & 0 \\ TB_1k_{i3} & Ts_{11}+1 & T(s_{12}-B_1C_1k_{p3}) & Ts_{13} & 0 \\ TB_2k_{i3} & Ts_{21} & T(s_{22}-B_2C_1k_{p3}) & Ts_{23} & 0 \\ TB_3k_{i3} & Ts_{31} & -TB_3C_1k_{p3} & 1 & 0 \\ 0 & Ts_{41} & Ts_{42} & 0 & Ts_{44}+1 \end{bmatrix},$$

$$B_{33}(\hat{x}_3) = \begin{bmatrix} T \\ TB_1k_{p3} \\ TB_2k_{p3} \\ TB_3k_{p3} \\ 0 \end{bmatrix}, \quad C_{33}(\hat{x}_3) = \begin{bmatrix} 0 & 0 & C_1 & 0 & 0 \end{bmatrix}, \quad A_{32}(\hat{x}_2) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & A_{sk} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

where:

$$A_{sk} = \frac{7.44T}{Vx_2(2)}$$

All coefficients of matrices $A_{33}(\hat{x}_3)$, $B_{33}(\hat{x}_3)$ and $C_{33}(\hat{x}_3)$ are the same as in equation(4.55), except coefficient s_{31} which includes the interactions from subsystem 2 and is rewritten as follows:

$$s_{31} = -\frac{x_3(3)x_3(4)}{Vx_3^2(2)}$$

As discussed in Chapter 3 the choice of SDC matrices is not unique. In order to ensure a solution to control and estimation problems the selection of pairs $(A_{ii}(x_i), B_{ii}(x_i))$ and $(A_{ii}(x_i), C_{ii}(x_i))$ should be controllable and observable in the linear sense for all *x*.

The gas turbine model used in this study is a linear model. Therefore, the gas turbine can be controlled separately using a supervisory linear MPC, as shown in Figure 5.9. The gas turbine control scheme is discussed in Chapter 1. Outputs of gas turbine h_G and w_G are considered as input variables to the boiler. It is assumed here that the values of these outputs are transmitted from the linear MPC controller to the boiler DMPC1 controller through the network.

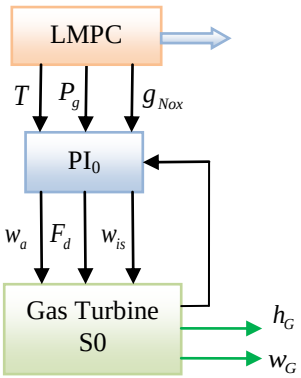


Figure 5.9: Gas turbine supervisory linear MPC

5.6 Simulation Results

5.6.1 Distributed nonlinear state estimation

The nonlinear distributed estimator algorithms described in section 5.4 are implemented in MATLAB/SIMULINK S-functions and applied to the CCPP distributed model described in section 5.5. These distributed algorithms are compared with the nonlinear centralized estimator described in Chapter 3 using the same tuning parameters and noise to make the comparative study as credible as possible. In this simulation, the process noise is assumed to be $\zeta \sim N(0, 0.001)$ and the sensor noise is $\eta \sim N(0, 0.01)$. The covariance matrices P_0, Q_w and R_w which

affect the performance and convergence are assumed to be diagonal. Q_w and R_w matrices depend on the plant model and sensor accuracy respectively. The initial state covariance matrix P_0 represents variances or mean squared errors of the states' initial conditions. Initial conditions of states used for the simulation are given in Chapter 4. In this simulation, it is assumed that the distributed filters communicate only once within a sampling interval and the communication channel introduces a delay of a single sampling time interval. This delay is simulated using the Memory SIMULINK block as shown in Figure C1 and Figure C2 in the appendices.

The evolution of the real states and their estimates obtained by different estimators for the selected states are presented in Figure 5.10. Since the true states are difficult to observe because the filters provide almost the exact values, absolute estimation error of the estimated states are computed and shown in Figure 5.11. For comparison and performance analyses, the Root Mean Squared Errors (RMSEs) of the estimated states using the different algorithms are computed and listed in Table 5.1. In this study, convergence of the nonlinear SDRE filters is verified through simulation, as shown in Figure 5.10. The simulation results demonstrate clearly that the performance of the proposed distributed estimation algorithms 1 and 2 are very close to the centralized estimation algorithm.

The computation load of the proposed Kalman filter algorithms is investigated using the MATLAB profile function. All computations are performed in MATLAB R2010a on a 2.93 GHz intel core Duo computer with 2 Gbyte RAM running Windows Xp SP3. The computation time required for each method for a 90 sec simulation time is shown in Figure 5.12. This figure clearly shows that distributed algorithm 2 is significantly faster than algorithm 1 and the centralized algorithm. This is because in algorithm 1 each subsystem model should include all models of previous subsystems. Consequently, subsystem 3 of algorithm 1 has the same computation burden as the centralized algorithm. However, distributed algorithm 1 can be used in a large-scale distributed system, which in turn has the advantage of more flexibility in case of any subsystem failure compared to the centralized algorithm.

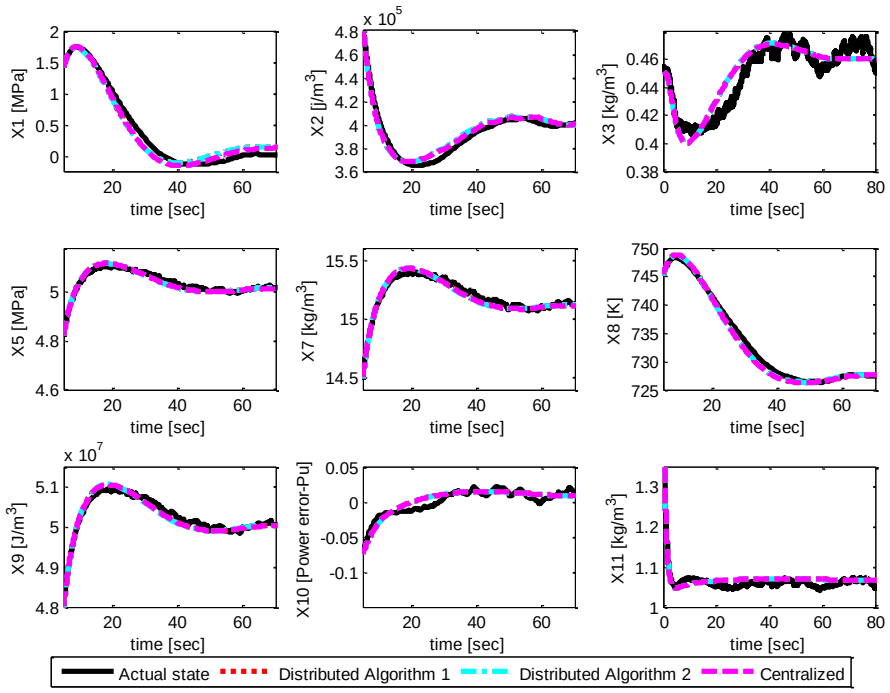


Figure 5.10: Nonlinear states and their estimation

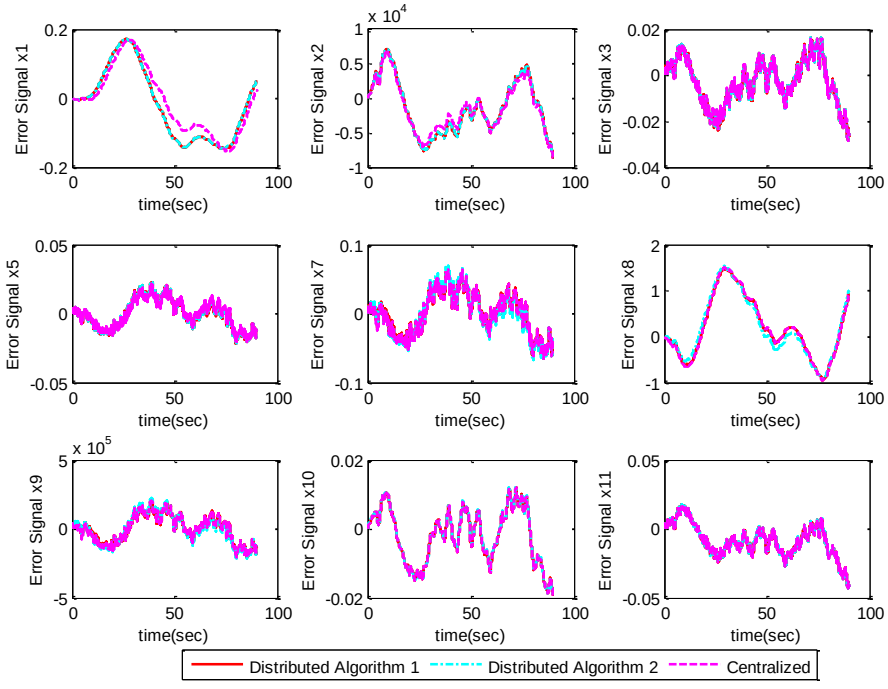


Figure 5.11: Absolute estimation error of nonlinear states

Table 5.1: RMSE of the different estimation methods

	Distributed algorithm 1	*Distributed algorithm 2*	*Centralized*
X1 (Mpa)	0.0278530	0.0278022	0.0252811
X2 (J/m^3)	1106.56571	1105.90222	1009.6161
X3 (kg/m^3)	0.0027167	0.0027112	0.0026667
X4 (m^3)	0.0039894	0.0039896	0.0041820
X5 (Mpa)	0.0027659	0.0027576	0.0026114
X6 (^oK)	0.2073501	0.2057503	0.1822900
X7 (kg/m^3)	0.0079011	0.0085511	0.0080042
X8 (^oK)	0.1828023	0.1827500	0.1852700
X9 (J/m^3)	25964.776	26882.123	26320.276
X10 (Pu.Watt)	0.0043938	0.0043971	0.0043941
X11(kg/m^3)	0.0028633	0.0028591	0.0028640
X12 (Kg/s)	0.0075909	0.0075973	0.0075910
X13 (J/m^3)	8423.9063	8410.1064	8425.6050
X14 (pu.Hz)	0.0043938	0.0043971	0.0043941

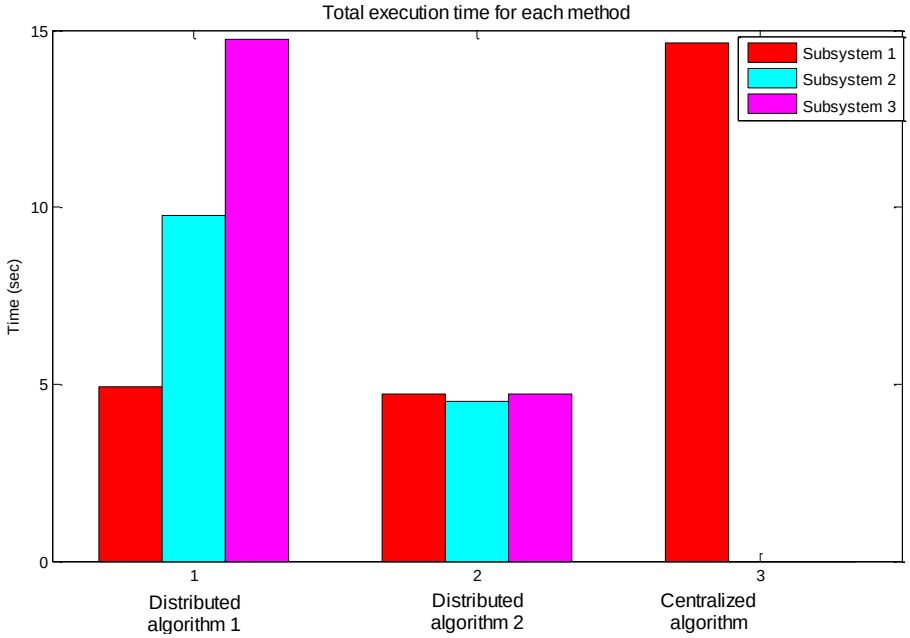


Figure 5.12: Comparison of execution time for all filters

5.6.2 Sequential distributed nonlinear model predictive control

The distributed control strategy described in Figure 5.5 is implemented in S-function MATLAB/SIMULINK environment, as shown in Figure C1 and Figure C2 in Appendix C. Numerical simulations are carried out to evaluate the performance of the proposed two nonlinear DMPC algorithms when compared with the centralized nonlinear MPC presented in Chapter 4. In this control strategy, DMPC's are used to provide the optimal set-point for PIDs to improve the tracking and disturbance rejection while satisfying output constraints. PIDs are mainly used to regulate and stabilize the system. The state-dependent distributed Kalman filter algorithm 1 and algorithm 2 are used to estimate the system states for DMPC algorithm 1 and algorithm 2 respectively.

All simulation studies apply the same nonlinear DMPC tuning parameters which are determined by trial and error to achieve suitable performance. The best values for the PID gains are found using the multivariable PID tuning methods, as described in Chapter 2. The controller has been configured by choosing the weights on tracking error Q= [45, 0.01, 0.5] and weights on control increments S= [43, 10, 50]. The prediction horizon for the optimization problem is N_p= 10, with a time step of T=0.02 sec. The optimization problems of each MPC scheme are solved using the MATLAB Quadprog function provided by the optimization toolbox. Regarding the termination criteria for solving the NMPC problem and the maximum number of iterations, the values used by all simulations are 0.005 and 45, respectively.

In this simulation, it is assumed that the control agents are synchronous and the communication channel introduces a delay of a single sampling time interval. Simulation results comparing the tracking performance of the two proposed nonlinear DMPC algorithms with the centralized nonlinear MPC controllers is shown in Figure 5.13 and Figure 5.14. The results show that the performance of the proposed DMPC algorithms is very close to the centralized NMPC.

