DEVELOPMENT OF ADAPTIVE AND FACTORIZED NEURAL MODELS FOR MPC OF INDUSTRIAL SYSTEMS

DESMOND KEAN SIONG TOK

A thesis submitted in partial fulfillment of the requirements of Liverpool John Moores University for the degree of Doctor of Philosophy

October 2016

ABSTRACT

Many industrial processes have non-linear and time-varying dynamics, for which the control and optimization require further investigations. Adaptive modelling techniques using radial basis function (RBF) networks often provide competitive modelling performances but encounter slow recovery speed when processes operating regions are shifted largely. In addition, RBF networks based model predictive control results as a non-linear programming problem, which restricts the application to fast dynamic systems. To these targets, the thesis presents the development of adaptive and factorized RBF network models. Model predictive control (MPC) based on the factorized RBF model is applied to a non-linear proton exchange membrane fuel cell (PEMFC) stack system. The main contents include three parts: RBF model adaptation; model factorization and fast long-range prediction; and MPC for the PEMFC stack system.

The adaptive RBF model employs the recursive orthogonal least squares (ROLS) algorithm for both structure and parameter adaptation. In decomposing the regression matrix of the RBF model, the R matrix is obtained. Principles for adding centres and pruning centres are developed based on the manipulation of the R matrix. While the modelling accuracy is remained, the developed structure adaptation algorithm ensures the model size to be kept to the minimum. At the same time, the RBF model parameters are optimized in terms of minimum Frobenius norm of the model prediction error. A simulation example is used to evaluate the developed adaptive RBF model, and the model performance in output prediction is superior over the existing methods.

Considering that a model with fast long-range prediction is needed for the MPC of fast dynamic systems, a f-step factorization algorithm is developed for the RBF model. The model structure is re-arranged so that the unknown future process outputs are not required for output prediction. Therefore, the accumulative error caused by recursive calculation in normal neural network model is avoided. Furthermore, as the information for output prediction is explicitly divided into the past information and the future information, the optimization of the control variable in the MPC based on this developed factorized model can be solved much faster than the normal NARX-RBF model. The developed model adaptation algorithm can be applied to this f-step factorized model to achieve fast and adaptive model prediction.

Finally, the developed factorized RBF model is applied to the MPC of a PEMFC stack system with a popular industrial benchmark model in Simulink developed at Michigan University. The optimization algorithms for quadratic and non-linear system without and with constraints are presented and discussed for application purpose in the NMPC. Simulation results confirm the effectiveness of the developed model in both smooth tracking performance and less optimization time used.

Conclusions and further work are given at the end of the thesis. Major contributions of the research have been outlined and achievements are checked against the objectives assigned. Further work is also suggested to extend the developed work to industrial applications in real-time simulation. This is to further examine the effectiveness of developed models. Extensive investigations are also recommended on the optimization problems to improve the existing algorithms.

ACKNOWLEDGEMENTS

First of all, I would like to express my respect and sincere gratitude to my Director of Studies, Prof. Dingli Yu, for his continuous support of my Ph.D study and related research, for his patience, motivation and immense knowledge. Without his invaluable guidance and comments, it would not be possible to conduct this research.

Besides, my deepest thanks go to my parents and family, for their great effort and sacrifices to offer me a great opportunity for pursuing a Ph.D in UK, and also for their emotional and financial support.

Last but not least, I would also like to thank anyone who has been involved during my times at Liverpool John Moores University providing any kind of help for the completion of this research.

Desmond Kean Siong Tok

Liverpool, UK

October 2016

CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

A. General Symbols

c	Centre vector
c_{active}	Number of active centres
C_{mc}	Centre with most contribution
e	Error vector
E	Error modelling matrix
J	Cost function for least square problems
k	Current sample instant
m	Number of system input
N	Number of collected sample data
n_h	Number of centre
n_p	Number of weights in Akaike's final prediction error
n_u	Maximum lag in the inputs
n_y	Maximum lag in the outputs
p	Number of system output
Q	Orthogonal matrix
R	Upper triangular matrix
r	Column of upper triangular matrix
u_s	Scaled system input
V	Loss function in Akaike's final prediction error
v_{cm}	Compressor motor voltage
W	Matrix of network weights
x	System input vector
y	System output vector
$\tilde{y}$	Model residual vector
y_s	Scaled system output
Y	Matrix of system output
$\hat{y}$	Network output vector
$\hat{Y}$	Matrix of network output
$\tilde{Y}$	Matrix of model residual

ε	Smooth factor
β	Weighting factor in Akaike's final prediction error
λ_{o2}	Oxygen excess ratio
$\hat{\lambda}_{o2}$	Prediction of oxygen excess ratio
ϕ	Activation function vector
Φ	Matrix of activation function outputs
σ	Width vector

B. Symbols for Model Predictive Control

H_p	Prediction Horizon
H_u	Control Horizon
H_w	Window Parameter for prediction horizon
Q	Penalty on error
R	Penalty on input moves
r	Reference point
V	Cost function
$\Delta\hat{u}$	Changes in input / input moves

C. Symbols for Optimization Algorithms

A	Active constraint
a	Artificial variable
B	Approximation of a hessian matrix
c	Coefficient
d	Descent direction
e	Equality constraints
f	Objective function
G	Matrix of inequality and equality constraints
g	Gradient of an objective function
H	Hessian matrix of an objective function
h	Inequality constraints
$\mathcal{L}$	Langrange function

P	Penalty function
q	Quadratic function
r_c	Reduced cost
s	Slack variable
x	Decision variable
x^*	Optimum solution
x_B	Basic variable
x_n	Non-basic variable
Z	Basis of active constraints
λ	Langrange multiplier for inequality constraints
μ	Langrange multiplier for equality constraints

D. Abbreviations

ARMAX	AutoRegressive Moving Average Model with eXogenous input model
ARX	AutoRegressive model with eXogenous input model
ASM	Active Set method
BFGS	Broyden-Fletcher-Goldfarb-Shannon method
BS	Bisection Search method
BOLS	Batch Orthogonal Least Squares algorithm
CSTR	Continuous-Stirred Tank Reactor
DFP	Davidon-Fletcher-Powell method
DMC	Dynamic Matrix Control
EHAC	Extended Horizon Adaptive Control
EPSC	Extended Predictive Self-Adaptive Control
FPE	Akaike’s Final Prediction Error
GGAP-RBF	Generalized Growing and Pruning Training Algorithm for RBF networks
GPC	Generalized Predictive Control
IDCOM	Identification Control
IMC	Internal Model Control
KKT	Karush-Kuhn-Tucker conditions
LMPC	Linear Model Predictive Control

LP	Linear Programming
LQG	Linear Quadratic Gaussian Controller
LQR	Linear Quadratic Regulator
LS	Least Square algorithm
MAE	Mean Absolute Error
M-RAN	Minimal Resource Allocating Network
MHE	Moving Horizon Estimate
MIMO	Multi-Input Multi-Output
MPC	Model Predictive Control
MPC-FS	f-step RBF Model-based Model Predictive Control
MPC-NARX	Non-linear ARX Model-Based Model Predictive Control
MPHC	Model Predictive Heuristic Control
MRT	Minimum Ratio Test
MSA	Multistep-Ahead Predictor
NARX	Non-linear ARX
NARX-RBF	NARX Model-Based RBF Network
NLP	Non-linear Programming
NMPC	Non-linear Model Predictive Control
OER	Oxygen Excess Ratio
OLS	Orthogonal Least Squares method
OSA	One-Step-Ahead Predictor
PEMFC	Proton Exchange Membrane Fuel Cell Stack System
PFC	Predictive Functional Control
PSC	p-Step Control Model
PSC-RBF	p-Step Control Model-Based RBF Network
QP	Quadratic Programming
RAN	Resource Allocating Network
RAS	Random Amplitude Signal
RBF	Radial Basis Function
RLS	Recursive Least Square training algorithm
ROLS	Recursive Orthogonal Least Square training algorithm
SD	Steepest Descent method
SISO	Single-Input Single-Output
SQP	Sequential Quadratic Programming

Chapter 1

INTRODUCTION

1.1 Background

With the advancement of technology driven by the high growth rate of consumerism, industrial expectations for process control have increased rapidly. For this reason, the monitoring and control of industrial processes have become more challenging. Processes are required to operate at a wider operating region to satisfy the operating criteria and economic objectives which often involves a large number of variable interactions. To integrate these demanding requirements, the complexity of processes has inevitably increased, which in turn has led to the development of advanced control approaches such as model predictive control (MPC).

As one of the optimal control theories, linear quadratic Gaussian controller (LQG) based on Kalman filter addressed linear control problems effectively. However, extensive practical concerns such as process constraints, non-linearity of a process and model uncertainty has restricted the application of LQG in the process industries (Qin and Badgwell, 2003). From an economic perspective, processes are often required to operate near to the limit of its constraints. The economic and psychological environment required for a successful control implementation is often not met in practice so that many constraints prevent the implementation of on-line control schemes on production plants (Richalet et al., 1978). Furthermore, industrial processes are typically of high complexity. The development of a process model may require vast variety of fundamental engineering disciplines such as chemistry, physics and other knowledge, which is not practically realistic. Another important aspect is that industrial processes are quite distinctive and have their own performance criteria and reliability requirements. The performance criteria includes safety regulations, physical limitations on equipment and others as stated in (Garcia et al., 1989).

MPC that has the ability to deal with the factors discussed above becomes one of the dominant control strategies. Process variable interactions and process constraints are the important features in MPC. Another advantage of MPC is the capability of its operating objective function in integrating the outlined practical performance criteria. Generally, MPC is divided into linear and non-linear types which is characterised by its internal model. Most industrial processes employ linear internal models where they are extended to cover different operating points of processes (Forbes et al., 2015). However, most industrial processes exhibit non-linear behaviours, which necessitate the employment of a non-linear internal model for a satisfying prediction and control performance, leading to the rise of non-linear MPC (NMPC). Obtaining a good prediction performance is essential in MPC.

System identification is introduced to deal with the problems in deriving a mathematical model of a complex process for the purpose of control system design. Although system identification techniques are developed independently, their efficient outcomes have led to the applications in MPC. Empirical models are useful as they are built from measured process data directly without requiring process knowledge. Thus, they are also known as data-driven models. Linear empirical models include autoregressive model with exogenous input model (ARX) and autoregressive moving average model with exogenous inputs model (ARMAX) (Ljung, 1999). These models are linear for unknown parameters, but are non-linear for variables. They can be easily identified using the linear identification methods. For non-linear cases, a non-linear ARMAX (NARMAX) model is used. One of the disadvantages of ARX, ARMAX and NARMAX models are they are constrained within the scope of data that used to build them. For time-varying processes, these models are required to be updated using by the latest process data.

Adaptive modelling, regarded as one of the advanced techniques in system identification, has been studied intensively in the past few decades. The implementation of a radial basis function (RBF) network as a function estimator for non-linear functions was presented by Broomhead and Lowe (1988). They proposed a linear function space which depends on the positions of the generally non-linear known data points according to an arbitrary distance measure. Following their work, the popularity of RBF networks in adaptive modelling rises rapidly and has been intensively studied for the adaptation of its structure and parameters. The dynamic structures of adaptive RBF networks are able to accommodate different operating points of processes and their parameters estimation can be updated in on-line mode according to the parameters of processes (Yu and Yu, 2007, Qiao and Han, 2012).

The on-line execution of MPC has become more effective with its implementation based on an adaptive RBF network (Yu et al., 2006, Han et al., 2013). The research interest in this work focuses on process modelling using adaptive RBF networks, RBF model-based MPC and its application including a discussion of related optimization algorithms.

1.2 Research Motivations

There are three motivations. The first motivation is on the adaptive RBF networks. Secondly, the motivation in studying the mathematical optimization in relation to MPC is described. Lastly, the desire to improve the RBF model-based MPC is included.

As outlined, the emergence of time-varying processes inspires adaptive modelling techniques. An adaptive RBF network is able to model non-linear time-varying processes without reconstructing its structure. The centre adding strategy plays a vital role in producing informative centres that adapts to the operating regions. The challenge is to produce new centres that can adapt to new process dynamics immediately and effectively. Existing methods (Yu et al., 2004, Yu and Yu, 2007, Han et al., 2011) produce new centres without assessing the contribution of centres, resulting in a slow recovery speed after the migration of operating points. The first part in the thesis focuses on developing an effective adaptive RBF network with a new centre adding strategy. Recursive orthogonal least squares (ROLS) algorithm is employed to train the RBF network. The main reason of selecting ROLS training algorithm is that it enables the assessment of the contribution of every centre, which formulates the forward and backward centre selection methods. In addition, the simplicity of ROLS training algorithm enables the training of network structure and parameters simultaneously.

MPC is popular due to its straightforward implementation framework which formulates a control problem into an operating objective function by taking process constraints into consideration. Nevertheless, optimizing the formulated objective function requires Non-linear Programming (NLP) algorithms when an NMPC is considered. Nowadays, optimization techniques are easily available through developed software tools such as Matlab that can be implemented in a straightforward manner. Many work have, instead, emphasized on improving process models such the development of RBF networks. But, the understanding and selection of an optimization algorithm are equally important. Mathematical optimization is a wide topic area that includes linear and non-linear problems. It requires a great effort to explore relevant optimization techniques for MPC. As part of the research, the experience of author in studying optimization algorithms is used to construct a foundation in this discipline, particularly for the techniques in relation to the MPC.

One of the advantages of a RBF network model-based on a NARX model (NARX-RBF) is that it can be trained as an independent model to make multi-step ahead predictions for MPC. The use of a RBF network as an internal model in MPC leads to a NMPC, which increases the computational efforts in solving the formulated optimization problem. This has restricted the use of RBF models in MPC as time constraint is one of main factors during the on-line execution for fast dynamic processes. The computation time required to solve an optimization problem at each sampling interval may be insufficient if a small sampling period is selected. In this work, a RBF network is made factorable by modifying the elements in the NARX model. The factorized RBF model enables the derivation of an explicit objective function for MPC with the aim to reduce the computation burden in executing the optimization algorithm. As one of the challenges, the factorization has to be done without compromising the prediction performance and model compactness of the RBF network. This is because model compactness has the direct impact on the computational load. An existing factorized RBF network (Bhartiya and Whiteley, 2001) is capable of achieving satisfactory prediction performance but the model compactness is compromised.

1.3 Aims and Objectives

Based on the motivations above, the aims and objectives are summarized as:

Aims:

1. To develop an effective adaptation algorithm for RBF networks.
2. To develop a more efficient RBF network model for MPC.