To maintain a high level of system safety, output constraints are introduced in this simulation. The introduced minimum and maximum output constraints are Y_{min}= [3.0MPa,715.8^{o}K,0.78Pu-watt] and Y_{max}=[6.0MPa,718^{o}K,0.82Pu-w] respectively.

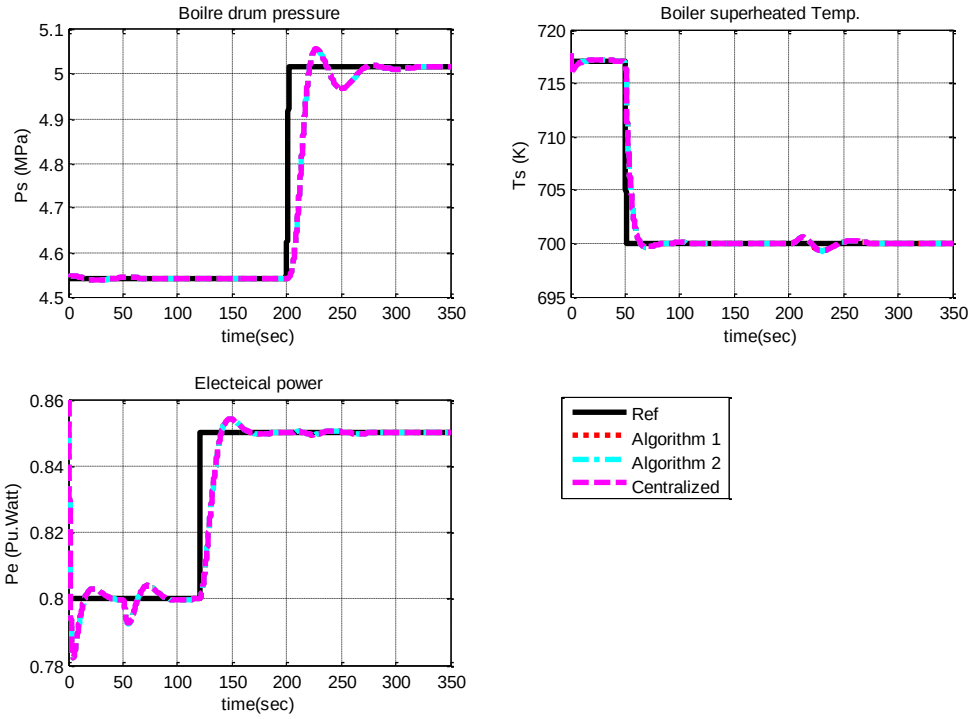


Figure 5.13: Output responses to set point changes

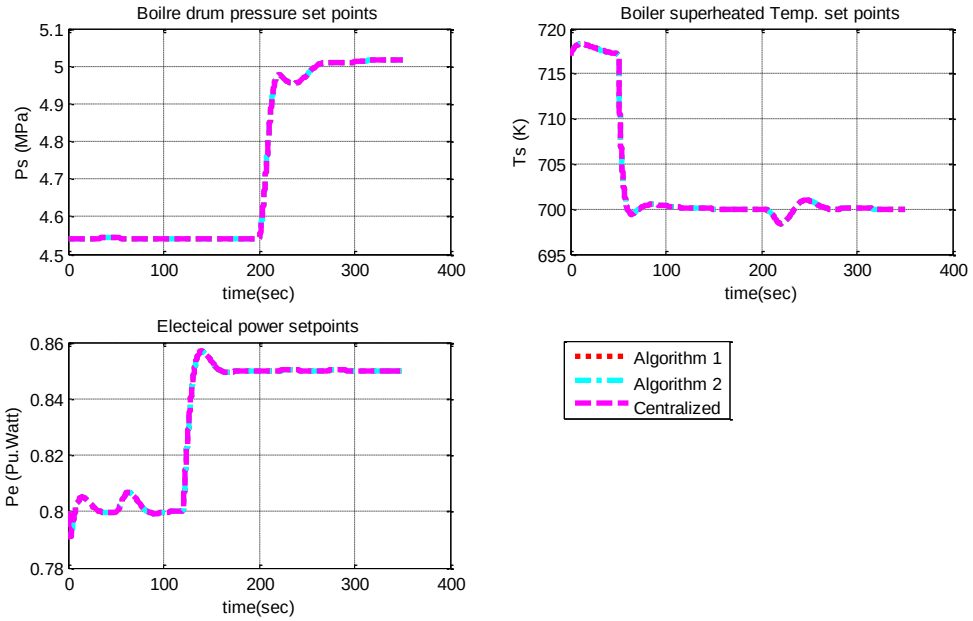


Figure 5.14: Control variables response to set point changes

Figure 5.15 shows the steam turbine output power response to set point changes in boiler superheated temperature. The superheated temperature response to boiler pressure set point changes is shown in Figure 5.16. These figures compare the performance of constrained centralized NMPC and DMPC algorithms with unconstrained centralized NMPC. From these figures it can be clearly observed that the imposed output constraints are satisfied using the proposed constrained DMPC algorithms. These figures also show how the controlled set points and the manipulated variables are changed in order to fulfil the imposed constraints. Figure 5.16, also shows that there is 50s reduction in settling time of the drum pressure compared to the result using PI controller alone as listed in Table 2.7.

Figure 5.17 and Figure 5.18 depict the closed-loop performance of the centralized NMPC and DMPC algorithms when process and output noises are added to the system. The process noise and sensor noise are $\zeta \sim N(0, 0.001)$ and $\eta \sim N(0, 0.01)$ respectively. The simulation results show the good robustness performance to noises, and also show that the DMPC algorithms perform very similar to a centralized NMPC.

In order to check the robustness of the proposed DMPC controllers with respect to plant model variation, a simulation test is performed where the superheater heat transfer coefficient is changed from 4.37×10^4 to 6.37×10^4 $(J / kg^o K)$ at 180sec and the specific heat of the steam turbine steam is changed from 2005 to 2505 $(J / kg^o K)$ at 150sec. The result of this simulation is shown in Figure 5.19, which demonstrates the DMPC controllers' robustness to a change in plant parameters.

Figure 5.20 shows the boiler pressure and superheated temperature response to output step disturbance of 0.1 MPa on boiler pressure at 150sec. As shown in this figure, the disturbance can be eliminated efficiently through the proposed DMPC algorithms.

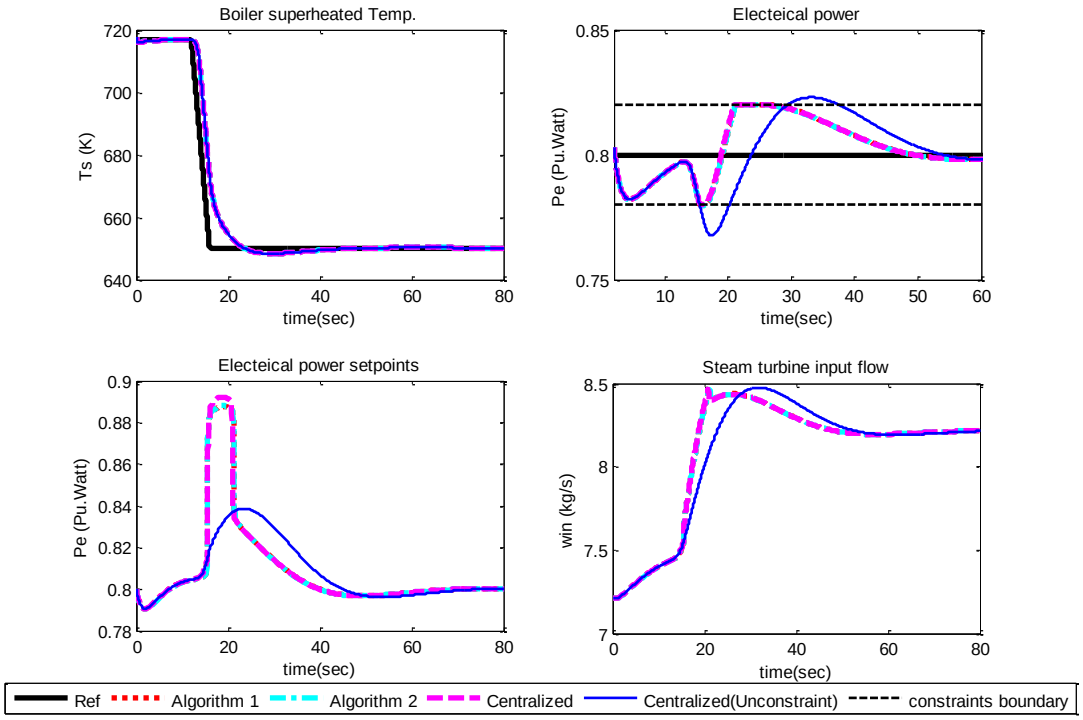


Figure 5.15: Output constraints on steam turbine electrical power

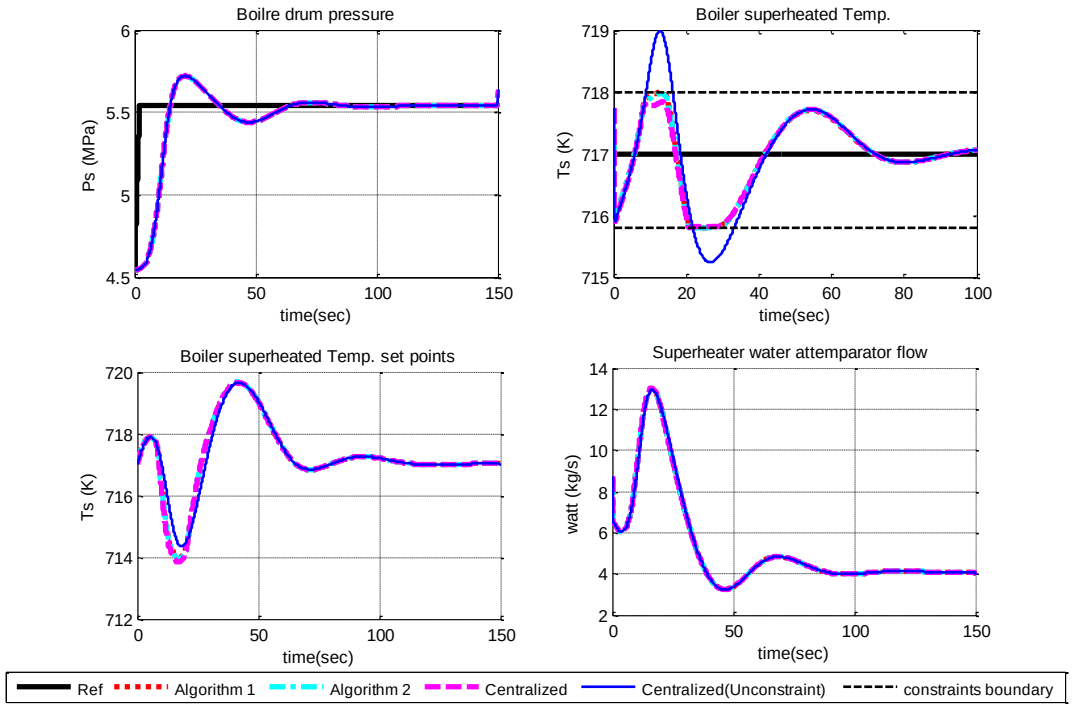


Figure 5.16: Output constraints on boiler superheated temperature

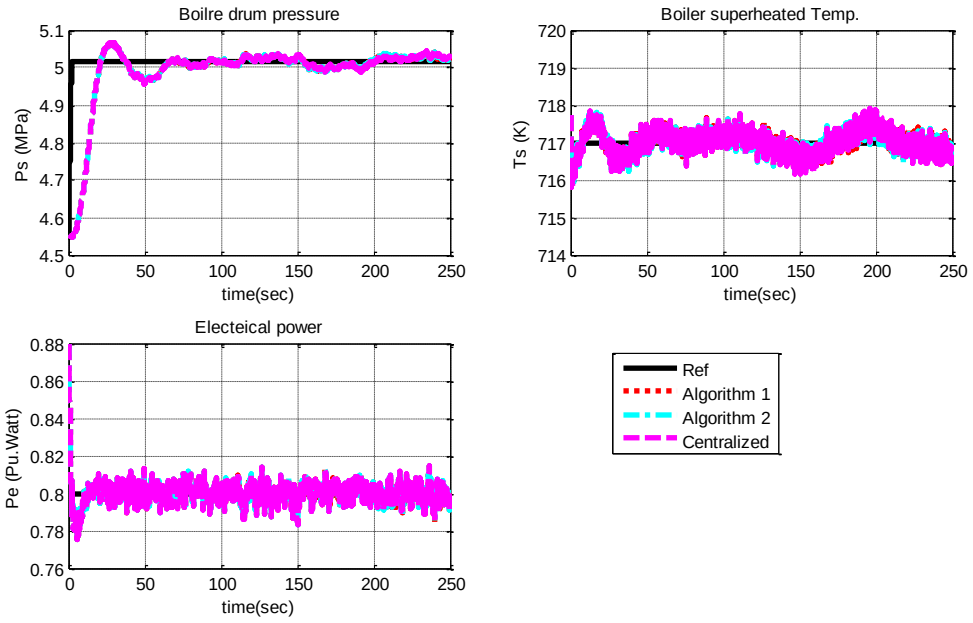


Figure 5.17: Output response to boiler set point change with state and output noises