Objectives:

The aims above will be accomplished by fulfilling the following objectives:

1. Develop a structure adaptation algorithm for RBF networks
2. Develop an adaptive RBF network model for time-varying processes
3. Develop a f-step prediction method for RBF network models
4. Develop a factorization algorithm for fast model prediction in RBF network models
5. Develop a model predictive control based on f-step model-based RBF network (f-step RBF)
6. Apply the f-step RBF network model-based MPC to a PEMFC stack system

1.4 Contribution to Knowledge

The main contributions of the research:

1. A new adaptive RBF network is developed using ROLS training algorithm (Tok et al., 2015). The developed network is constructed based on three main steps which are adding, pruning and grouping of the centres. An effective centre adding strategy is proposed to adapt the network structure to the operating

regions of the process to be modelled. In order to maintain the compactness of the network model, a grouping algorithm is introduced to select significant centres to form a final network model. The experiment result confirms that the developed technique has advantages in term of overall prediction performances and achieves a faster recovery speed compared to other methods.

2. A new factorized f-step model-based RBF network is developed and implemented in MPC (Gu et al., 2016). The developed network is able to predict future system response without requiring the unknown system response, which allows the RBF network to be factorized for the application in MPC. The outcomes demonstrate that the developed network is more compact than existing methods while maintaining the prediction accuracy. Furthermore, recorded results show that the developed network model-based MPC achieves a reduction in the computational load in solving the on-line optimization problem.

3. The f-step RBF network is applied in MPC to control a PEMFC stack system. First comparison results indicate the advantages of the developed network in term of modelling performance and model compactness. Secondly, the proposed model-based MPC achieves a satisfactory control performance while requiring less computation load. Both modelling and control outcomes demonstrate that neural network based MPC is a potential control strategy for the PEMFC stack system. Another key point is this application has validated the potential of the developed factorized RBF network.

1.5 Thesis Overview

This thesis consists of eight chapters and it is organized as follows. Chapter 1 introduces the background and covers the research motivations. Aims and objectives are clearly defined. The contributions to knowledge are highlighted.

Chapter 2 presents the literature review. The key aspects of a RBF network are discussed. The distinctions between fixed structure and adaptive structure RBF network models are examined. The recent developments for adaptive structure RBF networks are reviewed. This includes adaptive approaches that based on orthogonal decomposition methods such as batch orthogonal least squares and recursive orthogonal least squares algorithms which form the fundamental principles for the developed approach in the research. In next section, the historical background of MPC is firstly reported, followed by the studies on linear and non-linear models based MPC. In particular, the review of RBF network models based MPC is imperative in the development of the factorized RBF network approach. Finally, a review on the properties of MPC is also included.

Chapter 3 describes the methodology of the research. The basic concept of a RBF network and employed training algorithms are presented. An overview of a RBF network is given, including the structure, the non-linear prediction model and the activation function. The key training algorithms such as recursive K-means algorithm and P-nearest neighbour method are introduced which are used to decide network structure. Then, it is followed by the recursive least square training algorithm and ROLS training algorithm for network parameters estimation and structure adaptation. This chapter concludes with dynamic process modelling techniques, namely prediction models, data collection, a data scaling method and an error measurement method.

In Chapter 4, an adaptive structure RBF network model is proposed to model non-linear processes with operating region migration (Tok et al., 2015). The ROLS training algorithm is adopted to select new centres on-line, as well as to train the network weights. Based on the *R* matrix in the orthogonal decomposition, an initial centre bank is formed and updated in each sample period. A new learning strategy is proposed to gain information from the new data for network structure adaptation. A centre grouping algorithm is also developed to divide the centres into active and non-active groups, so that a structure with a smaller size is maintained in the final network model. The proposed RBF model is evaluated and compared to the three adaptive structure RBF networks by modelling a non-linear time-varying numerical example. Simulation results demonstrate that the proposed algorithm has several advantages in term of the adaptive tracking ability and a better recovery speed over the existing methods (Yu et al., 2004, Yu and Yu, 2007, Han et al., 2011) during the migration of process's operating points.

In the first part of Chapter 5, the concept of receding horizon approach in MPC is introduced and the following section describes the features of MPC. The effects of tuning parameters and their relationships are discussed. The following part presents the optimization algorithms. The order of this part is constructed based on author's experience. Linear Programming is firstly considered with a constrained algorithm, Simplex method. Next, the Lagrange method that employed for a general problem is included as it forms the basis for many optimization algorithms. It is followed by Quadratic Programming and Active Set method. Non-linear Linear Programming is outlined including the background of unconstrained methods and a constrained algorithm, Sequential Quadratic Programming. Ultimately, the selections of optimization algorithms in relation to MPC are discussed.

In Chapter 6, a new factorized f-step RBF network model for MPC is proposed. The strategy is to develop a f-step predictor for non-linear dynamic systems and implement it with a RBF network. In contrast to the popular NARX model based RBF network model, the developed f-step RBF network model is capable of making a designated sequence of future output prediction without requiring the unknown future process measurements. Furthermore, the developed f-step RBF network model is factorized into two parts, with one part including past plant input/output and the other part including the future input/output. When this model is used as the internal model in the MPC, the factorization enables an explicit objective function for the on-line optimization in the MPC. Thus, the computing load in solving the optimization problem is greatly reduced. The developed model is used in MPC and applied to a continuous-stirred tank reactor. The simulation results are compared with that of MPCs with other two models. The comparison confirms that the developed model makes more accurate predictions so that the MPC performance is better, it also uses much less computing time than the other two models based MPC.

In Chapter 7, a f-step RBF network model is developed for oxygen excess ratio prediction of the PEMFC stack system. Then, the MPC based on the factorized f-step RBF network model is developed and applied to control the oxygen excess ratio for PEMFCs. Compared with the widely used NARX-RBF model, the f-step RBF network model can predict more precisely for long-term prediction. Furthermore, the factorization algorithm developed in the f-step RBF network model significantly reduces the computing load when the model is used in the MPC for long-term prediction, and consequently reduces time for on-line optimization in the MPC. The developed scheme is successfully applied to a non-linear simulation of a PEMFC stack system to control the oxygen excess ratio, and the aforementioned features are demonstrated by comparing the performance with the control using the NARX-RBF model.

Finally, Chapter 8 concludes the contributions of this research including the achieved aims and objectives. The conclusions are generally divided into two parts, summarising the development of an adaptive RBF network, and f-step RBF network based MPC, respectively. The recommended future work is given in the final part of this chapter.

Chapter 2

LITERATURE REVIEW

This chapter is divided into two main sections. The first section covers the background of a RBF network including the on-line and offline parameters training approaches. The discussion is followed on network structures for both fixed and adaptive types. Existing methods for adaptive structure RBF networks are reviewed where the focus is on orthogonal least square training algorithm in both offline and on-line modes.

In the second part, MPC is discussed, with an overview and historical background at times. Next section discusses the internal models in linear and non-linear forms. Lastly, a brief overview of the properties of model predictive control is included.

2.1 RBF Network Models

Artificial neural networks (ANN) have been found in many applications such as non-linear control, signal processing, system diagnosis and faults detection due to their good approximation accuracy (Xie et al., 2011). There are two popular ANNs, which are RBF and multilayer perceptron (MLP) networks. Early development is focussed on MLP networks but RBF networks have gained more attentions due to reported superior performances (Finan et al., 1996).

Both RBF and MLP networks have different properties despite having similar network topologies. The MLP structures are global approximators whereas the RBF networks are local ones. MLP networks tend to have more hidden units, resulting in a more complex architecture. Therefore, the training of a MLP network can be computationally demanding (Dawson et al., 2002). With a simple architecture, RBF networks are simpler to train (Hao et al., 2011). Many applications compare the performance between MLP and RBF networks. In flood forecasting, Jayawardena et al. (1997) showed that RBF networks used less time and effort in model development while producing comparable prediction accuracy compared to that of MLP networks. A survey paper by Kumar and Yadav (2011) compared different MLP and RBF network techniques and concluded that RBF networks provided more accurate predictions and excellent generalization. An additional point is that the convergence to optimum values is guaranteed for a RBF network as it is a linear in the parameters model.

RBF network models have been popular due to their on-line learning abilities and simple architecture (Hao et al., 2011). The application of RBF networks to approximate non-linear functions was first studied by Broomhead and Lowe (1988) and the generalisation ability was investigated in (Lowe, 1989). Since then, the RBF networks have been studied intensively (Zhao and Huang, 2007, Herrera et al., 2011, Wang et al., 2012) and widely used in non-linear systems modelling and control (Seshagiri and Khalil, 2000, Li et al., 2001, Qin et al., 2015, Zhao et al., 2015, Wu et al., 2016).

In a RBF network, the approximation is done by mapping a non-linear function into a linear combination of weighted outputs using an activation function. There are a few of activation functions available for the RBF network. The RBF network centres have greater impacts on the network performance compared to the activations functions. Therefore, the choice of activation functions is often neglected as stated in (Chen et al., 1991).

In general, there are two phases in constructing a RBF network: forming a network structure and the estimation of network parameters (weights). The construction of a network structure involves in deciding an optimal number of centres and their width of radius. The network parameters can be trained in two modes: 1) an off-line mode using the batch least squares training algorithm, 2) an on-line mode using the recursive least squares training algorithm. In the off-line training mode, the network parameters are pre-trained using a set of collected data points. This means that the network has fixed parameters. Meanwhile, in the on-line training mode, the network parameters are updated with the arrival of new data points at each sampling interval. The advantage of this method is the network parameters are adapted recursively to the process dynamics, improving the network performance, in particular for processes with high parameter uncertainties as demonstrated in (Wang et al., 2006). However, the on-line parameter training mode only tends to be effective within the scope of training data points. For processes with operating regions migration, adaptive structure RBF networks are proposed which is discussed in Section 2.2.

In early implementations, RBF networks used all data points as centres, leading to a large network structure which required a huge memory space and also encouraged numerical ill-conditions (Mashor, 2000). This is practically unrealistic for industrial applications as a big amount of data points is usually employed. To overcome this drawback, Poggio and Girosi (1990) proposed a restricted network structure which used a finite amount of centres.

In general, the structure of a RBF network is classified into two categories: fixed-structure and adaptive structure. For a fixed-structure RBF network, the number and location of centres are static during the modelling process and the model parameters (weights) may be adapted. In early developments, input data vectors from the training data were arbitrarily chosen as network centres (S. Elanayar and Shin, 1994). Another approach was to employ random vectors from the input space (Kaminski and Strumillo, 1997). The other method is the position of centres is decided using a clustering method such as the popular K-means clustering algorithm (Moody and Darken, 1989, Mashor, 2000). As an statistical approach, the disadvantages of K-means clustering algorithm are its lacklustre learning ability in the training data and the effect of data noise (Cui et al., 2016). However, it is still effective in some cases if sufficient centres are used. One major drawback of the fixed-structure RBF network is the selection of centres are strictly bounded by the training data (Zhang and Li, 1996).

The performance of a RBF network is heavily dependent on its structure and it is imperative to optimize the network structure to achieve a satisfactory performance, especially in modelling a highly time-varying process. A satisfactory network performance can be achieved provided a sufficient number of centres is employed and there is no prior knowledge to find an exact number of centres that needed (Gomm and

Yu, 2000). The number of centre is often predefined by the user intuitively and empirically. An inappropriate number of centres may result in poor global generalization of the RBF network (Sun et al., 2009). A short amount of centres can lead the degradation in the prediction accuracy. For this reason, an unnecessary large RBF network is usually used, which causes numerical ill-conditioning in the training of the network and the worsen generalization of the trained model (Chen et al., 1992b, Zhang and Li, 1996).

2.2 RBF Network Structure Adaptation

To overcome the drawbacks of fixed-structure networks as discussed in the previous section, adaptive RBF network structures are proposed. An adaptive structure RBF network has the number and location of its hidden layer neurons adapted to better fit the dynamics of the process to be modelled, in addition to the adaptation of network parameters (Chen et al., 1992a).

In the past decades, the adaptations of RBF network structures have been intensively investigated. First of all, the development of a dynamic RBF network structure was initiated by Platt (1991) who introduced an algorithm called resource allocating network (RAN). For an RAN, the hidden units are gradually inserted into the hidden layer based on the novelty of new data. In a latter attempt, Karayiannis and Mi (1997) developed a framework for growing RBF networks which merged supervised and unsupervised learning with network growth techniques. They proposed that the structure of network could be gradually constructed by splitting and increasing the prototypes which represented the network centres. However, the insignificant hidden neurons in (Platt, 1991, Karayiannis and Mi, 1997) were not pruned which led to a final network with a huge structure.

To solve the oversized problem, Lu et al. (1997) proposed a sequential learning scheme for function approximation using a minimal RBF network which was referred as minimal RAN (M-RAN). Their pruning strategy was to prune the hidden units that had insignificant contributions to the network performance. The performance of M-RAN was evaluated in (Lu et al., 1998). However, the optimal network structure achieved in

(Lu et al., 1998, Lu et al., 1997) is only for a certain data set, while the performance would be degraded if it is used to predict future behaviour in other regions. Huang et al. (2005) developed a generalized growing and pruning training algorithm for RBF networks (GGAP-RBF). Their algorithm constructed the network structure by evaluating the hidden neuron's statistical contribution. Then, Bortman and Aladjem (2009) modified the GGAP-RBF and showed that modified GGAP-RBF outperformed the original GGAP-RBF. However, in order to implement both approaches (Huang et al., 2005, Bortman and Aladjem, 2009), full information of the training data is required which is not practically realistic.

In recent years, a few methods have been proposed for self-organizing RBF networks (Qiao and Han, 2012, Han et al., 2011). Although it was claimed that these methods (Han et al., 2011, Qiao and Han, 2012) outperformed M-RAN (Lu et al., 1997) and GGAP-RBF (Huang et al., 2005), the convergence of their algorithms needed to be investigated carefully for successful applications, which complicated the entire training algorithms. Moreover, there are many unknown parameters in (Han et al., 2011, Qiao and Han, 2012) which need preliminary runs to find optimal values for the parameters before the adaptation of network take places.

2.2.1 Orthogonal Least Squares Algorithm Based Model Adaptation

Orthogonal decomposition is a numerically stable method for solving least squares problems. The application of the orthogonal least squares algorithm (OLS) in non-linear system identification was firstly analysed in (Chen et al., 1989). Chen et al. (1989) introduced an efficient subset selection method to select significant terms for NARMAX polynomial models using the OLS algorithm. With this in mind, not only does the OLS algorithm is simple to execute, its convergence is also guaranteed with its least squares algorithm. A great number of research references for the applications of OLS algorithm are available in (Drioli and Rocchesso, 2003), (Sheng et al., 2009), (Lin et al., 2009), (Chang, 2013) and (Mehta et al., 2016).

2.2.1.1 Batch Orthogonal Least Squares Algorithm based Methods

An orthogonal estimation method was firstly proposed by Korenberg et al. (1988). To determine a RBF network structure, Chen et al. (1991) proposed a forward regression learning approach based on the batch orthogonal least squares algorithm (BOLS). In their approach, the OLS algorithm was employed to determine an appropriate set of centres from a large set of candidate centres in a forward regression procedure. Chen et al. (1991) exploited the decomposition transformation to investigate the individual contribution of each centre to the desired network output. In their method, using an error reduction ratio as the termination criterion, the centre was chosen systematically, one by one, until an adequate RBF network structure was achieved. Chen et al. (1992b) further extended this method (Chen and Billings, 1992, Chen et al., 1991) to train a multi-input multi-output (MIMO) RBF network. The application results in (Chen et al., 1991) and (Chen et al., 1992b) showed that satisfactory performances were achieved. In a further attempt, Chng et al. (1996) extended the work in (Chen et al., 1991) and (Chen and Billings, 1992) by introducing a local adaptation process for a RBF network structure. In the work of (Chng et al., 1996), the subset models with higher accuracy were achieved compared to (Chen et al., 1991, Chen and Billings, 1992). Zhou et al. (2010) suggested to incorporate Bayesian information criteria with a forward selection procedure based on OLS algorithm to decide the number of centres. The advantage in (Chen et al., 1991, Chen and Billings, 1992, Chng et al., 1996, Zhou et al., 2010) is the network structure and parameters were decided simultaneously by evaluating the contributions of centres to network performance. However, one major drawback is the optimization of network parameters is of off-line training mode as their methods (Chen et al., 1991, Chen and Billings, 1992, Chng et al., 1996) are based on the BOLS algorithm. This means that the process parameters are not adapted.

2.2.1.2 Recursive Orthogonal Least Squares Algorithm based Methods

In order to overcome the drawbacks of BOLS as mentioned above, Bobrow and Murray (1993) had developed a recursive algorithm based on OLS (ROLS) to identify the network parameters in an on-line mode. However, the developed algorithm in (Bobrow and Murray, 1993) is only applicable for a single-input, single-output (SISO) form. In a following effort, Yu et al. (1997) extended the work in (Bobrow and Murray, 1993) to a multi-input, multi-output form and employed it in an on-line application to train the weights of RBF networks. Yu et al. (1997) showed that the ROLS training algorithm was capable of maintaining the same accuracy of the RBF network model as the off-line training while requiring less computation. With the developed ROLS training algorithm, Yu (2004) improved the network performance by adopting a localized forgetting algorithm. Gomm and Yu (2000) developed a forward and a backward centre selection algorithms using ROLS training algorithm. For the backward selection algorithm, the structure of network is simplified by removing the centres which had smallest contribution to the network performance. On the other hand, for the forward selection algorithm the technique is to build a network by adding centres which will maximally enhance the network performance. The termination of both backward and forward centre selecting procedures is decided using a final prediction error criterion which gauges the trade-off between network performance and model complexity. Their method (Gomm and Yu, 2000) resulted in an acceptable level of efficiency and accuracy with a smaller network's size. Recently, Zhang et al. (2015) proposed a computationally efficient two-stage OLS method to construct the network structure. However, the developed RBF network models in (Gomm and Yu, 2000, Yu, 2004, Chen, 2009, Zhang et al., 2015) were not 'fully' adaptive as the centres can only be selected from a pre-specified candidate centre set.

The use of the backward centre selection method was extended in (Yu et al., 2004) to develop an adaptive RBF network model but the performance was not satisfactory due to the lack of efficiency in the selection of centres. In further work, Yu and Yu (2007) proposed an adaptive algorithm that incorporated the pruning strategy in (Gomm and Yu, 2000) to adapt a RBF network model using the ROLS training algorithm. The adding and pruning of centres was based on the error index between the desired and measured modelling performances. New data was added as new centres if the desired modelling performance was not achieved. Results showed that a compact RBF network was achieved while the desired modelling performance was maintained. However, in this method the added new centres did not play a role immediately as the performance was degraded for a few sample periods before the positive role was observed during the migration of the process's operating point.

In this work, the adaptive structure RBF network model based on ROLS training algorithm proposed by the author comprises an effective centre adding algorithm using a new learning strategy (See Chapter 4 and (Tok et al., 2015)). The RBF network model is updated in on-line mode according to the information of new process data at each sample period. The improvements in the adaptive tracking ability and recovery speed of this proposed approach over the existing methods (Yu et al., 2004, Yu and Yu, 2007, Han et al., 2011) are demonstrated.

2.3 Model Predictive Control

MPC is a popular advanced control method that has many successful industry applications (Forbes et al., 2015). Due to its popularity and promising long term success, the development and achievements of existing MPC methods have been consistently reviewed (Garcia et al., 1989, Morari and Lee, 1999, Qin and Badgwell, 2003, Mayne, 2014, Forbes et al., 2015) since its introduction as a software called Identification Control (IDCOM) by Richalet et al. (1978). The simple concept of MPC is one of the reasons that it gained a wider acceptance as the early popular modern control methodology (Qin and Badgwell, 2003). Another key reason is the economic considerations, processes are usually required to operate at boundary permitted state within the operating points respecting all the constraints; this is the situation where the MPC can address effectively (Mayne et al., 2000).

The general concept underlying the MPC is to solve an optimization problem to obtain an optimal control variable over a finite horizon at each sample time. The optimization problem is formulated based on a defined control objective respecting the desired set-points, future predictions, input control moves and constraints for inputs and outputs. The computed optimal control variable is then selected to apply to the process to obtain updated process conditions, resulting as a feedback control law. The procedure is repeatedly executed over a shifted horizon. Therefore, the control strategy is regarded as a receding horizon approach (see Chapter 5 for more details). In short, the feedback of MPC is formed by solving a formulated optimization problem using the process current states. Depending on the control problems, the general characteristic of the optimization problems can be categorized by its linearity, weight and constraint (Li et al., 1989).

2.3.1 The Relation to Other Control Methods

MPC serves as an effective replacement for traditional optimal control methodologies due to its ability in computing on-line control solutions, which is advantageous in the case where initial states of processes cannot be obtained easily (Mayne, 2014). For this reason, MPC allows processes to deviate from its actual states. However, it is important to point out that the feedback of other optimal control methods such the linear quadratic regulator (LQR) can be achieved easily by computing simple matrix multiplications, whilst, MPC is required to solve an optimization problem to achieve the feedback. An early comprehensive review by Garcia et al. (1989) included the MPC design techniques and its key relationship to the LQR. Different from the MPC, LQR is executed over a fixed window horizon (Wang, 2009). MPC resembles the LQR problem in linear cases under the circumstances of infinite control and prediction horizons without considering the constraints as discussed in (Morari and Lee, 1999).

One of the useful features in the MPC approach over conventional model-based control methods such the internal model control (IMC) and Smith predictor is it can be applied to MIMO processes by taking constraints into consideration in a systematic manner. The close relationship between the configurations of MPC, IMC and Smith predictor was discussed in (Garcia et al., 1989). The handling of constraints is of great importance as relevant performance criteria of industry processes can be directly reflected during the control execution. In practical applications, most actuators have their own nature limitation, forming as one of the constraints for inputs. On the other hand, it is important for processes to respect the safety constraints, leading to the constraints on outputs. Another key advantage of MPC is the dependence of the current control signal on future predictions is being considered (Rossiter, 2003), using the optimal control variables to observe the changes in future predictions. This helps reduce the difficulties in executing the future control signals. Despite the MPC is easy to understand and being a straightforward formulation, the optimization problem may become complex depending on the nature of internal models and defined constraints.

2.3.2 Historical Background

The MPC has a great background which its emergence was largely related to industrial applications before being investigated extensively in academia. It was reported in (Garcia et al., 1989, Morari and Lee, 1999) that the ideas of MPC could go back as far to 1960s in relation to Linear Programming methodologies (LP) which are the fundamental concept of MPC with respective publications:

- Zadeh and Whalen (1962) discovered the connections between minimum time problem and LP, proposing that it can be formulated as a series of LP problems.

- Propoi (1963) suggested that a LP problem to be solved at each sampling period which founded the moving horizon idea and this idea was named as open look optimal feedback by (Dreyfus, 1964).

- Chang and Seborg (1983) developed a feedback control strategy which was to solve on-line LP problems considering the inequality constraints.

As suggested by many researchers, MPC started to rise to prominence with its successful industrial applications. The tremendous impact of MPC on industrial processes, particularly petrochemical processes, was verified by Qin and Badgwell (2003) with over 4500 reported applications by mid of 1999. The first recognised MPC design was called model predictive heuristic control (MPHC) introduced by Richalet et al. (1978). In a later attempt, an improved version, IDCOM-M, presented by Grosdidier et al. (1988) had also attracted great attentions. Another MPC design, dynamic matrix control (DMC), was successfully applied in petrochemical industries (Prett and Gillette, 1980) by engineers from Shell Oil Co (Cutler and Ramaker, 1979), saving the respective industry an enormous cost. A following extended work of DMC developed by Morshedi et al. (1985), named as linear DMC, which merged the LP techniques with DMC. Both MPHC and DMC were regarded as first generation MPC methodology. The main difference between MPHC and DMC is on their models; the former employed an impulse response model while a step response model was used in the latter (Holkar and Waghmare, 2010). The benefit of using these models in MPHC and DMC is both models represent the processes with stable dynamic response, providing better prediction accuracy. However, unstable processes are not encouraged in MPHC and DMC.

As suggested by Morari and Lee (1999), the increasing interest in MPC was also due to the development of generalized predictive control (GPC), which is an adaptive MPC method developed by Clarke et al. (1987). The features between different adaptive control designs in the MPCs were reviewed in (Holkar and Waghmare, 2010), which included predictive functional control (PFC) (Richalet et al., 1995), extended predictive self-adaptive control (EPSC) (DeKeyser, 1988) and extended horizon adaptive control (EHAC) (Ydstie et al., 1988). The models employed in GPC, PFC, EPSC and EHAC reflects the dissimilarities between each method. As discussed in (Holkar and Waghmare, 2010), the emerging of these techniques is due to different control objectives and motivations, which means that there is no technique inherently superior than another. Although every MPC technique has distinctive properties in relation to their models, their control designs are based on a general framework such as the employment of a process model and the optimization of an objective function (Rossiter, 2003).

2.3.3 Linear Model MPC

The selection of an internal model for MPC is an important stage as a good prediction accuracy of the internal model is a fundamental requirement for a good control performance. Demonstrably, the demand of control quality is highly dependent on the prediction performance of the internal model (Rossiter, 2003). One of the advantages of MPC is its flexibility in selecting the internal model. Despite having stating this, different internal models lead to different optimization problems, namely Linear Programming, Quadratic Programming and Non-linear Programming. The use of linear and non-linear internal models characterises the MPC as linear MPC (LMPC) and non-linear MPC (NMPC), respectively.

For LMPC, the internal model can be a simple transfer function but the expansion to multivariable case for MPC is non-trivial despite having similar advantageous properties of black-box models (Rossiter, 2003). As discussed early, both FIR models, impulse response and step response models were regarded as the common models used in industrial. FIR models are easy to comprehend and they have high abilities in making accurate predictions but their applications are confined to stable processes. The other popular approach is to use a linear state space representation model as discussed in (Li et al., 1989). Kalpesh (2004) expressed the state space based MPC in factorized form and proposed to solve the resulting constrained quadratic optimization problem using Active Set method. The advantages of using a state space model is it can be expanded to multivariable case easily. Additionally, there are a lot of supportive theoretical works such as observers can be incorporated in the state space model as discussed in (Rossiter, 2003). The disadvantage of FIR models and state-space representation models is process parameters are needed.

Alternatively, linear parametric models are employed for LMPC such as the ARX model (Huusom et al., 2010, Tajjudin et al., 2010, Kon and Yamashita, 2010, Muddu et al., 2010) and the ARMAX model. The advantage of the ARX model is the formulated optimization problem is convex, which simplifies the optimizing process as discussed in (Bemporad et al., 2002). Although the ARMAX model has a more flexible disturbance handling ability compared to the ARX model, the formulated optimization problem in MPC becomes non-convex (Huusom et al., 2012). On top of that, the ARX model is comparably less difficult in identifying higher order systems. For this reason, the ARX model is widely used in MPC design. However, the incompetence of linear models in modelling non-linear processes has always been a hindrance to LMPC, encouraging the employment of non-linear models as an alternative approach.

2.3.4 Non-linear Model MPC

For non-linear cases, it was understood that the popular feedback linearization had been an effective method for non-linear control but their complicated design framework and its lack of ability in handling contraints were the drawbacks that lead to the NMPC (Johansen, 2011). Many real application processes are non-linear where a large mismatch between a linear internal model and a real process is often encountered, resulting in a poor control performance eventually. Beside from this, additional factors such as the high demand of product quality, productivity and other external economic factors require processes to extend the operating regions. Therefore, linear internal models are not capable to capture the process dynamics adequately (Allgower et al., 2004). Under these circumstances, a non-linear internal model provides a better model fit to the processes.

A distinguish factor between LMPC and NPMPC lies in their computation aspects as the challenge for NMPC is to solve non-linear optimization problems using Non-linear Programming algorithms, which requires expensive computational requirements. In other words, the implementation of NMPC is non-trivial as the LMPC that the optimization problem is normally a LP problem. This has remained as a hindrance to the practical application of NMPC. Moreover, the difficulties in solving an NMPC over an infinite horizon have leaded to the interest in implementing the NMPC with a finite prediction horizon iteratively. These disadvantages have restricted the application of NMPC to relatively slow processes, which only allows a considerable huge sampling period that provides sufficient time to solve the optimization problems. However, the improvement of hardware such as the fast computer has widened the applications of NMPC such in medical field (Haverbeke et al., 2008, Hafidi et al., 2008), aviation industry (Chemori and Marchand, 2008), ore milling industry (Coetzee et al., 2010), robotic application (Wilson et al., 2016) and the trending electric vehicle control (Yuan et al., 2016).

One of the advantages of NMPC is a first principle model is allowed in direct applications (Rodríguez and Pérez, 2005). However, using the first principle model can be advantageous and disadvantageous at the same time. On the positive side, the closed-loop performance of NMPC can be greatly improved (Allgower et al., 2004). On the negative side, as for complex processes, their first principle models are usually of high complexity and may cause numerical ill-condition in the on-line framework of MPC. Moreover, the development of a first principle model is non-trivial and it can be a very time consuming procedure as it may involve a wide area of engineering disciplines that governing the processes.