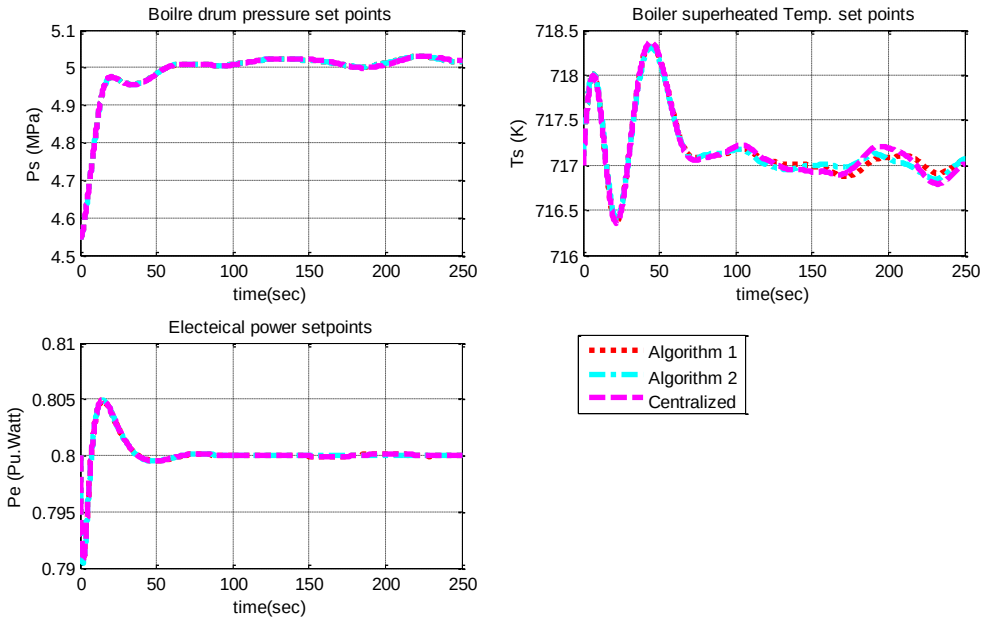


Figure 5.18: Control variables response to boiler set point change with noises

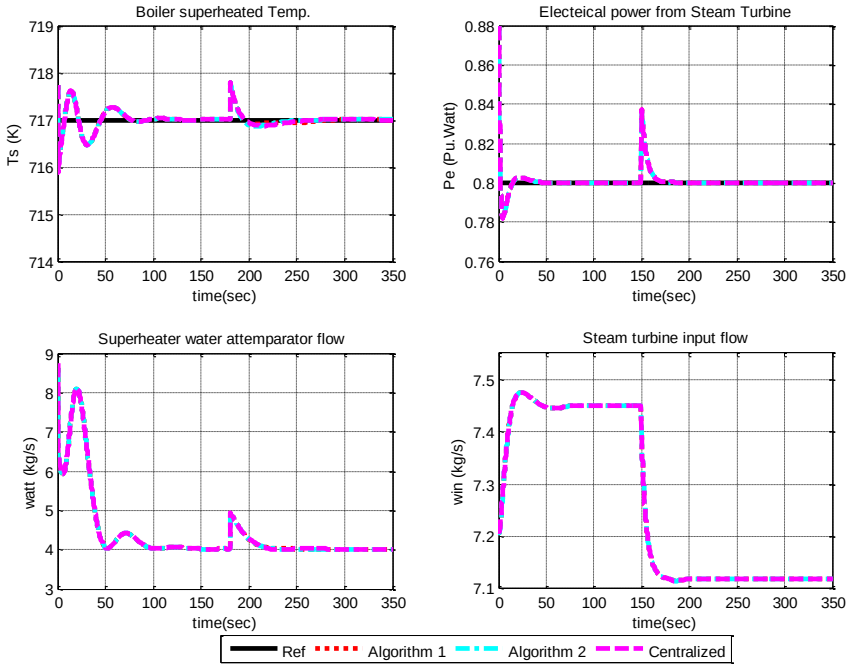


Figure 5.19: Outputs and manipulated variables response to model parameters changes at 180sec and 150sec respectively

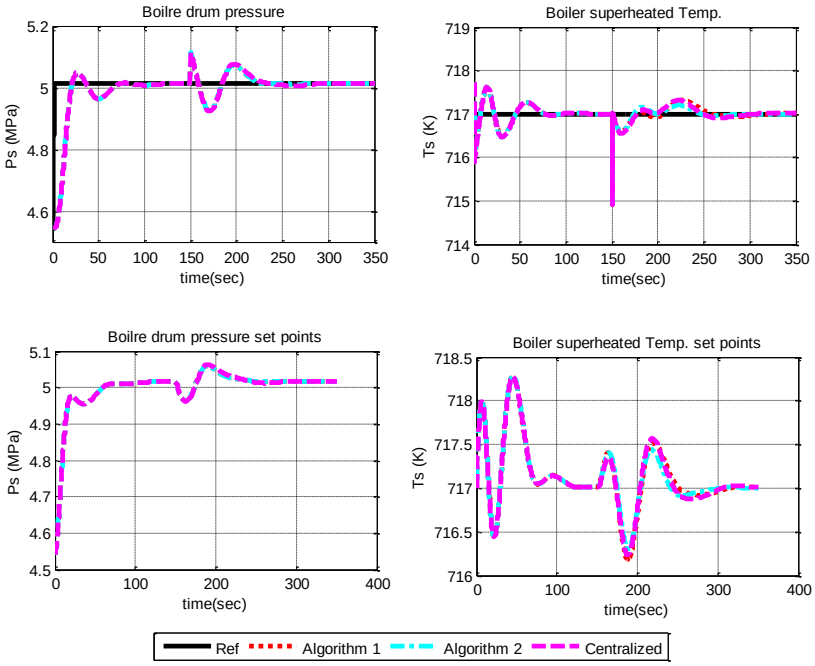


Figure 5.20: Boiler pressure and superheated temperature response to output disturbance on boiler pressure at 150sec

The computation loads of the centralized NMPC and the proposed DMPC algorithms are investigated using the MATLAB profile function. The total computation time required to calculate the optimal control input for each method for 100 sec simulation time is shown in

Figure 5.21. It can be seen that the time consumed by the proposed DMPC algorithms is rather less than that of the centralized NMPC, which clearly demonstrates the advantage of the distributed NMPC over the centralized NMPC.

The performance of the proposed DMPC strategies are compared with the centralized NMPC based on the root mean squared error (RMSEs) of the outputs (deviations from centralized NMPC algorithm results). The results listed in Table 5.2 show that the DMPC algorithms and the centralized performances match very closely.

In this study, the closed-loop stability of the nonlinear DMPC is guaranteed by a suitable tuning of the design parameters, such as prediction horizon and weighting matrices. However, for some processes, adding a terminal weight on the cost function may be required to ensure closed-loop stability.

Table 5.2: RMSE of the proposed DMPC methods

	Distributed algorithm 1			*Distributed algorithm 2*		
	P_s *(MPa)*	T_s *(^oK)*	P_e *(Pu-watt)*	P_s *(MPa)*	T_s *(^oK)*	P_e *(Pu-watt)*
Without noise	8.0147e-8	0.008692	1.1765e-5	8.0147e-8	0.0081552	3.3096e-5
With noise	0.0023376	0.049154	1.642e-5	0.0024422	0.055415	6.5433e-5
With output disturbance	0.0026972	0.043064	1.527e-5	0.0026972	0.040406	4.9073e-5

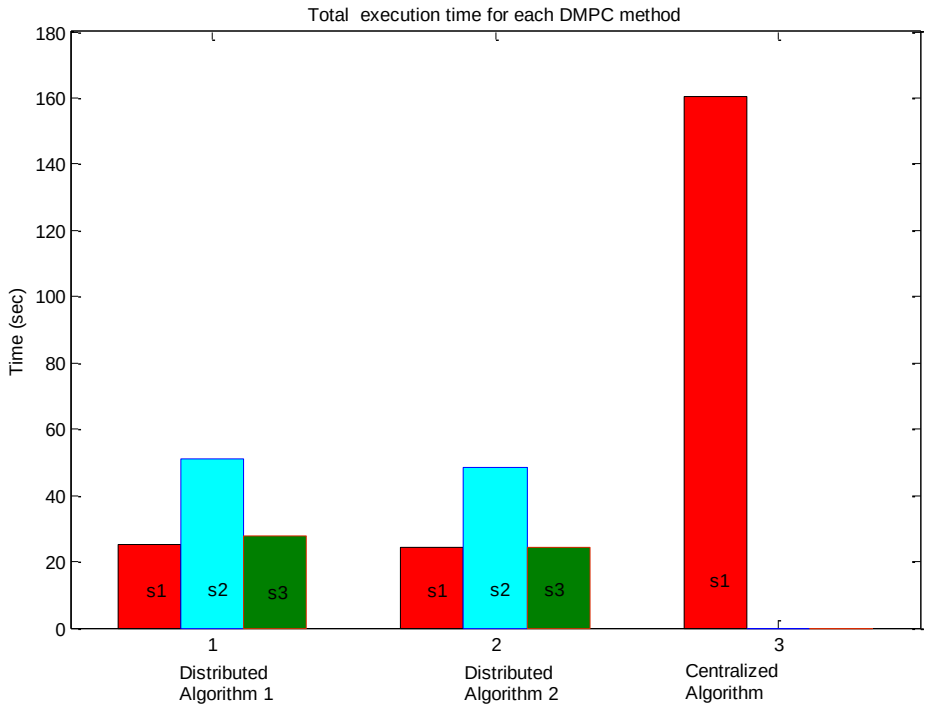


Figure 5.21: Comparison of the total CPU execution time using all control algorithms

5.7 Conclusion

This chapter has presented two novel sequential supervisory nonlinear DMPC algorithms for large-scale power plant processes that can handle constraints. These algorithms are based on the NMPC method discussed in Chapter 4, which uses a state-dependent nonlinear model in order to solve the complexity of the nonlinear programming (NLP) problem. All subsystems exchange information via one direction communication to achieve the objective of the whole system. Simulation results show that the performance of the proposed DMPC algorithms is similar to the centralized NMPC. However, the DMPC algorithms have less computation time compared to the centralized strategy. Moreover, in distributed algorithms, the tuning parameters can be tuned separately, which provides more flexibility in the control design. DMPC algorithm 2 has the advantage of less communication burden compared to DMPC algorithm 1. This chapter has also presented sequential distributed nonlinear estimation algorithms which are based on the state-dependent Riccati equation (SDRE) estimation method.

6. Conclusions

6.1 Thesis Summary

The solution of NMPC optimization problem involves solving nonlinear differential equations and a nonlinear dynamic optimization problem online which is generally non-convex and requires a high computation demand. This on-line computation effort is one of the main obstacles to NMPC applications. The NMPC approach based on state-dependent model has been demonstrated for many applications. In this strategy, the non-linearity is handled by converting the nonlinear system equations into LTV state-dependent state-space representation that requires the solution of a simpler optimization problem. Typically, MPC is studied within a centralized control framework in which all the control inputs are calculated in a single MPC problem. Centralized control strategy for large scale system is often considered impractical and unrealistic due to high on-line computation demand and low reliability. Distributed MPC (DMPC) is a feasible alternative to overcome the increasing computational complexity of centralized MPC.

Motivated by the lack of methodologies on nonlinear DMPC of nonlinear large-scale systems using state-dependent models, this thesis, focused on the development of distributed control and estimation techniques for large-scale systems by using the state-dependent NMPC technique in order to reduce the complexity of the on-line optimization problem.

In this chapter the main results and conclusions are presented. Then some ideas for future work are presented.

6.2 Main Conclusions

Firstly, in chapter 2, nonlinear model of combined cycle power plant (CCPP) based on first principles was developed and implemented in the MATLAB\SIMULINK S-function environment. A new boiler model simulator that can capture the shrink and

swell phenomenon was developed. Decentralized PID controllers were designed to control the CCPP subsystems at the regulatory level. PID controllers were tuned using IMC and Relay feedback tuning methods. Good model accuracy and control performance have been achieved, as confirmed by simulation.

Secondly, using the CCPP model developed in chapter 2, a supervisory linear MPC controller was presented in chapter 3 to improve the control performance of CCPP by providing the optimal set-points for the PID regulatory level. In this controller, output constraints are included in solving the optimization problem to provide safety limitations and satisfy environmental regulations. Simulation results showed that the supervisory MPC has better performance than classical PID control schemes and allows handling of constraints.

Thirdly, a supervisory NMPC algorithm based on LTV state-dependent approach was developed in chapter 4 to control the industrial power plant system. This control algorithm uses a state-dependent nonlinear model in order to reduce the complexity of the on-line optimization problem. The non-linear power plant model was represented in controllable and observable SDC state-space equations in order to be used in NMPC control design and Kalman filter estimator. The proposed NMPC algorithm is an extension of the linear MPC algorithm presented in chapter 3 and has the advantage of including constraints. A nonlinear Kalman filter state estimator based on state-dependent differential Riccati equation was used to estimate the system states.

Fourthly, chapter 5 presented two novel sequential supervisory nonlinear DMPC algorithms for large-scale processes that can handle constraints. These algorithms are based on the state-dependent NMPC method discussed in Chapter 4, which uses a state-dependent nonlinear model in order to solve the complexity of the nonlinear programming (NLP) problem. The difference between these algorithms is related to how the interactions between subsystems in the DMPC solution are considered. In this sequential distributed framework, local MPCs solve a convex optimization problem and exchange information via one directional communication channel at each sampling time to achieve the global control objectives of the system.

Comparison between the centralized NMPC, as discussed in chapter 4 and the new DMPC schemes was performed using the large-scale power plant model. Numerical simulation results showed that the performance of the proposed DMPC algorithms was close to the centralized NMPC but computationally much more efficient. Simulation results showed also that DMPC algorithm 2 has the advantage of less communication burden compared to DMPC algorithm 1, as well as the ability to select a different prediction horizon for each agent.

Finally, chapter 5 has also presented two sequential distributed nonlinear Kalman filter estimation algorithms which are based on the state-dependent differential Riccati equation (SDDRE) estimation method. These estimation algorithms were used to estimate the system states of the DMPC algorithms. The simulation results showed that the performance of the proposed distributed estimation algorithms 1 and 2 are very close to the centralized filter. Simulation results showed also that estimation algorithm 2 needs less computation effort than algorithm 1 and the centralized filter.

6.3 Future work

The following directions for the future research will be considered:

1- Writing the nonlinear system as SDC coefficient has an infinite choice of matrices A(x), B(x) and C(x). Parameterization of the nonlinear dynamics lead to different control laws, and hence, different performances. The problem of selecting of the optimal representation for SDRE technique is complex and depends strongly on the particular system being investigated. Therefore, future research is needed to propose an optimal parameterization.

2- Although state-dependent control methods for non-linear systems have demonstrated their effectiveness in different applications, the guarantee of closed loop stability is still a difficult issue. Therefore, stability analysis of presented state-dependent DMPC algorithms needs to be investigated.

3- Simulations demonstrate that DMPC algorithms have some degree of robustness to disturbances and to modelling errors due to plant parameter variations. Nevertheless, it is important to extend the proposed algorithms in order to deal explicitly with model errors.

4- In order to improve the CCPP model dynamics, the complete CCPP model presented in Chapter 2 should be used in the state-dependent DMPC analysis instead of using the reduced CCPP model.

5- The DMPC control design presented in this thesis can be used in CCPP start-up control mode to achieve better start time and reduced equipment stress.

6- Experimental tests needs to be performed to evaluate the performance of the proposed algorithms.

REFRENCES

Afonso, R. J. M. and R. K. H. Galvão (2012). "Infeasibility handling in constrained MPC." Frontiers of Model Predictive Control, Intech, Rijeka, Croatia:47-64.

Al-Gherwi, W., H. Budman and A. Elkamel (2011). "A robust distributed model predictive control algorithm." Journal of Process Control **21**(8):1127-1137.

Alessio, A., D. Barcelli and A. Bemporad (2011). "Decentralized model predictive control of dynamically coupled linear systems." Journal of Process Control **21**(5): 705-714.

Åström, K. J. and R. D. Bell (1988). "Simple drum-boiler models." IFAC international symposium on power systems modelling and control applications. Brussels, Belgium.

Åström, K. J. and R. D. Bell (2000). "Drum-boiler dynamics." Automatica **36**(3): 363-378.

Åström, K. J. and T. Hägglund (1984). "Automatic tuning of simple regulators with specifications on phase and amplitude margins." Automatica **20**(5): 645-651.

Åström, K. J. and T. Hägglund (1988). Automatic tuning of PID controllers, Instrument Society of America.

Banks, H., B. Lewis and H. Tran (2007). "Nonlinear feedback controllers and compensators: a state-dependent Riccati equation approach." Computational Optimization and Applications **37**(2): 177-218.

Beikzadeh, H. and H. D. Taghirad (2012). "Exponential nonlinear observer based on the differential state-dependent Riccati equation." International Journal of Automation and Computing **9**(4): 358-368.

Beikzadeh, H. and H. D. Taghirad (2012). "Robust SDRE filter design for nonlinear uncertain systems with an H∞ performance criterion." ISA transactions **51**(1): 146-152.

Bithmead, R. P., V. Wertz and M. Gerers (1991). "Adaptive optimal control: the thinking man's GPC." Prentice Hall Professional Technical Reference.

Blood, D. and G. Britain (2003). "Heat Recovery Steam Generators for Power Generation and Other Industrial Applications." Report No. COAL R232, Department of Trade and Industry,UK.

Bristol, E. H. (1966). "On a new measure of interactions for Multivariable process control." IEEE Trans. Aut. Control(AC-II): 133-134.

Busi, T. and R. Cori (1977). "Parameter identification of a drum boiler power plant." Power Plant Dynamics, Control and Testing Symposium. Knoxville.

Camacho, E. F. and C. Bordons (2007). "Nonlinear model predictive control: An introductory review. Assessment and Future Directions of Nonlinear Model Predictive Control." Springer**:** 1-16.

Camponogara, E., D. Jia, B. H. Krogh and S. Talukdar (2002). "Distributed model predictive control." Control Systems Magazine, IEEE **22**(1): 44-52.

Camporeale, S., B. Fortunato and M. Mastrovito (2006). "A modular code for real time dynamic simulation of gas turbines in simulink." Journal of Engineering for Gas Turbines and Power(Transactions of the ASME) **128**(3): 506-517.

Chase, D. and P. Kehoe (2000). "GE combined-cycle product line and performance." GE Power Systems, Schenectady, NY.

Chen, H. and F. Allgöwer (1998). "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability." Automatica **34**(10): 1205-1217.

Chen, W.-H. (2010). "Stability analysis of classic finite horizon model predictive control." International Journal of Control, Automation and Systems **8**(2): 187-197.

Chen, X., M. Heidarinejad, et al. (2012). "Distributed economic MPC: Application to a nonlinear chemical process network." Journal of Process Control. **22**:689-699.

Chen, X., J. Liu, D. M. de la Pena and P. D. Christofides (2010). "Sequential and iterative distributed model predictive control of nonlinear process systems subject to asynchronous measurements." Proceedings of the 9th International Symposium on Dynamics and Control of Process Systems, Leuven, Belgium: 611‑616.

Chien, K., E. Ergin, C. Ling and A. Lee (1958). "Dynamic analysis of a boiler." Trans. ASME **80**(4): 1719-1809.

Chokshi, N. N. and D. C. McFarlane (2008). A distributed coordination approach to reconfigurable process control, Springer.

Christofides, P. D., R. Scattolini, D. M. de la Pena and J. Liu (2012). "Distributed Model Predictive Control: A Tutorial Review." Proceedings of Chemical Process Control-8 **22**.

Christofides, P. D., R. Scattolini, D. M. de la Peña and J. Liu (2013). "Distributed Model Predictive Control: A Tutorial Review and Future Research Directions." Computers & Chemical Engineering **51:** 21-41.

Chui, C. K. and G. Chen (2009). "Kalman filtering: with real-time applications." Springer.

Çimen, T. (2012). "Survey of state-dependent Riccati equation in nonlinear optimal feedback control synthesis." Journal of Guidance, Control, and Dynamics **35**(4): 1025-1047.

Cloutier, J. R. (1997). "State-dependent Riccati equation techniques: an overview." Procedding of American Control Conference **2**: 932 - 936, IEEE.

Cloutier, J. R., C. N. D'Souza and C. P. Mracek (1996). "Nonlinear regulation and nonlinear H-infinity control via the state-dependent Riccati equation technique. I-Theory." International Conference on Nonlinear Problems in Aviation and Aerospace, 1 st, Daytona Beach, FL:117-130.

Committee, I. (1973). "Dynamic models for steam and hydro turbines in power system studies." IEEE Trans on Power Apparatus and Systems **92**(6): 1904-1914.

Cui, H. and E. W. Jacobsen (2002). "Performance limitations in decentralized control." Journal of Process Control **12**(4): 485-494.

De Mello, F. (1991). "Boiler models for system dynamic performance studies." Power Systems, IEEE Transactions on **6**(1): 66-74.

De Mello, F. (1991). "Dynamic models for fossil fueled steam units in power system studies." IEEE Transactions on Power Systems **6**(2): 753-761.

de Moura Oliveira, P. (2005). Modern heuristics review for PID control optimization: a teaching experiment. Control and Automation, 2005. ICCA'05. International Conference on, **2**:828-833, IEEE.

de Oliveira, N. and L. T. Biegler (1994). "Constraint handing and stability properties of model-predictive control." AIChE journal **40**(7): 1138-1155.

Di Rocco, M. and F. Pascucci (2007). "Sensor network localisation using distributed extended kalman filter." Advanced intelligent mechatronics, 2007 IEEE/ASME international conference on:1- 6, IEEE.

Dunbar, W. B. and R. M. Murray (2004). "Receding horizon control of multi-vehicle formations: A distributed implementation." Decision and Control, CDC Proceedings 43rd Conference on, **2**:1995-2002, IEEE.

Dutka, A. S., A. W. Ordys and M. J. Grimble (2003). "Non-linear predictive control of 2 dof helicopter model, Cecision and Control." CDC Proceedings 42nd Conference on, **4**: 3954-3959, IEEE.

Findeisen, R. and F. Allgöwer (2002). "An introduction to nonlinear model predictive control." 21st Benelux Meeeting on Systems and Control, Veldhoven.

Flynn, D. (2003). "Thermal power plant simulation and control." The Institution of Engineering and Technology,London,Uk.

Garriga, J. L. and M. Soroush (2010). "Model predictive control tuning methods: A review." Industrial & Engineering Chemistry Research **49**(8): 3505-3515.

Goncalves, E. N., R. M. Palhares and R. H. Takahashi (2008). "A novel approach for H2/H∞ robust PID synthesis for uncertain systems." Journal of Process Control **18**(1): 19-26.

Grimble, M. and A. Ordys (2001). "Predictive control for industrial applications." annual reviews in control **25**: 13-24.

Grimble, M. J. (2013). Benefits and problems arising from the use of advanced control methods in industrial applications. Speech presented at IET Conference of Control and Automation Birmingham, UK. (http://scpro.streamuk.com/uk/player/?g=7fc61860) (Last accessed date 08-05-2014).

Grosdidier, P. and M. Morari (1986). "Interaction measures for systems under decentralized control." Automatica **22**(3): 309-319.

Haessig, D. A. and B. Friedland (2002). "State dependent differential Riccati equation for nonlinear estimation and control." 15th Triennial World Congress of IFAC, Barcelona, Spain.

Harish, N. A., R. Ismail and T. Ahmad (2010). "Transformation of fuzzy state space model of a boiler system: a graph theoretic approach." WSEAS Transactions on Mathematics **9**(9): 669-678.

Herreros, A., E. Baeyens and J. R. Perán (2002). "Design of PID-type controllers using multiobjective genetic algorithms." ISA transactions **41**(4): 457-472.

Hooshmandi, K., Y. Alireza and M. Montazeri (2009). "Varying set-point with adaptive predictive supervisory controller for gas turbine of GILAN combined cycle." Advanced Intelligent Mechatronics, 2009. AIM 2009. IEEE/ASME International Conference on, IEEE.

Huang, R., S. C. Patwardhan and L. T. Biegler (2012). "Robust stability of nonlinear model predictive control with extended Kalman filter and target setting." International Journal of Robust and Nonlinear Control, 23: 1240-1264.

Hussain, A. and H. Seifi (1992). "Dynamic modelling of a single shaft gas turbine." IFAC symposium on control of power plants and power systems, Munich.

Iratni, A., R. Katebi and M. Mostefai (2012). "On-line robust nonlinear state estimators for nonlinear bioprocess systems." Communications in Nonlinear Science and Numerical Simulation **17**(4): 1739-1752.

Jaganath, C., A. Ridley and D. S. Bernstein (2005). "A SDRE-based asymptotic observer for nonlinear discrete-time systems." American Control Conference, 2005. Proceedings of the 2005, 5:3630-3635, IEEE.

Jia, D. and B. H. Krogh (2001). "Distributed model predictive control.", American Control Conference, Proceedings of the 2001,4:2767-2772,IEEE.

Katebi, M. and M. Johnson (1997). "Predictive control design for large-scale systems." Automatica **33**(3): 421-425.

Keerthi, S. a. and E. G. Gilbert (1988). "Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: Stability and moving-horizon approximations." Journal of Optimization Theory and Applications **57**(2): 265-293.

Kehlhofer, R., B. Rukes, F. Hannemann and F. X. Stirnimann (2009). "Combined-cycle gas & steam turbine power plants." PennWell.

Kouvaritakis, B., M. Cannon and J. Rossiter (1999). "Non-linear model based predictive control." International Journal of Control **72**(10): 919-928.

Lee, J. H. and N. L. Ricker (1994). "Extended Kalman filter based nonlinear model predictive control." Industrial & Engineering Chemistry Research **33**(6): 1530-1541.

Lee, M., K. Lee, C. Kim and J. Lee (2004). "Analytical design of multiloop PID controllers for desired closed-loop responses." AIChE journal **50**(7): 1631-1635.

Lee, Y., J. Lee and S. Park (2000). "PID controller tuning for integrating and unstable processes with time delay." Chemical Engineering Science **55**(17): 3481-3493.

Lee, Y., S. Park and M. Lee (2006). "Consider the generalized IMC-PID method for PID controller tuning of time-delay processes." Hydrocarbon Processing **85**(1): 87.

Lieslehto, J. and H. Koivo (1987). An expert system for interaction analysis of multivariable systems. American Control Conference, 1987, IEEE.

Liu, J., X. Chen, D. Muñoz De La Peña and P. D. Christofides (2010). "Sequential and iterative architectures for distributed model predictive control of nonlinear process systems." AIChE journal **56**(8): 2137-2149.

Liu, J., D. Muñoz de la Peña and P. D. Christofides (2009). "Distributed model predictive control of nonlinear process systems." AIChE journal **55**(5): 1171-1184.

Liu, J., D. Muñoz de la Peña and P. D. Christofides (2010). "Distributed model predictive control of nonlinear systems subject to asynchronous and delayed measurements." Automatica **46**(1): 52-61.

Lo, K. and Y. Rathamarit (2008). "State estimation of a boiler model using the unscented Kalman filter." Generation, Transmission & Distribution, IET **2**(6): 917-931.

Loh, A. P., C. C. Hang, C. K. Quek and V. U. Vasnani (1993). "Autotuning of multiloop proportional-integral controllers using relay feedback." Industrial & Engineering Chemistry Research **32**(6): 1102-1107.

Luyben, M. and W. Luyben (1997). "Essentials of Process Control McGraw-Hill Companies." Inc., New York.