Similar to the LMPC, a state space model is one of popular models but it is of non-linear version. Its formulation and applications were described in (Guerreiro et al., 2008, Mills et al., 2009). Another model such as an extended state-space model is also used to accommodate non-linear dynamics processes by employing a fuzzified model to estimate the parameters of state-space models as suggested in (Blanco et al., 2008). However, the prior knowledge of processes is required in order to construct the state-space models. It is understood that the states that are not available have to be estimated, which can be achieved using the extended Kalman filter. Another methodology is to employ a optimization-based moving horizon estimate (MHE) model (Haverbeke et al., 2008), which is proved to be a more effective technique as evaluated by Haseltine and Rawlings (2005). The estimation of state is done within a moving horizon window. Different from the LMPC, it is difficult to make comparisons technically between the non-linear internal models as the selections of these models mentioned above involve an extensive theoretical background from the processes. The efficacy of models hugely depends on the knowledge of the processes and often, the derivation of these models may require huge efforts if the processes are of high complexity. Given these conditions, it restricts the applications of these respective models in NMPC. For these reasons, an input-output model is preferable.

The improvement of non-linear system identification techniques has played an important role in NMPC. One of the popular non-linear empirical models is the NARX model. The application of a NARX model based MPC approach in the fermentation process by Mohd and Aziz (2015) had proved to have an obvious advantage in the prediction performance and simulation results demonstrated that it had a better disturbance handling with a faster response time compared to linear model-based approaches. There are a few classical non-linear identification methods derived from the Volterra series such as Hammerstein, Wiener and NDE models. The comparison results in (Nelles and Isermann, 1995) showed that the RBF network performed better in the absence of the process structure compared to the Volterra series models. The notable disadvantage of the Volterra model based MPC that pointed out by Shi et al. (2015) is a high order Volterra model, which gives a satisfying modelling performance, requires high computing loads.

For the reasons mentioned above, a RBF network based on a NARX model is renowned for its promising abilities in modelling non-linear dynamic systems (Diaconescu, 2008). The compact RBF network model structure is vital for MPC. A RBF network can be trained as a dependent model or an independent model (see

Chapter 3 for more details). The independent model plays a significant role in the MPC framework in which a multi-step future prediction is required instead of the one-step ahead prediction by the dependent model. The non-linear function in the NARX model that approximated by a regression model can also be formulated in a multi-step prediction form as illustrated in (Zulkeflee et al., 2011). However, the NARX-RBF model has superior prediction and adaptive abilities compared to that of the NARX model. Therefore, a NARX-RBF model is often used as an internal model in MPC (RBF-MPC) and the applications of this approach can be found in (Wang et al., 2006, Yu et al., 2006, Wang et al., 2007, Samek and Dostal, 2009). However, there are several drawbacks in the NARX-RBF model. The first drawback lies in the lack of efficiency in long range predictions due to its accumulated errors in each prediction step (Su and McAvoy, 1997, Bhartiya and Whiteley, 2001). Another disadvantage is that its future output predictions are made depending on future unknown process measurements. This means that the network cannot be factorized according to the past and future information, and the objective function is needed to be computed numerically. To overcome these issues, Bhartiya and Whiteley (2001) developed a factorable *p*-Step control model-based RBF network which produces efficient long predictions and applied it in MPC. Their results showed that the model performed better than the cascaded 1-step ahead prediction. However, one of the major drawbacks in (Bhartiya and Whiteley, 2001) is its unrealistic huge network structure which increases the model complexity and the computation times in predictions and solving the optimization problem in MPC.

2.3.5 Properties of MPC

As outlined in previous sections, MPC is regarded as a closed-loop control system where its feedback is achieved by solving an optimization problem. An ideal case would be solving the optimization problem in an infinite horizon considering the condition of a perfect modelling performance with no disturbance, which yields an open-loop problem (Allgower et al., 2004). Despite having stating this, a finite prediction and control horizon is required considering the computing burden to solve the optimization problem. In the meantime, to assume a perfect model matching scenario to exist is unrealistically optimistic, raising the issues of stability.

It is worth to study the background, ideas and approaches that have been done on the properties of MPC to develop fundamental knowledge (excluding the technical details) albeit it is beyond the scope of this work. In early developments, there were limited related studies in the properties of MPC as the advent of MPC was initiated with the successful industrial applications as discussed previously. However, extensive literatures on this aspect ensued after assorted MPC were proposed. The literature in this section does not intend to cover the complete properties of MPC. Instead, a more important aspect, namely the stability issue is considered.

Understandably, the investigation of stability of MPC was not considered theoretically in early process applications. For most of the applications, stability can be achieved provided the employed prediction horizon is sufficiently long as claimed in (Mayne, 2014). For this reason, the effect of stability in industrial applications was not heavily investigated. On the contrary, there are few efforts being made on this aspect academically.

A few approaches that modify the setup of NMPC have been developed to mimic the characteristic of the infinite horizon approach to achieve stability. A simple zero terminal constraint, also called as equality terminal constraint, was initially proposed but its computational requirements are very high and it often leads to aggressive behaviours as the predictive system state is being forced to origin; therefore, only a short control horizon is encouraged (Morari and Lee, 1999). To overcome this, other methods such as terminal region constraint and terminal cost function were proposed but it appeared that the addition of only the terminal cost function was insufficient to achieve stability. Thus, it came to the introduction of incorporating a terminal cost and a terminal constraint set (Mayne et al., 2000). Using this combined method, stability can be achieved provided a terminal region and a terminal cost function is suitably chosen. Alternatively, a terminal inequality constraint is used, which improves the computation efficiency and feasibility. However, the use of these approaches is a trade-off between the achievement of stability and control performance (Haverbeke et al., 2008). For a comprehensive review on terminal constraints, readers are referred to (Allgower et al., 2004), (Morari and Lee, 1999), (Mayne et al., 2000), (Magni and Scattolini, 2004) and (Mayne, 2014).

2.4 Summary

The linear-in-parameter feature of a RBF network has made it superior compared to other system identification methods. The advantages of a RBF network are its simple structure and relatively simple and efficient training approaches. The network parameters can be trained in offline mode or on-line mode using the least squares (LS) algorithm, leading to batch or recursive data training approaches. The recursive approach is preferred as new data can be taken into consideration, addressing the concern of the uncertainties in process parameters. Furthermore, the RBF network has the flexibility to be employed as a fixed or an adaptive form. An adaptive network structure is more effective for time-varying processes.

Among adaptive RBF network structure approaches, the OLS algorithm offers the advantage of being simple in executing the training algorithm effectively and in addition, for the model adaptation as well. Thus, the developed network model adaptation method in this work is based on the OLS algorithm. The OLS algorithm is initially employed to train network parameters and similar to the LS algorithm, it can be implemented in both batch (BOLS) and recursive (ROLS) training approaches. The ROLS training algorithm is extended as forward and backward centre selection methods for network structure determination. Both forward and backward methods were employed by several researchers to achieve adaptive structure RBF networks. Although the literature reviews showed that existing methods had achieved satisfactory overall modelling performances, little attention has been given to the efficacy of the new centres. The first part of the research focuses on the development of an adaptive structure RBF network using ROLS training algorithm with the objectives to improve the prediction performance (see Chapter 4).

MPC is a repetitive optimal control which its characteristics are varied depending on the characteristic of its internal model. The selection of an internal model, linear or non-linear, should be made by judging the characteristic of processes to be controlled. For linear cases, MPC has proved to be successful in industrial applications. For ineffective LMPC applications, NMPC is considered where a non-linear internal model is employed, resulting in the improvement of control performance. However, the use of

NMPC is very challenging theoretically and practically. From the literature review, it is understood that different internal models yields different optimization problems.

As reviewed, the propagation problem of the RBF network results in accumulative errors at each sampling instant which affect the modelling performance. Furthermore, the factorability of the Gaussian exponential activation function in the RBF network has played an important role in re-structuring the network model to suit the MPC approach. The major drawback of an existing method is the model compactness is compromised. In Chapter 6, a new factorable approach is introduced for RBF networks model-based MPC.

Chapter 3

RBF NETWORK MODEL & DYNAMIC PROCESS MODELLING

This chapter provides an overview of a RBF network which is used to model non-linear dynamics processes. The training algorithms are given: recursive K-means algorithm, P-nearest neighbour method and two least squares training algorithms. The final section introduces the techniques and procedure for dynamic process modelling.

3.1 RBF Network

A standard RBF network has three layers: an input layer, a hidden layer and an output layer, as depicted in Fig. 3-1. $[x_1, \dots, x_m]$ and $[\hat{y}_1, \dots, \hat{y}_p]$ are the input and output vectors with their entries being network m inputs and p outputs, respectively. The hidden layer consists of hidden neurons and each hidden neuron has a vector called centre. The hidden layer is equipped with a non-linear RBF activation function in which the information of each input neuron in the input layer is mapped into. Then, the network output is a linear combination of outputs of the activation function and network parameters.

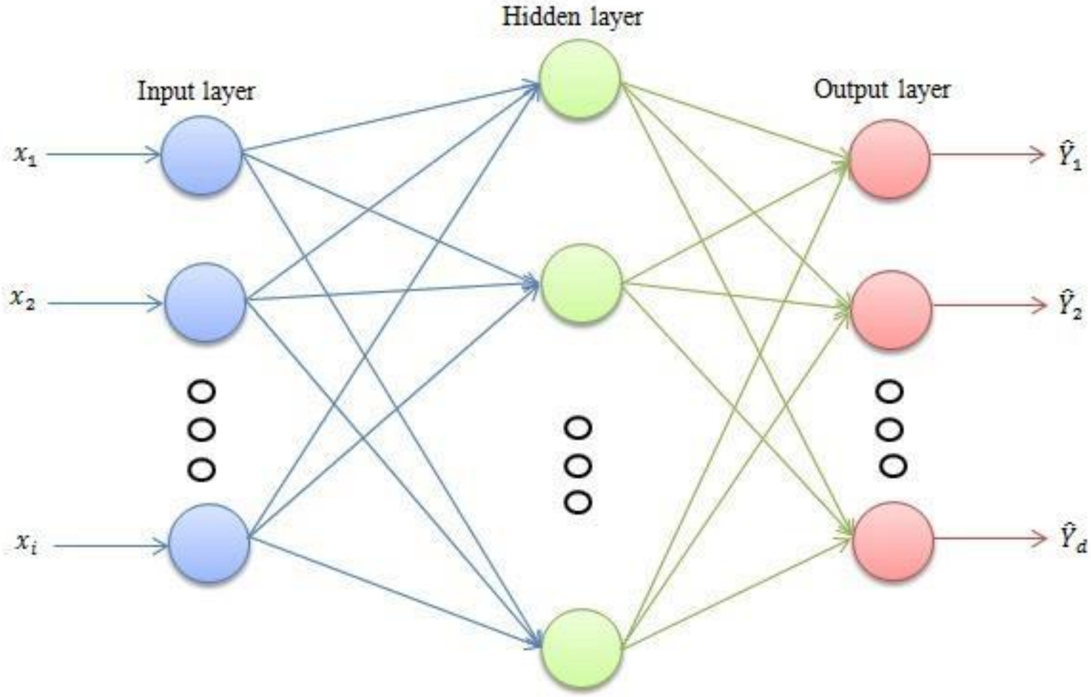

Fig. 3-1 The structure of a RBF network.

A non-linear dynamic process is presented by a NARX model in (3-1).

$$y(k) = f[(y(k-1),\dots,y(k-n_y),u(k),\dots,u(k-n_u)] + e(k) \qquad (3\text{-}1)$$

where $u \in \Re^m$ and $y \in \Re^p$ are system input and output, and n_u and n_y are input and output orders, respectively. $e \in \Re^p$ is measurement noise. A RBF network is used as an approximate for the non-linear function in (3-1), where the RBF network performs a non-linear static mapping via the linear output transformation (Gomm and Yu, 2000). The input vector x of the RBF network includes all variables in function $f(*)$ in (3-1), while the network output is $\hat{y}$. Here, the Gaussian function is used in the RBF network as the non-linear basis function in (3-1).

$$\phi_i(k) = \exp\left(-\frac{\|x(k)-c_i\|^2}{\sigma_i^2}\right), i = 1,\dots,n_h \qquad (3\text{-}2)$$

where $\phi(k)$ is the hidden layer output, n_h is the number of hidden layer nodes (centre); $x(k)$ is the network input vector and c_i is the ith centre with $i = 1,\dots,n_h$. The network output is the weighted sum of the hidden layer output and is given by,

$$y(k) = W\phi \tag{3-3}$$

where $W \in \Re^{n_h \times p}$ is the weighting matrix connecting the hidden layer nodes and network output.

3.2 Training Algorithms

In this section, the training algorithms for RBF networks are described. The recursive K-means algorithm is firstly introduced. Then, it is followed by the P-nearest neighbour method. Both of these methods are used to compute the network centres. The weight of the RBF network can be trained using different algorithms - recursive least square and recursive orthogonal least square algorithm.

3.2.1 Recursive K-means Algorithm

The recursive K-means algorithm is used to compute the position of centres (Chen et al., 1992a). The number of centres is usually decided by the user according to the complexity of the process. The procedure of K-means algorithm is as follows (Wang et al., 2006).

Choose q input data arbitrary to be the initial centres $c_1(k), c_2(k), \dots, c_q(k)$.

Assume $p(x)$ is the index of the best candidates for the centres for the input vector x, At the iteration k, find $p(x)$ by minimizing the sum squared distances using

$$p(x) = argmin\|x(k) - c_i(k)\|^2 \tag{3-4}$$

where $c_i(k)$ is the centre of the i^{th} activation at iteration k.

Step 1 Then, update the centres using the rules:

$$c_i(k+1) = \begin{cases} c_i(k) + a_c[x(k) - c_i(k)] & if\ k = p(x) \\ c_i(k) & otherwise \end{cases} \tag{3-5}$$

where a_c is the centre learning rate that lies in the range (0,1).

Step 2 Increases k by 1 and repeat **Step 1** until $c_i(k+1) = c_i(k)$.

3.2.2 P-nearest Neighbour Method

The P-nearest neighbour method that used to compute the radius of centres (Leonard and Kramer, 1991) is described as

$$\sigma_i = \sqrt{\sum_{j=1}^{p} \left\| c_i - c_j \right\|^2} \qquad i = 1, \dots n_h \tag{3-6}$$

This step is equally important as K-means algorithm as it decides the area of assigned centres that cover in the domain of the collected sample data.

3.2.3 Recursive Least Squares Algorithm

The recursive least squares (RLS) training algorithm (Ljung, 1999) is used to train the weight of the network,

$$L(t) = \frac{P(t-1)\varphi(t)}{\lambda(t) + \varphi^T(t)P(t-1)\varphi(t)} \tag{3-7a}$$

$$\widehat{w}(t) = \widehat{w}(t-1) + L(t)[y(t) - \varphi^T(t)\widehat{w}(t-1)] \tag{3-7b}$$

$$P(t) = \frac{1}{\lambda(t)}\left[P(t-1) - \frac{P(t-1)\varphi(t)\varphi^T(t)P(t-1)}{\lambda(t) + \varphi^T(t)P(t-1)\varphi(t)}\right] \tag{3-7c}$$

where $\widehat{w}(t)$ and $\varphi(t)$ represent the network weights and activation function outputs at time, t. $P(t)$ and $L(t)$ are middle terms. $\lambda(t)$ is a forgetting factor which is in the range of (0,1).

3.2.4 Recursive Orthogonal Least Squares Algorithm

The multi-variable ROLS training algorithm is developed in (Gomm and Yu, 2000). By considering a set of N input-output training data,

$$Y = \hat{Y} + E = \Phi W + E \tag{3-8}$$

where $Y \in \Re^{N \times p}$ is the desired output matrix of the system to be modelled; $\hat{Y} \in \Re^{N \times p}$ is the output matrix of neural network.; $\Phi \in \Re^{N \times n_h}$ is the hidden layer output matrix and $E \in \Re^{N \times p}$ is the error modelling matrix.

$$Y^T = [y(1), \dots, y(N)] \tag{3-9a}$$

$$\hat{Y}^T = [\hat{y}(1), \dots, \hat{y}(N)] \tag{3-9b}$$

$$\Phi^T = [\phi(1), \dots, \phi(N)] \tag{3-9c}$$

$$E^T = [e(1), \dots, e(N)] \tag{3-9d}$$

The least squares problem to solve W becomes

$$J(W) = \|E\|_F = \|Y - \Phi W\|_F \tag{3-10}$$

where $\|*\|_F$ is the F-norm of a matrix defined as $\|A\|_F^2 = trace(A^T A)$.
The hidden layer output matrix can be factorized as

$$\Phi = Q \begin{bmatrix} R \\ 0 \end{bmatrix} \tag{3-11}$$

by assuming Φ is of full rank. Q is an $N \times N$ orthogonal matrix which consists of orthonormal columns and R is an $n_h \times n_h$ upper triangular matrix.
With orthogonal transformation, (3-10) becomes

$$J(w) = \|Q^T Y - Q^T \Phi W\|_F \tag{3-12}$$

with

$$Q^T Y = \begin{bmatrix} \hat{Y} \\ \tilde{Y} \end{bmatrix}. \tag{3-13}$$

Then (3-12) becomes

$$J(W) = \left\| \begin{bmatrix} \hat{Y} \\ \tilde{Y} \end{bmatrix} - \begin{bmatrix} R \\ 0 \end{bmatrix} W \right\|_F = \left\| \begin{bmatrix} \hat{Y} - RW \\ \tilde{Y} \end{bmatrix} \right\|_F \tag{3-14}$$

where $\hat{Y}$ is an $n_h \times p$ matrix and $\tilde{Y}$ is an $(N - n_h) \times p$ matrix.
From (3-14), the optimal W can be solved from backward substitution,

$$RW = \hat{Y} \tag{3-15}$$

and leaves $\left\| \tilde{Y} \right\|_F$ as the residual. This is the batch algorithm.

For recursive ROLS training algorithm, the cost function becomes

$$J(k) = \|E(k)\|_F = \left\| \begin{bmatrix} Y(k-1) \\ y^T(k) \end{bmatrix} - \begin{bmatrix} \Phi(k-1) \\ \phi^T(k) \end{bmatrix} W(k) \right\|_F. \tag{3-16}$$

Applying QR decomposition to $\phi(k-1)$ in (3-16), and multiply the inverse of $Q(k-1)$ to $Y(k-1)$, we have

$$\Phi(k-1) = Q(k-1) \begin{bmatrix} R(k-1) \\ 0 \end{bmatrix} \tag{3-17a}$$

$$Q^T(k-1) Y(k-1) = \begin{bmatrix} \hat{Y}(k-1) \\ \tilde{Y}(k-1) \end{bmatrix} \tag{3-17b}$$

And the cost function in (3-16) becomes

$$J(k) = \left\| \left\{ \begin{array}{c} Q(t-1) \begin{bmatrix} \hat{Y}(k-1) \\ \tilde{Y}(k-1) \end{bmatrix} \\ \cdots \\ y^T(k) \end{array} \right\} - \left\{ \begin{array}{c} Q(k-1) \begin{bmatrix} R(k-1) \\ 0 \end{bmatrix} \\ \cdots \\ \phi^T(k) \end{array} \right\} W(k) \right\|_F$$

$$= \left\| \begin{bmatrix} \hat{Y}(k-1) \\ y^T(k) \\ \tilde{Y}(k-1) \end{bmatrix} - \begin{bmatrix} R(k-1) \\ \phi^T(k) \\ 0 \end{bmatrix} W(k) \right\|_F. \tag{3-18}$$

With the arrival of new data, the update using orthogonal decomposition is described as follows,

$$\begin{bmatrix} R(k-1) \\ \dots \\ \phi^T(k) \end{bmatrix} = Q_1 \begin{bmatrix} R(k) \\ \dots \\ 0 \end{bmatrix} \tag{3-19a}$$

$$\begin{bmatrix} \hat{Y}(k) \\ \dots \\ \tilde{y}^T(k) \end{bmatrix} = Q_1^T(k) \begin{bmatrix} \hat{Y}(k-1) \\ \dots \\ y^T(k) \end{bmatrix} \tag{3-19b}$$

The final cost function is

$$J(k) = \left\| \begin{Bmatrix} Q_1(t) \begin{bmatrix} \hat{Y}(k) \\ \tilde{y}^T(k) \end{bmatrix} \\ \cdots \\ \tilde{Y}(k-1) \end{Bmatrix} - \begin{Bmatrix} Q_1(k) \begin{bmatrix} R(k) \\ 0 \end{bmatrix} \\ \cdots \\ 0 \end{Bmatrix} W(k) \right\|_F$$

$$= \left\| \begin{bmatrix} \hat{Y}(k) - R(k)W(k) \\ \tilde{y}^T(k) \\ \tilde{Y}(k-1) \end{bmatrix} \right\|_F . \tag{3-20}$$

The optimal weight $W(k)$ is then solved as,

$$R(k)W(k) = \hat{Y}(k) \tag{3-21}$$

and leaves the residual as

$$\left\|\tilde{Y}(k)\right\|_F^2 = \left\| \begin{bmatrix} \tilde{y}(k) \\ \tilde{Y}(k-1) \end{bmatrix} \right\|_F^2 = \|\tilde{y}^T(k)\|_F^2 + \left\|\tilde{Y}(k-1)\right\|_F^2 . \tag{3-22}$$

The procedure of the ROLS training algorithm is therefore as follows.

1) Set the initial value for R, $\hat{Y}$ and $\left\|\tilde{Y}(k)\right\|_F^2$ as below,
   a. $R(0) = \alpha I$ where α is a small positive value.
   b. $\hat{Y}(0)$ and $\left\|\tilde{Y}(0)\right\|_F^2 = 0$.

2) At iteration k, with the arrival of new data $y^T(k)$, compute $\phi(k)$. Then, calculate $R(k)$ and $\hat{Y}(k)$ using (3-19a) and (3-19b), respectively.

3.3 Dynamic Process Modelling with RBF networks

The RBF network model can be trained as a dependent model and an independent model. In practice, a dependent model is usually more accurate as it is updated and trained using the process measurements at every sample period. However, this confines a dependent model to a single step prediction; thus, it is called one-step-ahead (OSA) predictor. On the other hand, a RBF network that trained as an independent model produces a series of multi-step ahead predictions, as illustrated in Fig. 3-2. Both dependent and independent models are NARX model based. Their relationship plays an important role in the developed f-step RBF network model in Chapter 6.

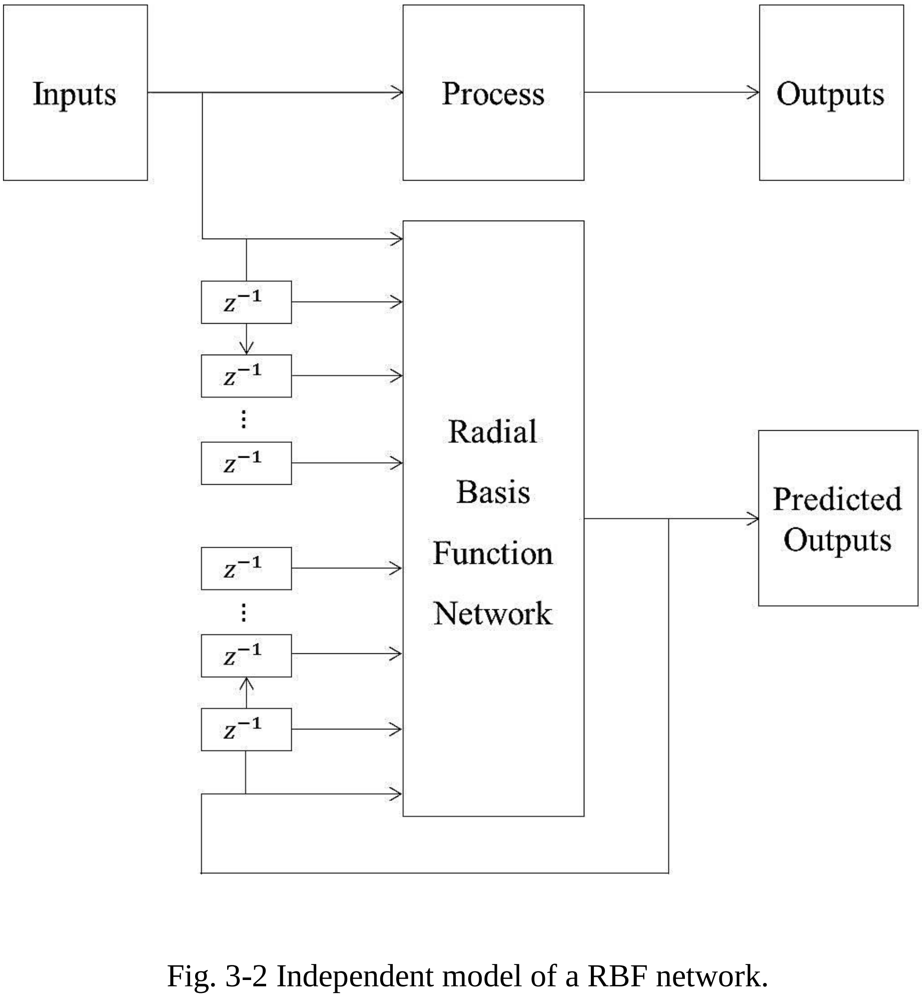

Fig. 3-2 Independent model of a RBF network.

3.3.1 Data Collection and Scaling

Data collection is a key procedure in obtaining an effective RBF network model. The collected data samples are divided into training data and validation data. The training data samples are required to cover all the interested process dynamics. In other words, the accuracy of network model is dependent on the content of information in the training data. To collect a set of informative input/output data samples, a random amplitude signal (RAS) is proposed to excite the process dynamics. The boundary of RAS is selected subject to the process constraints that correspond to interested operating regions of the process. For the pulse width of RAS, it is carefully selected depending on the response of the process. Small pulse width is preferable for a fast dynamics process as it enables the capture of fast response of the process. However, the selection of a pulse width can be an amalgamation of both small and large widths if both fast and slow responses are observed.

Both collected training and validation data sets are firstly scaled to [0 1] to minimize the error caused by the difference between ranges of different variables using a linear scale:

$$u_s = \frac{u - \min(u)}{\max(u) - \min(u)} \tag{3-23}$$

$$y_s = \frac{y - \min(y)}{\max(y) - \min(y)} \tag{3-24}$$

where u and y are input and output in raw data and u_s and y_s are the scaled data; $\min(.)$ and $\min(.)$ are the minimum values of input and output, respectively. The scaled output predictions are then scaled back after the model is used.

3.3.2 Network Structure Determination

After the data collection, the next step is to decide the model order for process inputs n_u and outputs n_y described in (3-1). The model orders can be decided by evaluating the process order. If this information is inaccessible, an empirical approach is employed. It is understood that a higher model order gives better modelling performances but it also compromises the model complexity. After this, the number of centres is predefined by users and the location of centre is computed using the recursive K-means algorithm following the procedure in Section 3.2.1. The radius of centre is calculated using the P-nearest neighbour method described in Section 3.2.2. Finally, the network parameters can be trained using RLS or ROLS training algorithms as in Section 3.2.3 and 3.2.4, respectively.

It is apparent that model orders and the number of centres decide the complexity of the network structure; thus, a trade-off between model complexity and modelling performance has to be taken into consideration. The mean absolute error (MAE), which is used to indicate the modelling performance by measuring the prediction errors, is described as

$$MAE = \frac{1}{N}\sum_{j=1}^{N}|y_i - \hat{y}_i| \tag{3-25}$$

where N is the number of data samples; y_i and $\hat{y}_i$ are outputs and predicted outputs of a process.

3.4 Summary

This chapter gives an overview of a RBF network together with its training algorithms. This includes the recursive K-means algorithm and P-nearest neighbour method for network centres. For network parameter estimation, two different least squares training algorithms, RLS and ROLS, are considered. The ROLS training algorithm is used to develop an adaptation algorithm for RBF networks in next chapter.

Dynamic process modelling techniques are described, which comprise a data selection approach, a data scaling method and the network structure determination procedure. It can be concluded that in order to obtain a good network model, the training data samples that generated using the RAS have to fully cover the interested process dynamics. In following chapters, RAS is used to generate the input signals specifically according to the processes. The network model is obtained by considering the balance between model complexity and modelling performance. In addition, basic concepts and features of dependent and independent models are explained. This forms a basis for the development of a factorized RBF network model in Chapter 6.

Chapter 4

RBF NETWORK MODEL STRUCTURE ADAPTATION

4.1 Introduction

In this chapter, a new adaptation algorithm of a RBF network structure for process with operating point migration using ROLS training algorithm is presented (Tok et al., 2015). The advantage of this proposed algorithm is that the RBF network is able to be adapted effectively and immediately to fit the new dynamics in the new operating region of the process and achieving a satisfactory overall prediction performance. In this developed algorithm, the RBF network structure, the number and location of centres, and parameter (weight) are adapted based on the novelty of new data.

An initial centre bank with a pre-specified number of centres is formed which involves the actions of adding, pruning and grouping of centres. In adding new centres, a new strategy is designed to spread more significant centres in the current operating regions to maximize the network performance. The pruning method in (Yu and Yu, 2007) is extended to prune insignificant centres from the centre bank. Then, the centres in the centre bank are divided into two groups – active centre and redundant centre groups. The centre grouping algorithm is developed using a different criterion that improves the selection of more efficient centres. Active centres are used to predict the process output, while redundant centres are preserved for next sample time. When the process operating point migrates largely, the original centres will not be effective to act for output prediction and the new centres in the region where the operating point moves to will be added. The developed algorithm is evaluated using a non-linear operating point-migrating numerical example. The effectiveness of the developed algorithm is verified by comparing it with three adaptive structure models.