Luyben, W. L. (1986). "Simple method for tuning SISO controllers in multivariable systems." Industrial & Engineering Chemistry Process Design and Development **25**(3): 654-660.

Maciejowski, J. M. (2002). Predictive control with constraints, Prentice Hall.

Maestre, J., D. Muñoz de la Peña and E. Camacho (2011). "Distributed model predictive control based on a cooperative game." Optimal Control Applications and Methods **32**(2): 153-176.

Magni, L. and R. Scattolini (2006). "Stabilizing decentralized model predictive control of nonlinear systems." Automatica **42**(7): 1231-1236.

Mayne, D. Q. (1973). "The design of linear multivariable systems." Automatica **9**(2): 201-207.

Mayne, D. Q., J. B. Rawlings, C. V. Rao and P. O. M. Scokaert (2000). "Constrained model predictive control: Stability and optimality." AUTOMATICA-OXFORD- **36**: 789-814.

McDonald, J. P. and H. Kwatny (1970). "A Mathematical Model for Reheat Boiler Turbine-generator Systems." In Proceedings of IEEE.PES winter power meeting, New York. Paper 70 CP221-PWR.

Menighed, K., C. Aubrun and J.-J. Yame (2009). "Distributed state estimation and model predictive control: Application to fault tolerant control." Control and Automation, 2009. ICCA 2009. IEEE International Conference on, IEEE.

Michalska, H. and D. Q. Mayne (1993). "Robust receding horizon control of constrained nonlinear systems." Automatic Control, IEEE Transactions on **38**(11): 1623-1633.

Mohammadpour Velni, J. and K. M. Grigoriadis (2008). "Delay-dependent $H\infty$ filtering for time-delayed LPV systems." Systems & Control Letters **57**(4): 290-299.

Mracek, C. P., J. Clontier and C. A. D'Souza (1996). "A new technique for nonlinear estimation." In Proceedings of the IEEE Conference on Control Applications, Dearborn,MI, IEEE.

Mrunalini, K., P. Kundu and K. Dutta (2006). "State Space Model for Drum Boiler System." Journal Insttitution of Engineers India Part El Electrical Engineering Division **86**(R): 260.

Muske, K. R. and T. A. Badgwell (2002). "Disturbance modeling for offset-free linear model predictive control." Journal of Process Control **12**(5): 617-632.

Mutambara, A. G. (1998). "Decentralized estimation and control for multisensor systems." CRC.

Nash, J. (1951). "Non-cooperative games." Annals of mathematics **54**(2): 286-295.

Necoara, I., C. Savorgnan, D. Tran, J. Suykens and M. Diehl (2009). "Distributed nonlinear optimal control using sequential convex programming and smoothing techniques." Proceedings of the 48th Conference on Decision and Control, Shanghai:543-548, IEEE.

Negenborn, R., B. De Schutter and J. Hellendoorn (2009). "Multi-agent model predictive control: A survey." Delft University of Technology Report No 04-010, Netherlands, Arxiv preprint arXiv:0908.1076.

Nemra, A. and N. Aouf (2010). "Robust INS/GPS sensor fusion for UAV localization using SDRE nonlinear filtering." Sensors Journal, IEEE **10**(4): 789-798.

Nicholson, H. (1964). "Dynamic optimisation of a boiler." Electronics and Power **10**(11): 384-385.

Olfati-Saber, R. (2007). "Distributed Kalman filtering for sensor networks." Decision and Control, 2007 46th IEEE Conference on, IEEE.

Ordys, A. and M. Grimble (2001). "Predictive control design for systems with the state dependent non–linearities." Proceeding of SIAM Conference on Control and its Applicatiions.

Ordys, A. W. and D. W. Clarke (1993). "A state-space description for GPC controllers." International Journal of Systems Science **24**(9): 1727-1744.

Ordys, A. W., A. W. Pike, M. A. Johnson, R. M. Katebi and M. J. Grimble (1994). "Modelling and simulation of power generation plants." Springer Verlag.

Orlowski, P. (2011). "Convergence of the Discrete-Time Nonlinear Model Predictive Control with Successive Time-Varying Linearization along Predicted Trajectories." Electronics And Electrical Engineering **113**(7): 27-31.

Prasad, G., G. Irwin, E. Swidenbank and B. Hogg (2002). "A hierarchical physical model-based approach to predictive control of a thermal power plant for ef" cient plant-wide disturbance rejection." Transactions of the Institute of Measurement and Control **24**(2): 107-128.

Qin, S. J. and T. A. Badgwell (2003). "A survey of industrial model predictive control technology." Control Engineering Practice **11**(7): 733-764.

Rao, B. and H. Durrant-Whyte (1991). "Fully decentralised algorithm for multisensor Kalman filtering." Control Theory and Applications, IEE Proceedings D, IET.

Rao, B., H. Durrant-Whyte and J. Sheen (1993). "A fully decentralized multi-sensor system for tracking and surveillance." The International Journal of Robotics Research **12**(1): 20-44.

Rawlings, J. B. and B. T. Stewart (2008). "Coordinating multiple optimization-based controllers: New opportunities and challenges." Journal of Process Control **18**(9): 839-845.

Ray, A. (1980). "Dynamic modelling of power plant turbines for controller design." Applied Mathematical Modelling **4**(2): 109-112.

Rayaprolu, K. (2010). "Boilers for power and process." CRC Press.

Rhine, J. M., R. J. Tucker and B. Gas (1991). "Modelling of gas-fired furnaces and boilers and other industrial heating processes." British Gas.

Richalet, J., A. Rault, J. Testud and J. Papon (1978). "Model predictive heuristic control: Applications to industrial processes." Automatica **14**(5): 413-428.

Richards, A. and J. How (2007). "Robust distributed model predictive control." International Journal of Control **80**(9): 1517-1531.

Rlvera, D. E., M. Morarl and S. Skogestad (1986). "Internal Model Control: PID Controller Design.", Industrial & engineering chemistry process design and development **25**(1) : 252-265.

Roshany-Yamchi, S., M. Cychowski, R. R. Negenborn, B. De Schutter, K. Delaney and J. Connell (2011). "Kalman Filter-Based Distributed Predictive Control of Large-Scale Multi-Rate Systems: Application to Power Networks." Control Systems Technology, IEEE Transactions on, **21** (1): 27-39.

Rowen, I. (1983). "Simplified mathematical representations of heavy-duty gas turbines." Journal of engineering for power **105**(4): 865-869.

Saez, D., F. Milla and L. S. Vargas (2007). "Fuzzy predictive supervisory control based on genetic algorithms for gas turbines of combined cycle power plants." Energy Conversion, IEEE Transactions on **22**(3): 689-696.

Saez, D., A. Ordys and M. Grimble (2005). "Design of a supervisory predictive controller and its application to thermal power plants." Optimal Control Applications and Methods **26**(4): 169-198.

Sáez, D., R. Zúniga and A. Cipriano (2008). "Adaptive hybrid predictive control for a combined cycle power plant optimization." International Journal of Adaptive Control and Signal Processing **22**(2): 198-220.

Scattolini, R. (2009). "Architectures for distributed and hierarchical model predictive control-a review." Journal of Process Control **19**(5): 723-731.

Scheu, H. and W. Marquardt (2011). "Distributed Model-Predictive Control Driven by Simultaneous Derivation of Prices and Resources." proceedings of the 2011 IFAC World Confgress.

Schobeiri, M., M. Attia and C. Lippke (1994). "GETRAN: A generic, modularly structured computer code for simulation of dynamic behavior of aero-and power generation gas turbine engines." Transactions-American Society of Mechanical Engineers journal of engineering for gas turbines and power **116**: 483-483.

Scokaert, P. O. and J. B. Rawlings (1999). "Feasibility issues in linear model predictive control." AIChE journal **45**(8): 1649-1659.

Shah, G. and S. Engell (2011). "Tuning MPC for desired closed-loop performance for MIMO systems." American Control Conference (ACC), 2011, IEEE.

Shakouri, P. and A. Ordys (2011). "Application of the state-dependent nonlinear model predictive control in adaptive cruise control system." Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on, IEEE.

Shamma, J. S. and M. Athans (1991). "Guaranteed properties of gain scheduled control for linear parameter-varying plants." Automatica **27**(3): 559-564.

Simonetto, A., T. Keviczky and R. Babuška (2010). "Distributed Nonlinear Estimation for Robot Localization using weighted consensus." IEEE international conference of Robotics and Automation (ICRA), Anchorage,AK:30226-3031.

Skogestad, S. (2003). "Simple analytic rules for model reduction and PID controller tuning." Journal of Process Control **13**(4): 291-309.

Skogestad, S. and M. Morari (1987). "Implications of large RGA-elements on control performance." Industrial & Engineering Chemistry Research **26**(11): 2323-2330.

Skogestad, S. and M. Morari (1989). "Robust performance of decentralized control systems by independent designs." Automatica **25**(1): 119-125.

Sørensen , M. and S. Kristiansen (2007). "Model Predictive Control for an artificial pancreas." Technical University of Denmark Kongens Lyngby **Msc**.

Stewart, B. T., A. N. Venkat, J. B. Rawlings, S. J. Wright and G. Pannocchia (2010). "Cooperative distributed model predictive control." Systems & Control Letters **59**(8): 460-469.

Stewart, B. T., S. J. Wright and J. B. Rawlings (2011). "Cooperative distributed model predictive control for nonlinear systems." Journal of Process Control **21**(5): 698-704.

Takahashi, R. H., P. Peres and P. Ferreira (1996). Guaranteed cost PID design for uncertain systems: a multiobjective approach. Computer-Aided Control System Design, 1996., Proceedings of the 1996 IEEE International Symposium on, IEEE.

Tomlinson, L. O. and S. McCullough (1996). "Single-Shaft Combined-Cycle Power Generation System." GER-3767C, GE Power Systems, Schenectady, New York.

Vaccarini, M., S. Longhi and M. R. Katebi (2009). "Unconstrained networked decentralized model predictive control." Journal of Process Control **19**(2): 328-339.

Vadigepalli, R. and F. J. Doyle Iii (2003). "Structural analysis of large-scale systems for distributed state estimation and control applications." Control Engineering Practice **11**(8): 895-905.

Venkat, A. N., J. B. Rawlings and S. J. Wright (2005). "Stability and optimality of distributed model predictive control." In Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on: 6680-6685, IEEE.

Vu, T. N. L., J. Lee and M. Lee (2007). "Design of multi-loop PID controllers based on the generalized IMC-PID method with Mp criterion." International journal of Control Automation and systems **5**(2): 212.

Vu, T. N. L. and M. Lee (2010). "Independent design of multi-loop PI/PID controllers for interacting multivariable processes." Journal of Process Control **20**(8): 922-933.

Wills, A. (2004). "Technical Report EE04025-Notes on Linear Model Predictive Control." School of Electrical Engineering and Computer Science, University of Newcastle, Tech. Rep.

Wittchow, E. (2002). "3.7 Power plant operation. Fossil Energy", Volume 3 Energy Technologies' of Landolt-Börnstein- Group VIII Advanced Materials and Technologies, Springer**:** 144-148.

Wood, J. (2008). "Local Energy: Distributed generation of heat and power, Institution of Engineering and Technology." IET **55**.

Xu, Z., J. Zhao, J. Qian and Y. Zhu (2009). "Nonlinear MPC using an identified LPV model." Industrial & Engineering Chemistry Research **48**(6): 3043-3051.

Youssef, A., M. Grimble, A. Ordys, A. Dutka and D. Anderson (2003). "Robust nonlinear predictive flight control." In European Control Conference ECC 2003, University of Cambridge, UK.

Yu, C. C. (2006). "Autotuning of PID Controllers: A Relay Feedback Approach." Springer.

Zhang, Y. and S. Li (2007). "Networked model predictive control based on neighbourhood optimization for serially connected large-scale processes." Journal of Process Control **17**(1): 37-50.

Zheng, A. and M. Morari (1995). "Stability of model predictive control with mixed constraints." Automatic Control, IEEE Transactions on **40**(10): 1818-1823.

Zheng, Y., S. Li and N. Li (2011). "Distributed model predictive control over network information exchange for large-scale systems." Control Engineering Practice. 19:757-769.

Zheng, Y., S. Li and X. Wang (2009). "Distributed model predictive control for plant-wide hot-rolled strip laminar cooling process." Journal of Process Control **19**(9): 1427-1437.