This chapter is organized as follows. The adaptation algorithm is presented in Section 4.2 which includes the adding, pruning and grouping of centres. The evaluation of the developed adaptive RBF network and comparison studies is demonstrated in Section 4.3.

4.2 RBF Network Structure Adaptation

The model structure adaptation for a RBF network in this work is mainly achieved by updating the number and locations of the centres according to the current operating region. More centres will enable the network to have more accurate mapping but result in a big network size, whilst fewer centres will reduce the mapping accuracy but result in a smaller network, which consequently enhance the model generalization and reduce computing load.

Based on the ROLS training algorithm described in Section 3.2.4, the adaptation of the RBF network is implemented by evaluating the contribution of each centre to the model prediction performance, and then according to the contribution to decide which centre will be added or pruned. Also, the location of the added centre needs to be determined to reflect the migration of the system operating point. Firstly, an initial centre bank with a pre-specified number of centres is formed by arbitrarily selecting some input data points as initial centres. Secondly, at each sample time, the network learns the information of the centre with the most contribution and the information of the new data. Then, the location of the added centre is determined according to the learned information.

The third step is to prune a centre, which has the least contribution among the centres in the centre bank at each sample time. This is to maintain the size of the centre bank, which also maintains the computational demand that have been increased from the addition of new centres. The last step is that, after updating the centre bank with the added and pruned centres, the centres are classified into two groups, active centre group and redundant centre group, based on their contributions to the network performance. The aim of the strategy to group these centres is to achieve a compact optimal network structure without degrading the network performance. Active centres will have bigger weight in contribution to the network output compared to redundant centres. Active centres are used for network prediction, while redundant centres are preserved for the later selection at the next sample time.

4.2.1 Centres Adding Strategy

For the structure of the RBF network, adding a new centre means adding a hidden neuron. A new strategy of adding new centres is designed based on the information combining the centre giving the most contribution to the network performance and the new data. At sample time k, the matrix $R(k-1)$ is updated with new data $x(k)$ using ROLS training algorithm. From the updated matrix $R(k)$ that contains the information of new data, the contribution of each centre to the network performance is evaluated. Consider the evaluation index for contribution of each centre proposed in (Gomm and Yu, 2000),

$$\|\hat{Y}\|_F = \sum_{i=1}^{n_h} \|\hat{y}_i \hat{y}_i^T\|_F \tag{4-1}$$

where $\hat{y}_i^T$ is the ith row of $\hat{Y}$. This shows that ith centre has a separable contribution of $\|\hat{y}_i \hat{y}_i^T\|_F$ to $\|\hat{Y}\|_F$. Thus, the centre with the most contribution C_{mc} can be found by computing $\|\hat{y}_i \hat{y}_i^T\|_F$ for each centre and then compare them. The assessment of centre contribution is done by moving jth column in matrix R^* to lth column position sequentially as

$$R^* = [r_1, \dots, r_{l-1}, \dots, r_j, r_l, \dots, r_{j-1}, r_{j+1}, \dots, r_{n_h}] \tag{4-2}$$

For $j \neq l$, matrix R^*, which is no longer an upper triangular matrix after the moving of its column, is re-triangularized by applying an orthogonal decomposition

$$R^* = Q^* R_j^* \tag{4-3a}$$

$$\hat{Y}_j^* = \begin{bmatrix} \hat{Y}_j^1 \\ \hat{y}_{j,l}^T \\ \hat{Y}_j^2 \end{bmatrix} = Q^{*T} \hat{Y} \tag{4-3b}$$

with $\left[\hat{Y}_j^1\right]^T = [\hat{y}_{l,1}, \dots, \hat{y}_{j,l-1}]$ and $\left[\hat{Y}_j^2\right]^T = [\hat{y}_{j,l+1}, \dots, \hat{y}_{j,n_h}]$. Thus, the contribution of jth centre at lth stage is $\left\|\hat{y}_{j,l}\hat{y}_{j,l}^T\right\|_F$.

The location of the added centre should consider both the centre with C_{mc} and the new data $x(k)$. The former represents the location for more effective centre, while the latter represents the current operating region of the process. Ideally the best location for the added centre should be found by the line search along the connection line of the most effective centre and the new data, which is the optimal location in terms of maximal contribution to the prediction of current system output. In this research, the location of the new centre is determined by the equation in (4-4) with a proper ε,

$$C_{new} = \varepsilon C_{mc} + (1 - \varepsilon) x(k) \tag{4-4}$$

where $0 < \varepsilon < 1$ is the parameter to be selected using the trial and error method for specific process. Smaller ε tends to use the current effective centre location, while the bigger ε tends to move the new centre to the new operating region. A compromise between the two can generate a smoother move to the new operating region which will benefit the future predictions. After adding a new centre, new matrix R_{new} with previous N samples is retrained using (Yu and Yu, 2007),

$$\begin{bmatrix} \phi(k-N+1) \\ \phi(k-N+2) \\ \vdots \\ \phi(k) \end{bmatrix} = Q_{new} \begin{bmatrix} R_{new} \\ 0 \end{bmatrix} \tag{4-5}$$

$$\hat{Y}_{new} = Q_{new}^T \begin{bmatrix} y(k-N+1) \\ y(k-N+2) \\ \vdots \\ y(k) \end{bmatrix} \tag{4-6}$$

where R_{new} and $\hat{Y}_{new}$ are the updated matrices with newly added centres.

4.2.2 Centres Pruning Method

In order to maintain the size of the centre bank, an insignificant centre is pruned from the centre bank. In other words, a centre which has the least contribution to the network performance is removed. For a RBF network structure, pruning a centre implies removing a hidden layer neuron which is associated to a column vector in matrix R. To calculate the modelling residual, each column of matrix R is removed, sequentially, and the matrix R is re-triangularized (Hong and Billings, 1997, Gomm and Yu, 2000). The pruning algorithm using orthogonal decomposition developed in (Hong and Billings, 1997, Gomm and Yu, 2000) is as follows. If jth centre is removed, the corresponding jth column vector of matrix R, r_j is removed as well, which results in matrix R',

$$R' = \left[r_1, \dots, r_{j-1}, \dots, r_{j+1}, \dots, r_{n_h}\right] \tag{4-7}$$

and the weight matrix is now

$$W'^T = \left[w_1, \dots, w_{j-1}, \dots, w_{j+1}, \dots, w_{n_h}\right] \tag{4-8}$$

where w_i^T is the ith row of W. After the removal of the column r_j, the cost function in (3-20) becomes

$$J_j^2 = \left\|\begin{bmatrix} \hat{Y} - R'W' \\ \tilde{Y} \end{bmatrix}\right\|_F^2 = \left\|\begin{bmatrix} r_j w_j^T \\ \tilde{Y} \end{bmatrix}\right\|_F^2$$

$$= \left\|\left[r_j w_j^T\right]\right\|_F^2 + \left\|\tilde{Y}\right\|_F^2. \tag{4-9}$$

The matrix R' is no longer an upper triangular matrix. Thus, it is necessary to re-triangularize the matrix R',

$$R' = Q' \begin{bmatrix} R_j \\ \dots \\ 0 \end{bmatrix} \tag{4-10a}$$

$$\begin{bmatrix} \hat{Y}_j \\ \dots \\ \tilde{y}_j^T \end{bmatrix} = Q'^T \hat{Y} \tag{4-10b}$$

and the cost function becomes

$$J_j^2 = \left\| Q' \begin{bmatrix} \hat{Y}_j \\ \cdots \\ \tilde{y}_j^T \end{bmatrix} - Q' \begin{bmatrix} R_j \\ \cdots \\ 0 \end{bmatrix} W_j \atop \tilde{Y} \right\|_F^2 = \left\| \begin{matrix} \hat{Y}_j - R_j W_j \\ \tilde{y}_j^T \\ \tilde{Y} \end{matrix} \right\|_F^2$$

$$= \left\| \hat{Y}_j - R_j W_j \right\|_F^2 + \left\| \tilde{y}_j^T \right\|_F^2 + \left\| \tilde{Y} \right\|_F^2. \tag{4-11}$$

The weight, W_j can be solved from

$$R_j W_j = \hat{Y}_j. \tag{4-12}$$

The residual is given as

$$\left\| \tilde{Y}_j \right\|_F^2 = \left\| \tilde{y}_j^T \right\|_F^2 + \left\| \tilde{Y} \right\|_F^2. \tag{4-13}$$

From (4-13), it can be seen that the increment in residual caused by removing the jth column of matrix R, jth centre, is $\left\| \tilde{y}_j^T \right\|_F^2$. Thus, the procedure is summarized as: use (4-7) to remove the column of matrix R in turn and compute the residual $\left\| \tilde{y}_j^T \right\|_F^2$ using (4-13). Then, the jth column of matrix R with least residual $\left\| \tilde{y}_j^T \right\|_F^2$ is removed, and matrix R is re-triangularized using (4.10).

4.2.3 Centres Grouping Strategy

After the adding and pruning centres, the centres in the centre bank are classified into two groups which are active centres and redundant centres. The centres in the active group will be used to predict the process output, while the centres in the redundant group will not be included in the network for process output prediction at this sampling period, but will be preserved for later use in the consequent sampling instants. So, the relation between the hidden neurons and the output neurons for active and inactive centres are illustrated in Fig. 4-1. While, the redundant centres, which may contain the information for next sample time, are preserved in the centre bank.

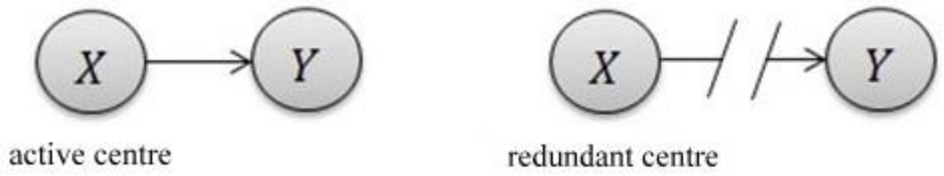

Fig. 4-1 The connections between hidden neuron X and output neuron Y for active and redundant centres.

The centre pruning algorithm provides a good foundation for centre grouping. This is implemented by evaluating the modelling residual when each centre is grouped as a redundant centre, sequentially. When the grouping procedure stops, the remaining centres would be active centres. In other words, it is the contribution of each centre to the network performance that decides which group the centre belongs to. Akaike's final prediction error (FPE) criterion in is used to stop the grouping procedure,

$$FPE = \frac{1+\beta(n_p/N)}{1-\beta(n_p/N)}V \tag{4-14a}$$

$$V = \left\|\tilde{Y}(N)\right\|_F^2/(N) \tag{4-14b}$$

where V is the loss function, n_p is the number of weights and β is a weighting factor. The value of $\beta = 2$ is suggested in (Gomm and Yu, 2000). However, due to that the sample data N is a fixed parameter in (4-14) for every sample time k, the value of β can be manipulated to decide the number of active centres. In order to stop the grouping procedure, FPE has to be larger than the past FPE FPE_i. Thus, the equation to calculate number of active centres is derived as

$$1-\beta(n_p/N) = 0 \tag{4-15a}$$

$$n_p = \frac{N}{\beta} \tag{4-15b}$$

$$c_{active} = \frac{N}{\beta} \tag{4-15c}$$

where c_{active} is the number of active centres.

The procedure of centre grouping algorithm is summarized as follows:

Step 1 Initialize V and FPE for the network after updating the centre bank.

Step 2 Compute the new loss function V_j when each centre is grouped in turn using (4-7) and (4-13).

Step 3 Set i = arg min (V_j) and compute the FPE for the smallest loss function, FPE_i using (4-14).
If $FPE_i < FPE$, group the centre i as redundant centre and go to **Step 4**.
If $FPE_i > FPE$, go to **Step 5**.

Step 4 Then, set $= R_i$, $\hat{Y} = \hat{Y}_i$, $V = V_i$, FPE $=$ FPE$_i$ and $n_h = n_h - 1$. Go to **Step 2**.

Step 5 Stop the grouping procedure. The remaining centres in the centre bank are active centres and the optimal weight W_j can be computed using (4-12).

4.2.4 RBF Network Adaptation Procedure

At each sample time, the centre bank will be updated with the adding, pruning and grouping of centres. The main step of the proposed adaptive algorithm is summarized as follows.

Step 1 Initialize an initial RBF network by using a set of N samples data, form a centre bank by arbitrarily choosing data points and obtain an initial matrix R and W.

Step 2 At each sample time k, update the matrix R with new data $x(k)$ using (4-10a). Evaluate the contribution of centres and add a new centre into centre bank using (4-1) and (4-4), respectively. Then, generate a matrix R_{new} and $\hat{Y}_{new}$ using (4-5) and (4-6), respectively.

Step 3 Prune a centre that causes the least increase in modelling residual from the centre bank by following the summarized pruning procedure given in Section 4.2.2.

Step 4 Group the centres in the centre bank into two groups: active and redundant centres, using the provided grouping procedure in Section 4.2.3. Use the active centre to form a network model to make prediction.

Step 5 if $k = N$, terminate the algorithm, else $k = k + 1$ and go to **Step 2**.

4.3 Performance Evaluation

In order to verify the effectiveness and to demonstrate the advantages of the proposed adaptive RBF network model, three existing RBF network models with structure adaptation in (Han et al., 2011, Yu et al., 2004, Yu and Yu, 2007) are employed for performance comparison purpose. Both methods in (Yu et al., 2004, Yu and Yu, 2007) are also ROLS training based. Thus, these two employed adaptive structure RBF network models (Yu et al., 2004, Yu and Yu, 2007) are briefly explained here as to show the differences of features being employed between the compared networks and the proposed algorithm.

The adaptation of network structure in (Yu et al., 2004) is done by updating the centres with new measurements and the least significant centre is pruned in each sample period. Then, a set of centres is selected by making use of the Akaike's FPE criterion to form a network structure for prediction. In the method in (Yu and Yu, 2007), the network structure is adapted by adding the new data as a new centre when the prediction error is larger than the error threshold and a centre with least contribution is pruned. Similarly, a network structure is then formed using the all the centres in the centre bank.

The main difference between the existing methods and the proposed method in this thesis is that, the remaining centres in the proposed method, after adding and pruning, are divided into active and inactive groups with a different criterion. Then the centres in the active group are used for prediction, while the centres in the inactive group are preserved for later selections. In addition, a new learning strategy is proposed to optimize the option of centres in the centre bank. This makes the proposed method outperforms over the existing methods.

To compare the prediction performance of the proposed method and the three existing methods, a non-linear dynamic system with a large migration of the operating point is used to be modelled by the four methods. The system is chosen from (Narendra and Parthasarathy, 1990),

$$y(k+1) = \frac{y(k)}{1+y(k)^2} + u(k)^3 \tag{4-16}$$

A set of 900 input/output data samples has been generated and collected in a specific way where the system outputs fall into three obvious different regions. Region 1 represents the first 300 data, region 2 represents data samples from 301 to 600, and region 3 represents the 601 to 900 data. This is to validate the effectiveness of adaptation and the performance in recovery speed of the proposed algorithm during the migration of system's operating point.