APPENDICES

Appendix A: CCPP Model Equations

A.1 Boiler Model

A.1.1 Furnace Model

The furnace model is described by 13 algebraic equations and 2 differential equations as follows:

Algebraic equations:

$$h_{EG} = \frac{x_{f1}}{\rho_{EG}} \tag{A.1}$$

$$T_g = \frac{h_{EG} - h_{ref}}{c_{pg}} + T_{ref} \tag{A.2}$$

$$p_G = R_{EG}\rho_{EG}T_g \tag{A.3}$$

$$w_{EG} = k_f p_G \tag{A.4}$$

$$Q_{ir} = \theta k V_f \sigma T_g^4 \tag{A.5}$$

$$Q_{is} = (1-\theta) k V_f \sigma T_g^4 \tag{A.6}$$

$$Q_{gs} = Q_{is} + k_{gs} w_{EG}^{0.6}(T_g - T_{st}) \tag{A.7}$$

$$T_{gr} = T_g + \frac{1}{c_{gs}}\frac{1}{w_{EG}}(Q_{is} - Q_{gs}) \tag{A.8}$$

$$Q_{rs} = k_{rs} w_{EG}^{0.6}(T_{gr} - T_{rh}) \tag{A.9}$$

$$T_{ge} = T_{gr} - \frac{1}{c_{gs}}\frac{1}{w_{EG}}Q_{rs} \tag{A.10}$$

$$Q_{es} = k_{es} w_{EG}^{0.6}(T_{ge} - T_{et}) \tag{A.11}$$

$$T_{g1} = T_{ge} - \frac{1}{c_{gs}} \frac{1}{w_{EG}} Q_{es} \tag{A.12}$$

$$y = 100(w_A + \gamma w_G - w_f R_s) \frac{1}{w_F R_s} \tag{A.13}$$

Differential equations:

$$\frac{dx_{f1}}{dt} = \frac{1}{V_f}\left(c_f w_f + h_A w_A + h_G w_G - Q_{ir} - Q_{is} - w_{EG} R_s (1 + \frac{y}{100}) h_{EG} \right) \tag{A.14}$$

$$\frac{d\rho_{EG}}{dt} = \frac{1}{V_f}\left(w_f + w_A + w_G - w_{EG} \right) \tag{A.15}$$

The parameters and inputs, which are necessary for the definition and formulation of the SIMULINK model, as well as the control models, of the furnace are listed in Table A.1 and A.2 respectively:

Table A.1: Parameters of Furnace model:

Symbol	Description	Values
K_f	Chimney flow coefficient (m.s)	0.001
K	Attenuation coefficient	0.18
k_{gs}	Experimental heat transfer coefficient to Superheater (J/kgK)	3532
C_{gs}	Combustion gas specific heat capacity (J.s/(kg.K))	1045
k_{rs}	Experimental heat transfer coefficient to the Reheter (J/kgK)	1.3926×10^4
V_f	Combustion chamber volume (m^3)	2000
C_f	Fuel caloric value (J/kg)	2.91×10^7
R_s	Stoichiometric air/fuel ratio	3.5
γ	Content of fresh air in exhaust gas turbine	0.1
k_{es}	Experimental heat transfer coefficient to Economizer (J/kgK)	247.549

Table A.2: Inputs to Furnace model:

Symbol	Description	Values
w_f	Fuel flow to the furnace (kg/s)	14.083
w_A	Air flow to the furnace (kg/s)	64.093
h_G	Enthalpy of exhaust gas from the gas turbine (J/kg)	6.9×10^5
w_G	Exhaust gas flow from the gas turbine (kg/s)	23.168
θ	Tilt angle coefficient ($0<\theta<1$)	0.88041
T_{st}	Temperature of Superheater metal tubes (^oK)	737.06
T_{rh}	Temperature of Reheater metal tubes (^oK)	743.66
T_{et}	Temperature of Economizer metal tubes (^oK)	412
h_A	Inlet air enthalpy (J/kg)	2.5×10^5

A.1.2 Superheater and attemperator model

The following equations are used to simulate the superheater and attemperator model.

Algebraic equations:

$$h_s = \frac{x_{s1}}{\rho_s} \tag{A.16}$$

$$T_s = \frac{h_s - h_{ref}}{c_{ps}} + T_{ref} \tag{A.17}$$

$$P_s = R_s \rho_s T_s \tag{A.18}$$

$$h_f = -0.0892 P_s + 2.2464\times10^6 \tag{A.19}$$

$$w_v = \sqrt{\frac{(P_V - P_s)\rho_V}{f_s}} \tag{A.20}$$

$$Q_s = K_s w_v^{0.8} (T_{st} - T_s) \tag{A.21}$$

Differential equations:

$$\frac{d\rho_s}{dt} = \frac{1}{V_s}\left(w_v - w_s\right) \tag{A.22}$$

$$\frac{dT_{st}}{dt} = \frac{1}{M_s C_{st}}\left(Q_{gs} - Q_s\right) \tag{A.23}$$

$$\frac{dx_{s1}}{dt} = \frac{1}{V_s}\left[Q_s + w_v h_v - w_s h_s + (h_a - h_f)w_a\right] \tag{A.24}$$

Parameters and inputs of superheater and attemperator model are listed in Table A.3 and A.4 respectively.

Table A.3: Parameters of Superheater and attemperator model:

Symbol	Description	Values
f_s	Superheater friction coefficient	2615
K_s	Experimental heat transfer coefficient [J/(kg. K)]	4.37×10^4
V_s	Superheater volume (m^3)	8.462
M_s	Superheater mass (kg)	1.04×10^4
C_{st}	Heat capacitance of Superheater tubes [J/(kg. K)]	481.4
$C_{p\text{-}ref}$	Ideal gas reference specific heat [J/(kg. K)]	2330
T_{ref}	Ideal gas reference temperature (K)	723.15
h_{ref}	Ideal gas reference specific enthalpy (J/kg)	3.32×10^6

Table A.4: Inputs to Superheater and attemperator model:

Symbol	Description	Values
w_a	Attemporation water flow (kg/s)	0
w_s	Steam flow from the Superheater (kg/s)	12
P_V	Steam drum pressure (Pa)	4.5417×10^6
ρ_V	Density of saturated steam from the drum (kg/m^3)	22.763
Q_{gs}	Heat flow from the furnace (J/s)	6.1626×10^6
h_V	Specific enthalpy of saturated steam from the drum (J/kg)	2.7977×10^6
h_a	Specific enthalpy of Attemporation water (J/kg)	5.5217×10^5

A.1.3 Economizer model

The economizer heat exchanger model is described by the following equations:

Algebraic equations:

$$h_{eo} = \frac{x_{e1}}{\rho_e} \tag{A.25}$$

Approximation from steam tables:

$$\begin{aligned} h_{ei} &= 0.5563P_{ei} + 3.619\times10^{5} \\ T_{eo} &= 268.3632 + 0.26922\times10^{-3}h_{eo} - 0.34182\times10^{-10}h_{eo}^{2} \\ P_{eo} &= 0.1245h_{eo} - 9.7369\times10^{-7}h_{eo}^{2} + 3.0143\times10^{-12}h_{eo}^{3} \qquad if\ h_{eo} \le 567680 \\ P_{eo} &= 2.3815\times10^{6} - 10.1102h_{eo} + 1.0905\times10^{-5}h_{eo}^{2} \qquad if\ h_{eo} > 567680 \end{aligned} \tag{A.26}$$

$$Q_e = K_e w_{ei}^{0.8}\left(T_{et} - T_{eo}\right) \tag{A.27}$$

Differential equations:

$$\frac{d\rho_e}{dt} = \frac{1}{V_e}\left(w_{ei} - w_{eo}\right) \tag{A.28}$$

$$\frac{dT_{et}}{dt} = \frac{1}{M_e C_e}\left(Q_{es} - Q_e\right) \tag{A.29}$$

$$\frac{dx_{e1}}{dt} = \frac{1}{V_e}\left(Q_e + w_{ei}h_{ei} - w_{eo}h_{eo}\right) \tag{A.30}$$

Parameters and inputs of economizer model are listed in Table A.5 and A.6 respectively.

Table A.5: Parameters of Economizer model:

Symbol	Description	Values
K_e	Experimental heat transfer coefficient [J/(kg. K)]	4.37×10^{4}
V_e	Economizer volume (m^3)	3
M_e	Economizer mass (kg)	7000
C_e	Heat capacitance of Economizer tubes [J/(kg. K)]	481

Table A.6: Inputs to Economizer model:

Symbol	Description	Values
P_{ei}	Inlet water pressure (Pa)	1.8339×10^5
w_{ei}	Inlet feed water flow (kg/s)	12.5
w_{eo}	Outlet water flow (kg/s)	12
Q_{es}	Heat absorbed by the economizer (J/s)	1.2465×10^5

A.1.4 Reheater model

The Reheater heat exchanger model is described by the following equations:

Algebraic equations:

$$h_{ro} = \frac{x_{RH1}}{\rho_{rh}} \tag{A.31}$$

$$T_r = \frac{h_{ro} - h_{ref}}{c_{pr}} + T_{ref} \tag{A.32}$$

$$P_{ro} = R_r \rho_{rh} T \tag{A.33}$$

$$Q_{rh} = K_{rh} w_{ri}^{0.8} \left(T_{rh} - T_r\right) \tag{A.34}$$

Differential equations:

$$\frac{d\rho_{rh}}{dt} = \frac{1}{V_{rh}} \left(w_{ri} - w_{ro}\right) \tag{A.35}$$

$$\frac{dT_{rh}}{dt} = \frac{1}{M_r C_{rh}} \left(Q_{rs} - Q_{rh}\right) \tag{A.36}$$

$$\frac{dx_{RH1}}{dt} = \frac{1}{V_{rh}} \left[Q_{rh} + w_{ri} h_{ri} - w_{ro} h_{ro}\right] \tag{A.37}$$

$$\frac{dw_{ro}}{dt} = \frac{1}{\tau_{rh}} \left(w_{ri} - w_{ro}\right) \tag{A.38}$$

Parameters and inputs of Reheater model are listed in Table A.7 and A.8 respectively.

Table A.7: Parameters to the Reheater model:

Symbol	Description	Values
K_{rh}	Experimental heat transfer coefficient [J/(kg. K)]	2.95×10^{4}
V_{rh}	Reheater volume (m^3)	10
M_r	Reheater mass (kg)	7000
C_{rh}	Heat capacitance of Reheater tubes [J/(kg. K)]	481
$C_{p\text{-}ref}$	Ideal gas reference specific heat [J/(kg. K)]	2200
T_{ref}	Ideal gas reference temperature (K)	723.16
h_{ref}	Ideal gas reference specific enthalpy (J/kg)	3.3244×10^{6}

Table A.8: Inputs to the Reheater model

Symbol	Description	Values
w_{ri}	Flow of steam at the inlet to the reheater (kg/s)	10.459
h_{ri}	Specific enthalpy of inlet steam (J/kg)	3.0298×10^{6}
Q_{rs}	Heat flow from the furnace (W)	3.1748×10^{6}

A.2 Gas turbine Model

A.2.1 Compressor

The Compressor model is described by the following algebraic equations:

$$m_a = \frac{\gamma_a}{\gamma_a - (\gamma_a - 1)/\eta_{\infty c}} \tag{A.39}$$

$$r_c^{\left(\frac{2}{m_a}\right)} - r_c^{\left(\frac{m_a+1}{m_a}\right)} = \frac{\eta_{\infty c}(m_a - 1)}{2m_a}\frac{1}{\rho_i P_{cin}}\left(\frac{w_a}{A_0}\right)^2 \tag{A.40}$$

$$P_{cout} = P_{cin} r \tag{A.41}$$

$$T_{cout} = T_{cin} r_c^{\left(\frac{\gamma_a - 1}{\gamma_a \eta_{\infty c}}\right)} \tag{A.42}$$

$$\eta_c = \frac{1 - r_c^{\left(\frac{\gamma_a - 1}{\gamma_a}\right)}}{1 - r_c^{\left(\frac{\gamma_a - 1}{\gamma_a \eta_{\infty c}}\right)}} \tag{A.43}$$

$$\Delta h_i = c_{pair} T_{cin} (r_c^{\left(\frac{R_{air}}{c_{pair}}\right)} - 1) \tag{A.44}$$

$$P_c = \frac{w_a \Delta h_i}{\eta_c \eta_{trans}} \tag{A.45}$$

Parameters and inputs of gas turbine compressor model are listed in Table A.9 and A.10 respectively

Table A.9: Parameters of gas turbine Compressor model:

Symbol	Description	Values
γ_a	Ratio of specific heats for air (C_p/C_v)	1.4
$\eta_{\infty c}$	Polytrophic efficiency of compressor	0.9
A_0	Exit flow area (m^2)	0.01
R_{air}	Ideal gas constant for air [J/(kg ^oK)]	287
C_{pair}	Specific heat at constant pressure for air [J/(kg ^oK)]	1005
η_{trans}	Transmission efficiency	0.99

Table A.10: Inputs to gas turbine Compressor model:

Symbol	Description	Values
W_a	Inlet air flow (Kg/s)	46.137
ρ_i (rhoi)	Inlet air density (Kg/m^3)	1.21
P_{cin}	Inlet air pressure (Pa)	1.0×10^5
T_{cin}	Inlet air temperature (^oK)	288

A.2.2 Combustion chamber

The combustion chamber model is described by the following algebraic equations:

$$w_G = w_a + w_f + w_{is} \tag{A.46}$$

$$T_{Tin} = \frac{1}{w_G}\frac{1}{c_{pg}}\left[w_a c_{pa}(T_{cout} - 298) + w_{is} c_{ps}(T_{is} - 298) - w_f \Delta h_{25} \right] \tag{A.47}$$

$$\Delta p = P_{cout}\left[\left(k_1 + k_2 \left(\frac{T_{Tin}}{T_{cout}} - 1 \right) \right) \frac{R_{cg}}{2} \left(\frac{w_G}{A_m P_{cout}} \right)^2 T_{cout} \right] \tag{A.48}$$

$$P_{Tin} = P_{cout} - \Delta p \tag{A.49}$$

$$h_{Tin} = h_{ref} + c_{pg}(T_{Tin} - T_{ref}) \tag{A.50}$$

$$g_{cnox} = 23.333\left(\frac{w_{is}}{w_f} \right)^2 - 94.333\left(\frac{w_{is}}{w_f} \right) + 100 \tag{A.51}$$

$$g_{cco} = 38.1959\left(\frac{w_{is}}{w_f} \right)^2 - 25.4874\left(\frac{w_{is}}{w_f} \right) + 8 \tag{A.52}$$

Parameters and inputs of gas turbine combustion chamber model are listed in Table A.11 and A.12 respectively

Table A.11: Parameters of gas turbine combustion chamber model:

Symbol	Description	Values
c_{pg}	Specific heats at constant pressure for outlet gas [J/(kg ^oK)]	1144
c_{pa}	Specific heats at constant pressure for air [J/(kg ^oK)]	1005
c_{ps}	Specific heats at constant pressure for steam [J/(kg ^oK)]	2160
Δh_{25}	Fuel enthalpy of combustion at reference temperature J/kg	-4.0×10^7
K_1	Combustion chamber pressure drop empirical coefficient	1
K_2	Combustion chamber pressure drop empirical coefficient	0.9345
R_{cg}	Ideal gas constant for combustion gases [J/(kg ^oK)]	287
A_m	Combustion chamber cross sectional area (m^2)	1
h_{ref}	Reference combustion gas enthalpy [J/(kg ^oK)]	1.2041×10^6
T_{ref}	Reference combustion gas enthalpy (^oK)	1000

Table A.12: Inputs to gas turbine combustion chamber model:

Symbol	Description	Values
P_{cout}	Outlet air pressure (Pa)	1.0033×10^{6}
T_{cout}	Outlet air temperature (^{o}K)	598.8
w_a	Air flow to combustor from compressor (kg/s)	46.137
w_f	Fuel flow to combustor (kg/s)	1.8484
w_{is}	Injected steam flow to combustor (kg/s)	0.18566
T_{is}	Temperature of injected steam (^{o}K)	601.69

A.2.3 The turbine

The following algebraic equations are used to build the turbine model:

$$m_{cg} = \frac{\gamma_{cg}}{\gamma_{cg} - \eta_{\infty T}(\gamma_{cg} - 1)} \tag{A.53}$$

$$\rho_{Tin} = \frac{P_{Tin}}{R_{Cg} T_{Tin}} \tag{A.54}$$

$$r_T^{\left(\frac{2}{m_{cg}}\right)} - r_T^{\left(\frac{m_{cg}+1}{m_{cg}}\right)} = \frac{(m_{cg} - 1)}{2\eta_{\infty T} m_{cg} \rho_{Tin} P_{Tin}} \left(\frac{w_g}{A_{T0}}\right)^2 \tag{A.55}$$

$$P_{Tout} = P_{Tin} r_T \tag{A.56}$$

$$T_{Tout} = T_{Tin} r_T^{\left(\frac{\eta_{\infty T}(\gamma_{cg}-1)}{\gamma_{cg}}\right)} \tag{A.57}$$

$$\Delta h_i = c_{pg} T_{cin} (r_T^{\left(\frac{R_{cg}}{c_{pg}}\right)} - 1) \tag{A.58}$$

$$\eta_T = \frac{1 - r_T^{\left(\frac{\eta_{\infty T}(\gamma_{cg}-1)}{\gamma_{cg}}\right)}}{1 - r_T^{\left(\frac{(\gamma_{cg}-1)}{\gamma_{cg}}\right)}} \tag{A.59}$$

$$P_T = \eta_T w_G \Delta h \tag{A.60}$$

$$h_{Tout} = h_{Tin} + \eta_T \Delta h \tag{A.61}$$

$$P_{mech} = P_T - P_c \tag{A.62}$$

Parameters and inputs of turbine model are listed in Table A.13 and A.14 respectively.

Table A.13: Parameters of turbine model:

Symbol	Description	Values
γ_{cg}	Ratio of specific heats for combustion gases	1.333
$\eta_{\infty T}$	Polytrophic efficiency of Turbine	0.9
A_{T0}	Turbine exit flow area (m^2)	0.14
R_{cg}	Ideal gas constant for combustion gases [J/(kg ^oK)]	287
C_{pg}	Specific heat at constant pressure for air [J/(kg ^oK)]	1144
r_{T0}	Nominal outlet/inlet pressure ratio for turbine	0.1

Table A.14: Inputs to turbine model:

Symbol	Description	Values
w_g	Inlet gas flow from combustor (kg/sec)	46.939
T_{Tin}	Inlet gas temperature (^oK)	1892
P_{Tin}	Inlet gas pressure (Pa)	1.0027×10^6
P_c	Compressor power consumption (W)	1.4547×10^7
h_{Tin}	Inlet gas enthalpy (J/kg)	1.1804×10^6

A.3 Steam turbine Model

The same set of equations is used to model HP and LP sections. In IP section, the storage dynamics equations are not applied, because they are modeled in the Reheater. Turbines equations are described by the similar algebraic equations as in the turbine module of the gas turbine:

Algebraic equations:

$$\gamma = \frac{c_p}{c_p - R_{shst}} \tag{A.63}$$

$$h_o = \frac{x_o}{\rho_o} \tag{A.64}$$

$$T_o = \frac{h_o - h_{in}}{c_p} + T_{in} \tag{A.65}$$

$$P_o = R\rho_o T_o \tag{A.66}$$

$$m = \frac{\gamma}{\gamma - \eta_\infty(\gamma - 1)} \tag{A.67}$$

$$r^{\left(\frac{2}{m}\right)} - r^{\left(\frac{m+1}{m}\right)} = \frac{(m-1)}{2\eta_\infty m \rho_o P_o}\left(\frac{w_{ou}}{A}\right)^2 \tag{A.68}$$

$$P_{ou} = P_o r \tag{A.69}$$

$$T_{ou} = T_o r^{\left(\frac{\eta_\infty(\gamma-1)}{\gamma}\right)} \tag{A.70}$$

$$\Delta h_i = c_p T_o (r^{\left(\frac{R}{c_p}\right)} - 1) \tag{A.71}$$

$$\eta = \frac{1 - r^{\left(\frac{\eta_{\infty T}(\gamma-1)}{\gamma}\right)}}{1 - r^{\left(\frac{(\gamma-1)}{\gamma}\right)}} \tag{A.72}$$

$$P = \eta\, w_{ou} \Delta h \tag{A.73}$$

$$h_{ou} = h_o + c_p\left(T_{ou} - T_o\right) \tag{A.74}$$

Differential equations:

$$\frac{d\rho_{ou}}{dt} = \frac{1}{V}(w_{in} - w_{ou}) \tag{A.75}$$

$$\frac{dw_{ou}}{dt} = \frac{1}{\tau_s}(w_{in} - w_{ou}) \tag{A.76}$$

$$\frac{dx_{ou}}{dt} = \frac{1}{V}(w_{in}h_{in} - w_{ou}h_o) \tag{A.77}$$

The equations (A.64), (A.65), (A.66), (A.75), (A.76) and (A.77) are not applied to IP section. Parameters and inputs of steam turbine model are listed in Table A.15 and A.16 respectively.

Table A.15: Parameters of Hp turbine model:

Symbol	Description	Values
τ_s	Hp section mass flow time constant (sec)	1
η_∞	Polytrophic efficiency of Turbine	0.8
A_{T0}	Turbine exit flow area (m^2)	0.0032
C_{pg}	Specific heat at constant pressure for high pressure steam [J/(kg ^oK)]	2430
r_{T0}	Nominal outlet/inlet pressure ratio for turbine	0.33
V	Hp section steam storage volume (m^3)	5.664

Table A.16: Inputs to Hp turbine model:

Symbol	Description	Values
w_{in}	Inlet steam flow from boiler (kg/sec)	12
T_{Tin}	Inlet steam temperature (^oK)	717.72
P_{Tin}	Inlet steam pressure (Pa)	4.5251×10^6
ρ_{in}	Inlet steam flow density (kg/m^3)	13.662
h_{Tin}	Inlet steam enthalpy (J/kg)	3.3117×10^6

Appendix B: MATLAB block diagrams of CCPP Model

B1.1 CCPP Model

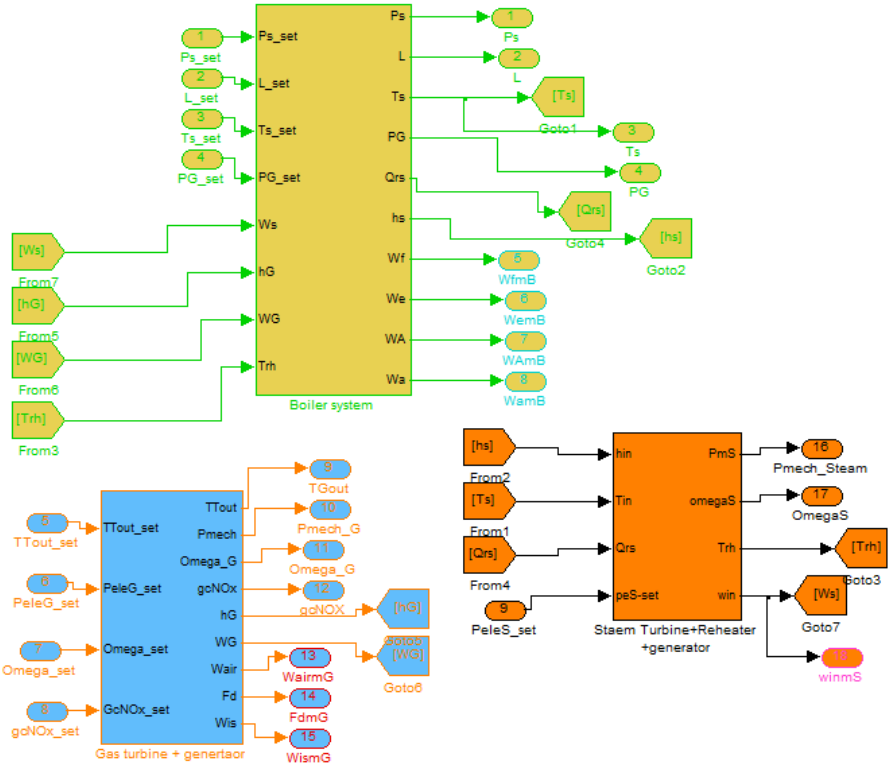


Figure B1: Block diagram of CCPP system

B1.2 Boiler System

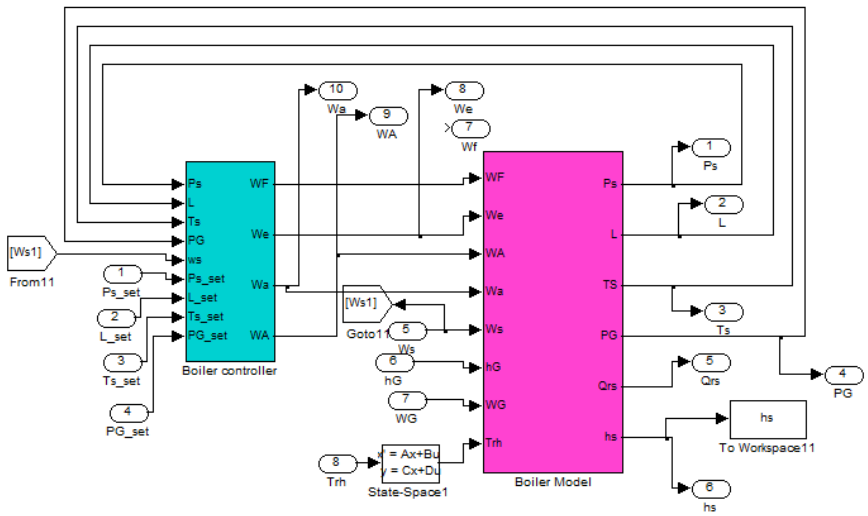


Figure B2: Block diagram of Boiler system

B1.3 Boiler model

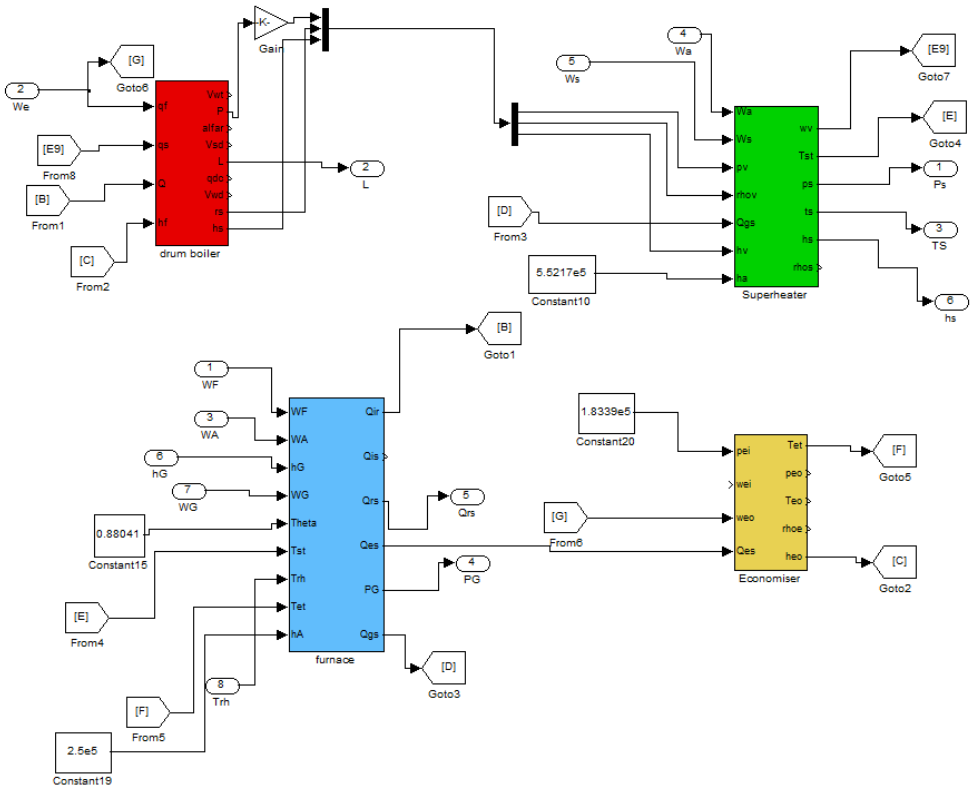


Figure B3: Block diagram of boiler model

B1.4 Gas Turbine System

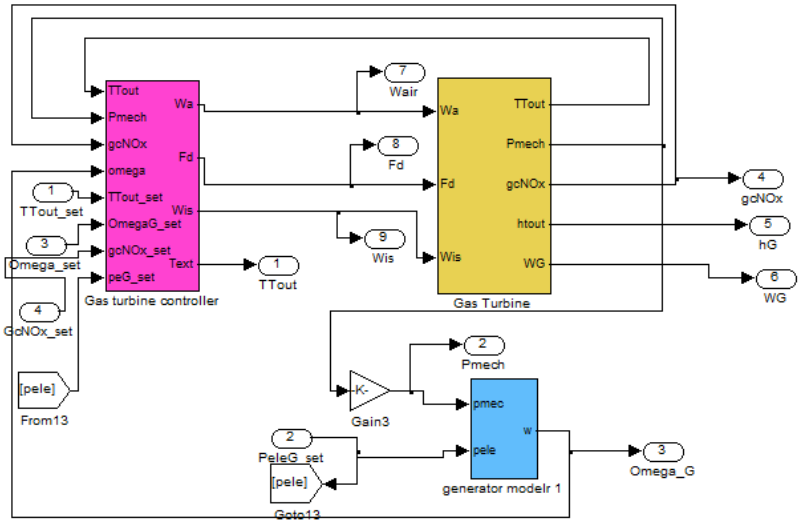


Figure B4: Block diagram of gas turbine system

B1.5 Gas Turbine model

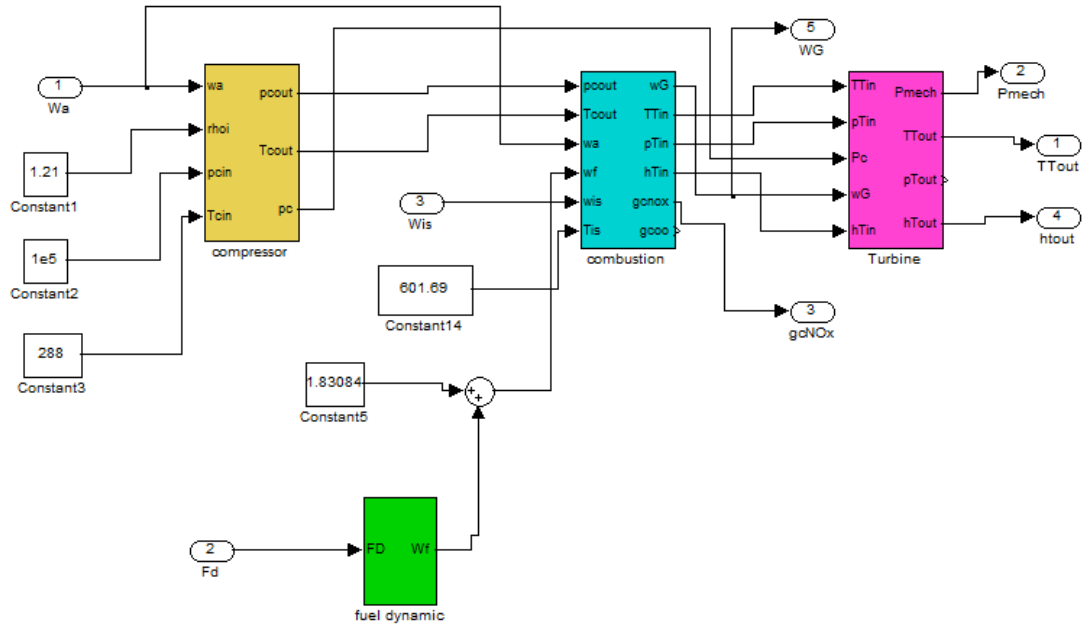


Figure B5: Block diagram of gas turbine model

B1.6 Steam Turbine System

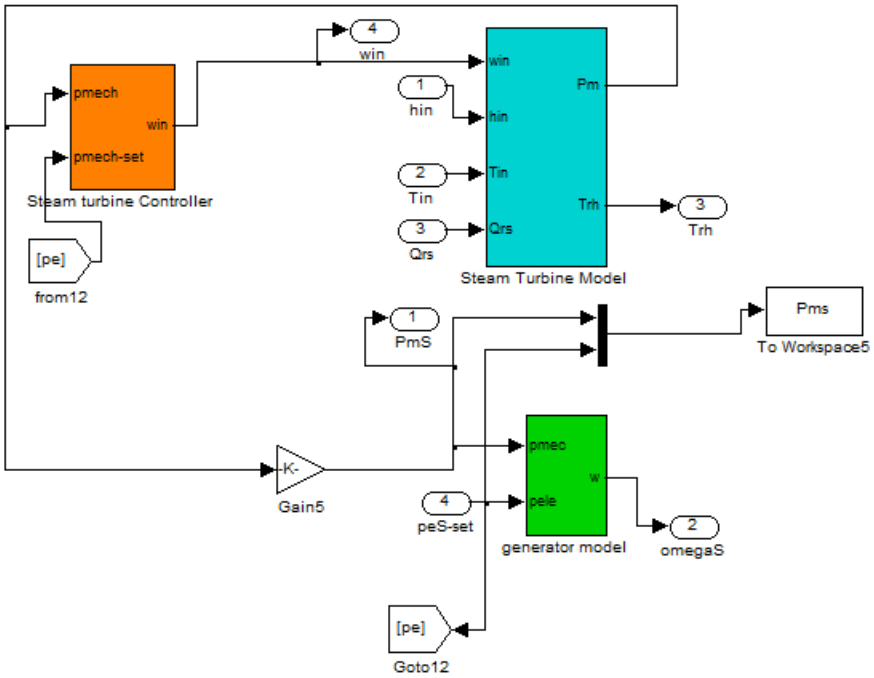


Figure B6: Block diagram of steam turbine system

B1.7 Steam Turbine Model

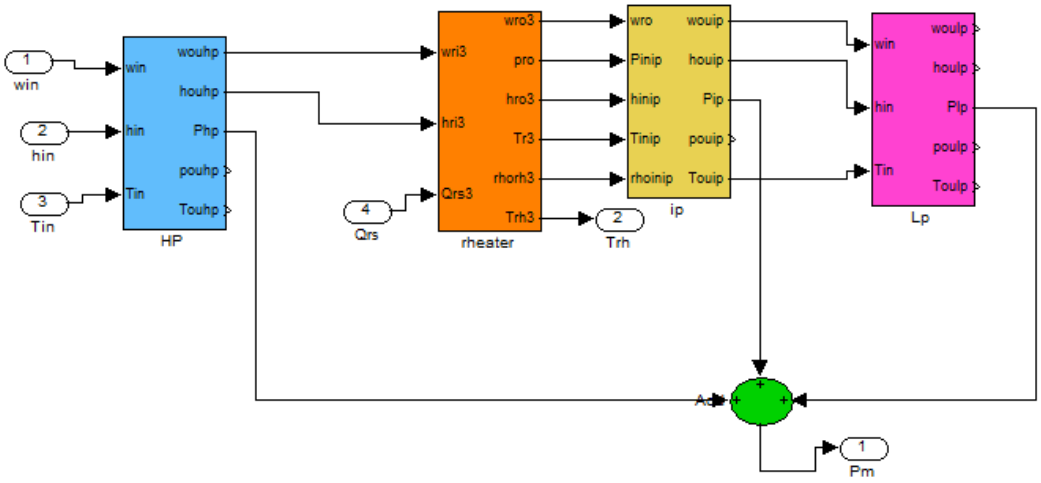


Figure B7: Block diagram of steam turbine model

Appendix C: MATLAB block diagrams of DMPC Algorithms

C1.1 Algorithm 1

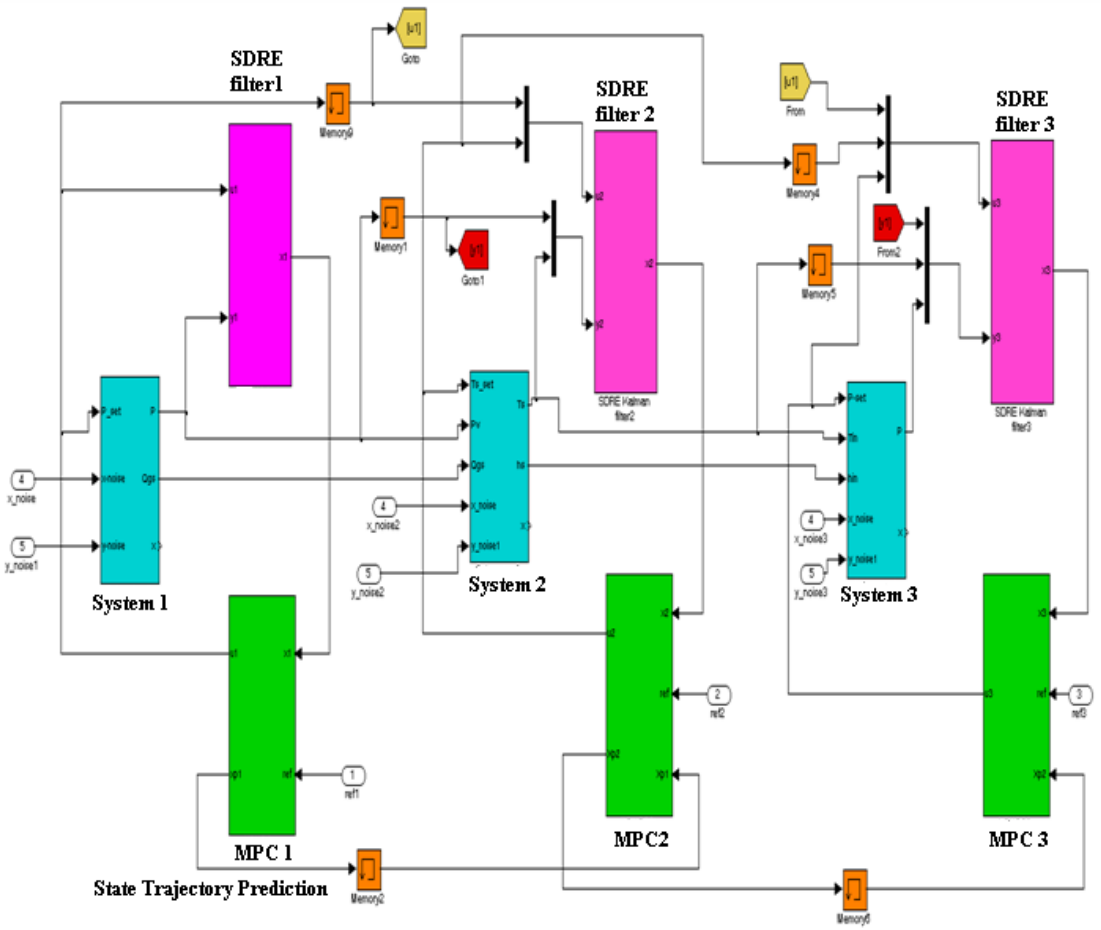


Figure C1: Algorithm 1 MATLAB block diagram

C1.2 Algorithm 2

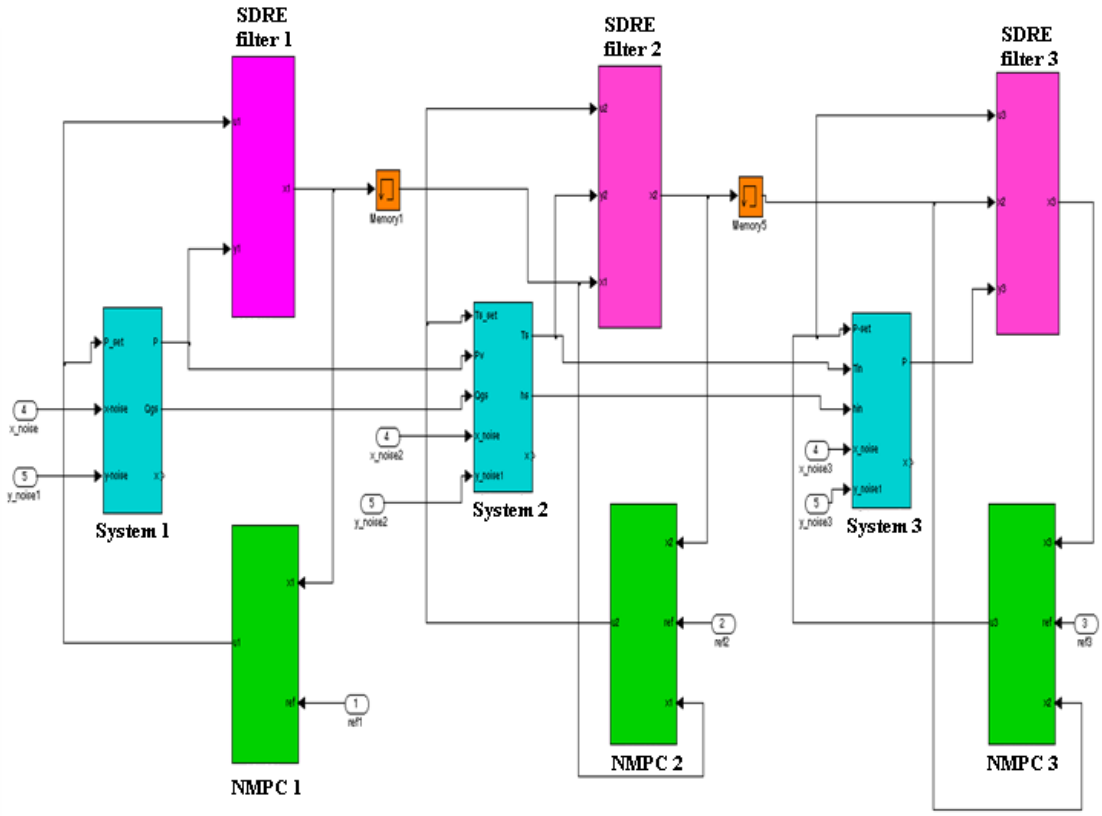


Figure C2: Algorithm 2 MATLAB block diagram