In this simulation, the proposed adaptive RBF network model is chosen to have two inputs, one output and an initial centre bank with 20 centres. In this simulation, the proper value for ε is 0.3. The N and β in (4-15) are selected as 50 and 4, respectively. The number of active centres is calculated using (4-15c). Thus, there are 13 active centres and 7 redundant centres in the centre bank. The two employed adaptive structure RBF network models (Yu et al., 2004, Yu and Yu, 2007) are chosen to have same centre bank size with the proposed algorithm of 20 centres. While the flexible structure RBF network (FS-RBF) (Han et al., 2011) is chosen to have 2 initial centres and the 'best' values of the parameters for the network are obtained after 40 preliminary runs.

The evaluation of the network performance is emphasized on its overall prediction error and the recovery speed during the change of system's operating regions. The MAE is used to measure the network prediction error while the performance in recovery speed of the network can be observed during the changing of system's operating regions. The MAE values of the four RBF network models are compared in Table 4-1.

As can be seen from Fig. 4-2, the proposed algorithm and the adaptive RBF network in (Yu et al., 2004) have achieved similar recovery speeds during the migration of system's operating regions. But from Fig. 4-3, it is observed that the prediction performance of the adaptive RBF network in (Yu et al., 2004) is not satisfactory and from Table 4-1, it has the worst performance among the four networks. The comparison between the performance of network in (Yu et al., 2004) and proposed algorithm shows that the proposed algorithm is able to select more significant centres based on the modified centre grouping criterion that being employed in this work. Apart from the efficacy of selecting more efficient centres, the option of centres that available in the centre bank is also critically important. Thus, the learning strategy in this research has an important role in providing a set of good centre candidates in the centre bank.

Networks	**MAE**
Proposed Adaptive RBF Network	0.0436
Adaptive RBF Network in (Yu et al., 2004)	0.1770
Adaptive RBF Network in (Yu and Yu, 2007)	0.0780
FS-RBF network in (Han et al., 2011)	0.0594

Table 4-1 The performance comparison of the four different RBF networks.

From Table 4-1, it can be noticed that the proposed algorithm and the adaptive RBF networks in (Han et al., 2011, Yu and Yu, 2007) have achieved comparable overall prediction performance but the proposed algorithm has a better recovery speed during the changing of system's operating region as illustrated in Fig. 4-2 (top and bottom). After the migration of system's operating region, the performance of adaptive RBF network in (Yu and Yu, 2007) is degraded and it needs 8 to 11 sample periods to recover as shown in Fig. 4-2 (top and bottom). Meanwhile, the FS-RBF network (Han et al., 2011) needs 3-5 sample periods to recover its performance. Beside from this, the drawback of FS-RBF network is a series of preliminary runs is needed to obtain the 'best' parameters of the network. This comparison demonstrates that the learning strategy in proposed algorithm is more capable of generating more significant centres which adapts the system quickly and effectively.

Fig. 4-4 shows the predicted system output by the proposed method only. The 13 active centres were adapted effectively to all regions of the system output, as illustrated in Fig. 4-5. It can be observed that the 13 active centres emigrate from region 1 to region 2, then to region 3 following the moving of the system's operating points. Also from Table 4-1, the values of MAE clearly suggest that the proposed algorithm has the best performance among four networks. All simulation results can be summarized that the proposed algorithm achieves the best prediction performance and shows the fastest recovery speed during the changes of system's operating regions.

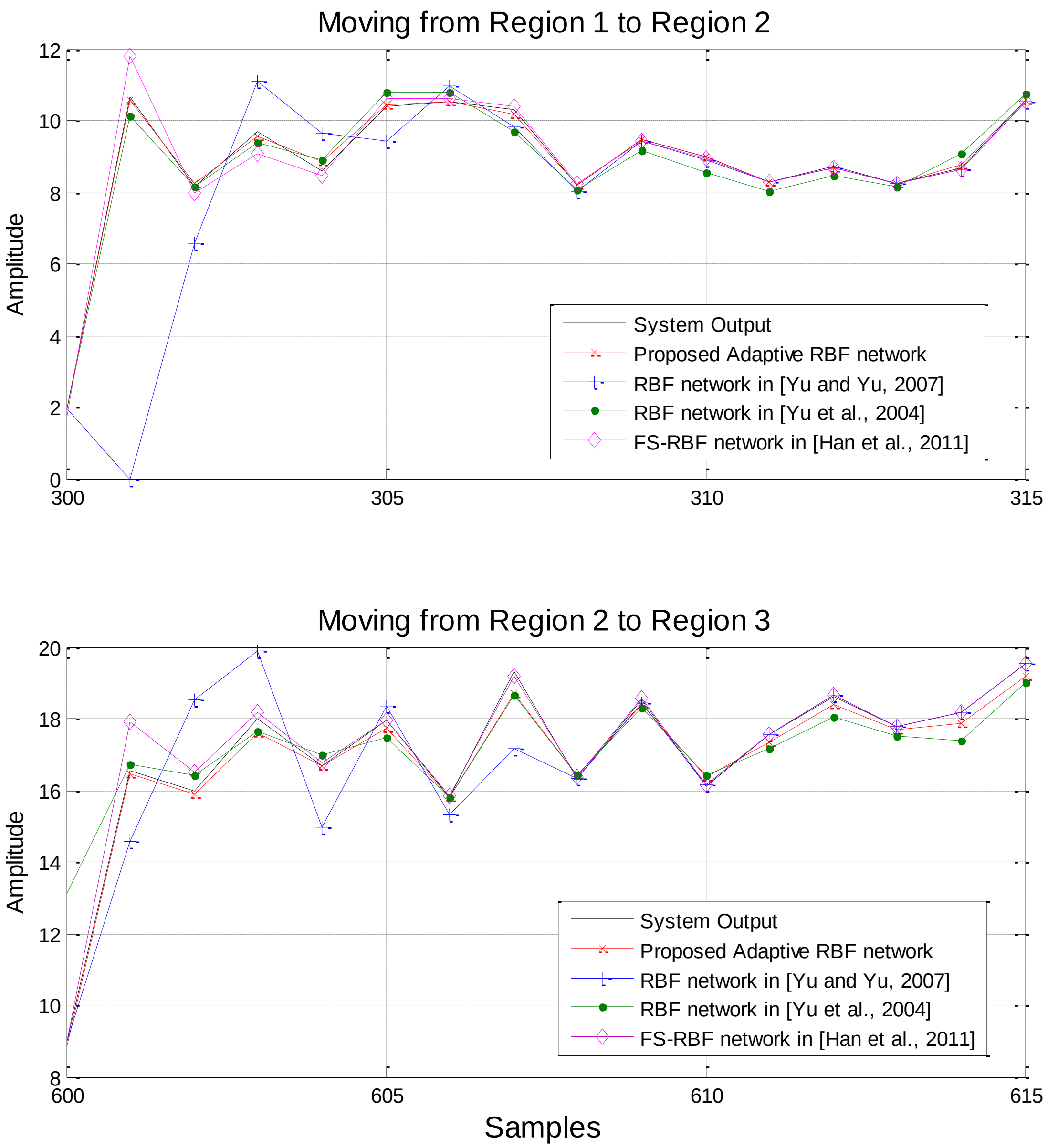

Fig. 4-2. The performance of four networks during the change of system’s operating regions.

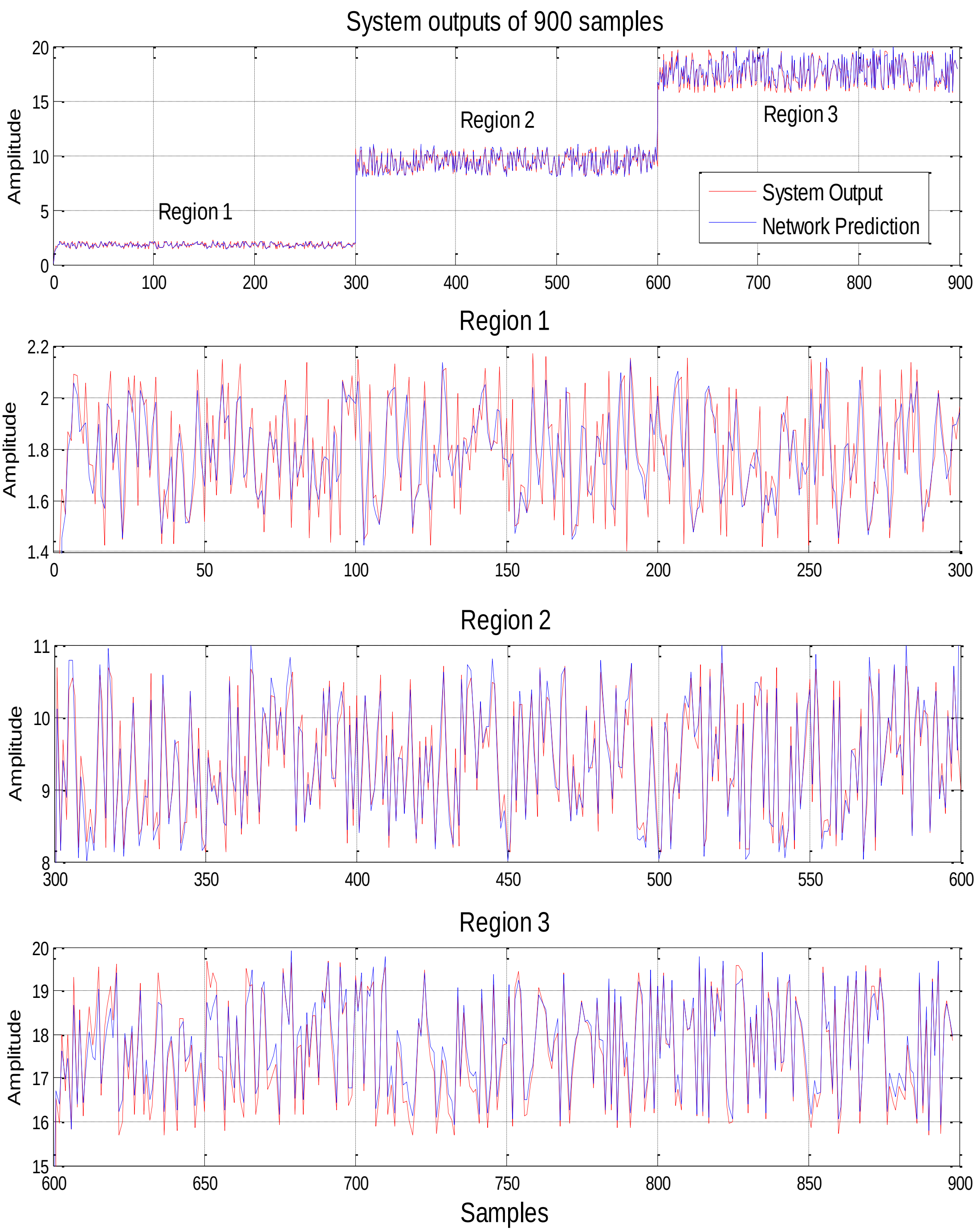

Fig. 4-3 The performance of the adaptive structure RBF network in (Yu et al., 2004).

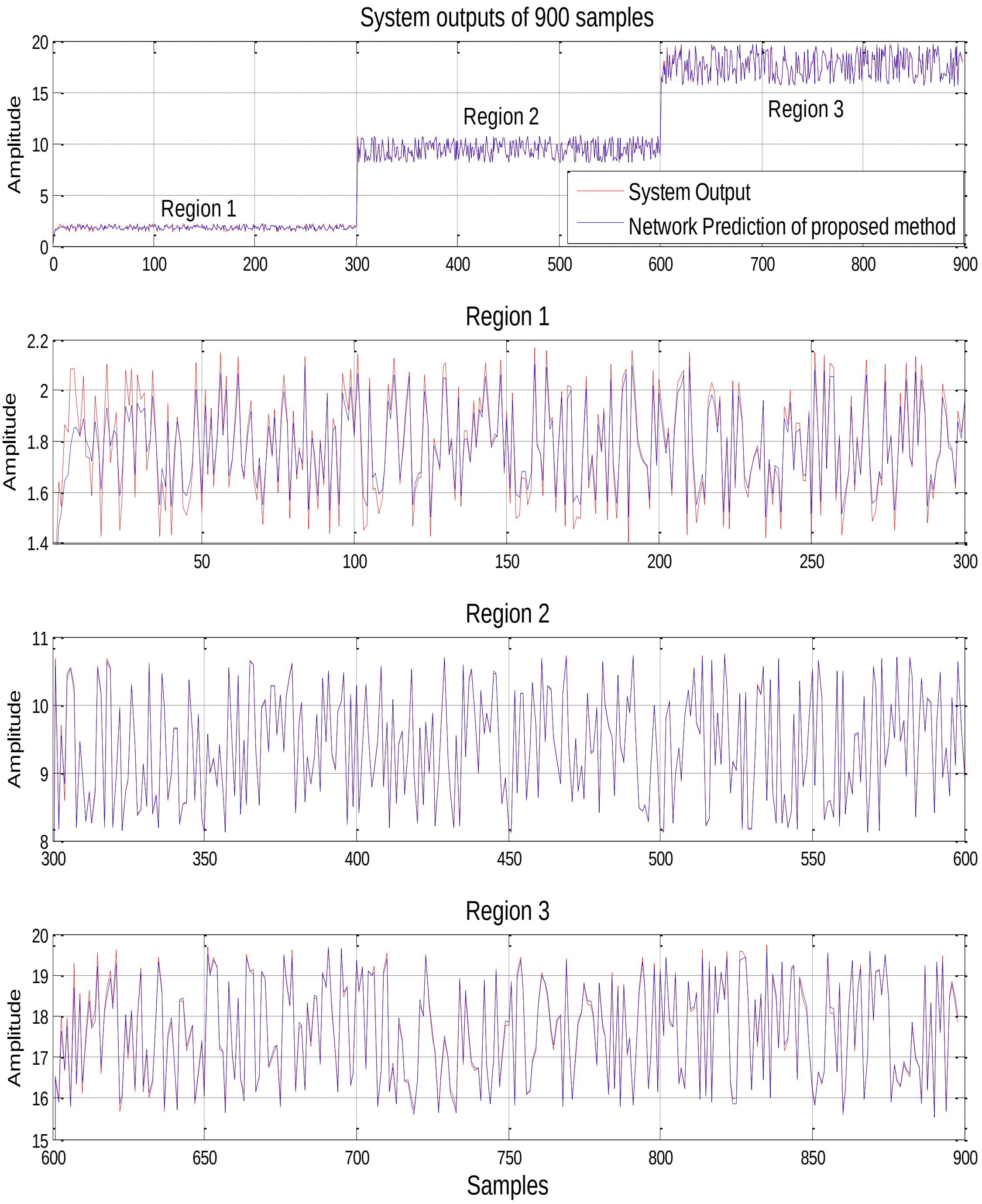

Fig. 4-4 The performance of the proposed adaptive structure RBF network.

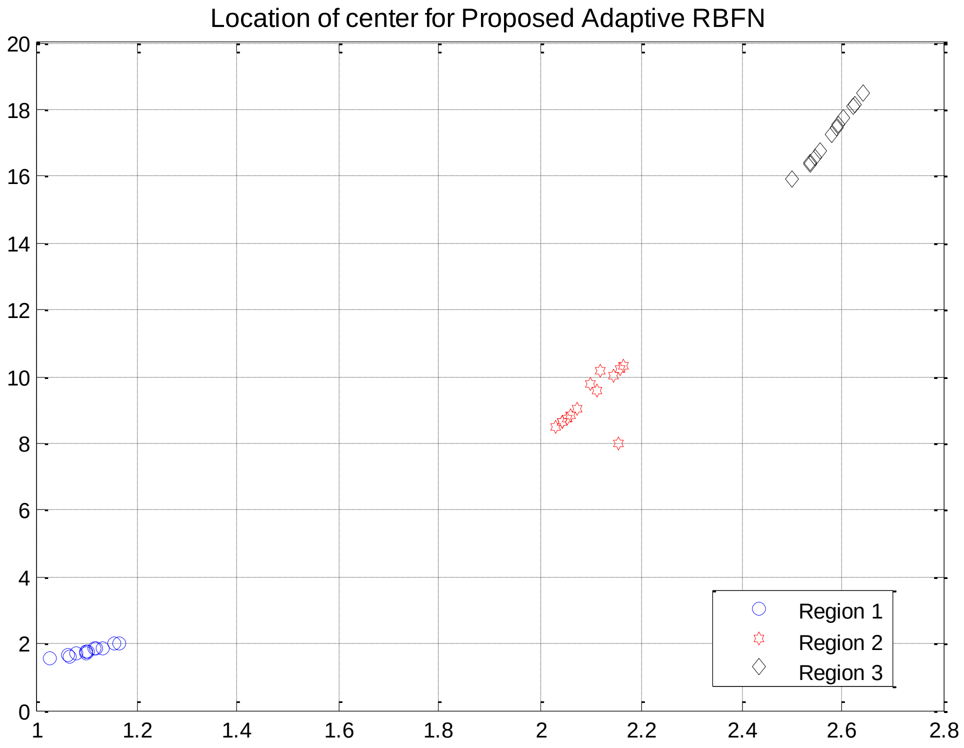

Fig. 4-5 The location of centres of the proposed network in three regions.

4.4 Summary

In this chapter, a new algorithm based on the ROLS training has been proposed for designing a structure adaptive RBF network model. A new strategy of adding new centres based on the information of the centre with the most contribution and the new data is developed. In the meantime, the insignificant centre is pruned from the centre bank to maintain the minimum size of the network model. In addition, a small modification on the parameter of FPE enables the network to improve the efficiency in grouping the centres in the centre bank. The effectiveness of the proposed algorithm is demonstrated by applying it in modelling a non-linear numerical example with significant operating point migration. The simulation results demonstrate that the developed RBF network adapts its structure dynamically following the migration of the system operating point without degrading the prediction performance. Comparison with the three existing adaptive structure RBF networks demonstrates the advantages of proposed algorithm in term of the prediction performance and the recovery speed during the migration of system's operating regions.

Chapter 5

MODEL PREDICTIVE CONTROL & OPTIMIZATION ALGORITHMS

This chapter includes two main parts. The first part introduces the concept of the model predictive control, the receding horizon idea and the main features. The key tuning parameters are explained. The optimization algorithms in the second part have three subsections, namely Linear Programming, Quadratic Programming and Non-linear Programming. The respective techniques for constrained optimization problems included are Simplex method, Active Set method, and Sequential Quadratic Programming. Besides, related methods for unconstrained problems are discussed. In addition, the formulation of Lagrange method and optimality conditions is discussed.

5.1 Model Predictive Control

Model predictive control (MPC) has a simple but effective concept that reflects human behaviours (Rossiter, 2003). Human normally reacts with a purpose that yields an intended outcome. In relation to the MPC, it can be said that human decides control actions based on vision, predicting that the control actions will result to the desired output. For MPC, an internal model is employed to make future predictions which act like the human vision. These predictions are constantly updated with new observations to assist in deciding the control actions. Eventually, this set of on-line control action leads to a desired result, expressing the concept of a receding horizon approach. With this idea in mind, the MPC has unique abilities to control multivariable processes and incorporate defined constraints for inputs and outputs which is critical in preventing processes to exhibit aggressive behaviours (Qin and Badgwell, 2003).

5.2 Receding Horizon Approach

A block diagram of the MPC in generalised form is shown in Fig. 5-1, obtained in (Seborg et al., 2010). The role of the model is to predict a series of outputs within a defined prediction horizon. The residuals between the process actual outputs and model predicted outputs are used as a feedback to the prediction block. This helps correct the deficit of the prediction outputs which eliminates offset steady-state errors in the set-point calculation block. In the control calculation block, an optimization algorithm is employed to solve a cost function that defined by a control objective respecting all the constraints.

The set-points calculation is useful when an economic objective function is considered such as a production rate or a profit function. In this research, this part is ignored as desired outputs are already known. For more detail on set-point calculations, readers are referred to (Seborg et al., 2010). In process control, the set-points are normally set to a constant value or time-based varying desired values according to the control objective.

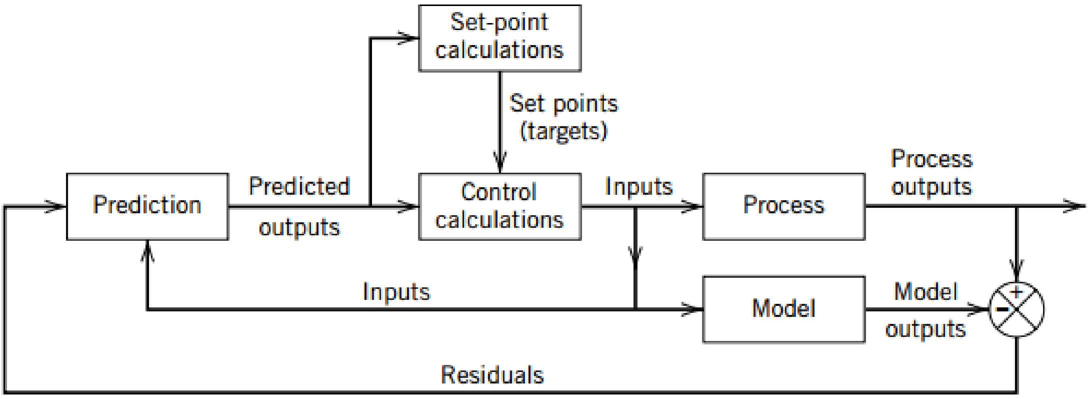

Fig. 5-1 Block diagram of the MPC (Seborg et al., 2010).

The control calculations are implemented in adopting the receding horizon mechanism as illustrated in Fig. 5-2, which involves two important user-defined parameters: a control horizon and a prediction horizon. Assuming a single-input and single-output (SISO) process, the calculations are computed based on current process output y_k and future output predictions over a prediction horizon $\left[y_{k+1}, \dots, y_{k+H_p}\right]$. The reference point trajectory is denoted as r_k at time instant k. The strategy is to compute a sequence of control moves $\left[u_k, \dots, u_{k+H_u}\right]$ to drive the predicted outputs to the reference point trajectory. The control inputs are remained constant after the control horizon H_u. The control moves are determined by optimizing a cost function V_k in (5-1) that described in Fig. 5-2.

$$V_k = \sum_{i=H_w}^{H_p} \left\| \hat{y}_{k+i|k} - r_{k+i|k} \right\|_{Q_i}^2 + \sum_{i=0}^{H_u-1} \left\| \Delta\hat{u}_{k+i|k} \right\|_{R_i}^2 \tag{5-1}$$

where R_i and Q_i are the penalty on the changes in inputs $\Delta\hat{u}_{k+i|k}$ and the errors between predicted outputs $\hat{y}_{k+i|k}$and desired set-points $r_{k+i|k}$, respectively. H_w and H_u represent a window parameter for the prediction horizon and the control horizon, respectively.

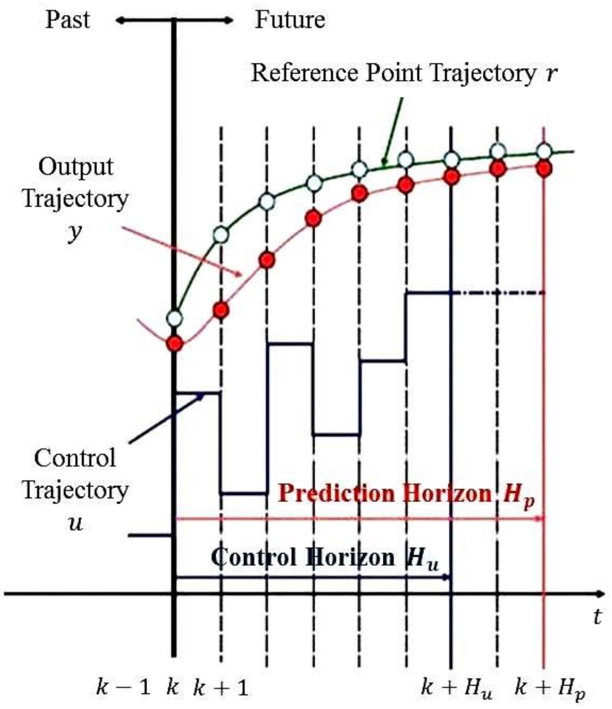

Fig. 5-2 Receding horizon strategy.

After calculating the control moves, only the first element is applied as the input to the process. With this implementation, all updated process measurements (outputs) and conditions are obtained for the next sample time instant $k + 1$. Then, the whole procedure is repeated. Therefore, it is called a receding horizon approach. Notice that this on-line optimization method yields a feedback control strategy.

5.3 The Features of Model Predictive Control

The features of MPC and some important remarks are summarized as (Rossiter, 2003):

1) The control law is based on the predicted behaviour.

This is a major benefit as the execution of control inputs is dependence on the future behaviours of a process to be controlled. This helps improve the efficiency in computing the control signals.

2) The output predictions are computed using an internal model.

An internal model should be descriptive enough to represent system dynamics while the model complexity should not be ignored. In addition, a precise internal model helps to envisage the possibility of potential problems.

3) The control input is computed by optimising the measure of predicted performance.

This point implies the efficiency of an internal model plays an important role in the control performance. It is only possible to demand high quality control performance if the internal model is accurate enough.

4) The control design for multivariable systems.

The MPC framework allows a multivariable system (square or non-square system) to be controlled in a straightforward manner. Compared to conventional control methods, a PID controller has to employ a complicated design such as a decoupling framework for a multivariable system. In addition, the interactions between system inputs and outputs are addressed directly in MPC.

5) The constraint handling.

As discussed in Chapter 2, this is the biggest selling point and also the most important feature for MPC. The adoption of constraints in the on-line framework of MPC is compellingly unmatchable by other control methods as described in (Seborg et al., 2010). The inequality constraint on inputs is particularly useful as it provides an operating window for practical applications such as the physical limitation of an actuator. Moreover, the inequality constraint on outputs offer an advantageous approach called range control, which is useful for a process without a fine set-point.

5.4 Tuning Parameters

There are few key parameters that have impacts on the control performance of the MPC, as illustrated in Fig. 5-2. These parameters are decided by users and their interaction effects are discussed as follows.

a) Sampling Period

The first parameter to be decided is a sampling period which has to be chosen adequately to match the computation time for solving the optimization problem. The choice of sampling period is, however, restricted as it also has to be selected depending on the process dynamics in relation to the process modelling performance.

b) Prediction and control horizon

The window parameter H_w is selected according to the dead time of a process. Understandably, a small prediction horizon H_p leads to an aggressive controller. On the hand, a large control horizon H_u yields a complex controller. The control horizon H_u is normally smaller than the prediction horizon H_p. This means that it has a small degree of freedom and the control performance might be degraded. However, on the positive side the computational load is greatly reduced as the dimension of decision variables to be optimized is smaller. Thus, the control horizon is selected by considering a trade-off between the computational requirement and the desired control performance.

c) Penalty Matrices

The other additional tuning parameters are penalties on the errors R_i and control moves Q_i, as expressed in (5-1). Both penalties are diagonal matrices. For R_i matrix, its penalty values can be varied individually according to the importance of each output for a multivariable case. A larger ratio is used for a more important output. The higher the penalty values in R_i matrix, the more aggressive the controller becomes. The Q_i matrix has a similar advantage comparatively; however, it has an opposing influence where smaller values will lead to a more aggressive control effect.

The guideline of tuning parameters mentioned above can be found in (Seborg et al., 2010). However, the tuning effect may differ depending on the internal model. This can be observed in the selections between a sampling period and a control horizon. A selected sampling period has to satisfy a desired modelling performance while meeting the computational requirement for solving the optimization problem in (5-1) which its complexity is decided by the selection of control horizon. At the same time, a long prediction horizon is preferred if a precise internal model is available but this also increases the complexity of the optimization problem which has an indirect impact on the sampling period. With the effects contributed by each parameter, they are tuned empirically for an optimum control performance considering necessary trade-offs.

5.5 Optimization Algorithms

This section discusses the optimization algorithms for MPC. Solving an optimization problem is a major part in the MPC. Control engineers often have less experience with optimization algorithms. Mathematical optimization is a very wide topic with many algorithms available with different implementations but many algorithms are based on similar philosophies. It is imperative to stress that this section focuses on the general concepts of optimization algorithms of interest to develop a foundation of knowledge in this discipline. The section starts with a simple problem, Linear Programming (LP), and proceeds to a special case, Quadratic Programming (QP). Finally, the key concept of Non-linear Programming (NLP) is included. All related techniques with respect to the three optimization problems are included. The concepts of relevant techniques and their respective relationships are considered. The constructed materials are mainly taken from (Luenberger, 1984), (Griva et al., 2009) and (Borwein and Lewis, 2010).

5.6 Linear Programming

A Linear Programming (LP) problem consists of a linear objective function subjects to linear constraints. Understanding the LP is vital as it forms the basis for mathematical optimization. In this section, Simplex method is covered as its concept of searching an optimum solution is identical to a QP algorithm (Active Set method). Moreover, Phase 1 problem is an important first step to find an initial feasible solution for LP and QP problems. The linear behaviours of a LP problem identify the characteristics that provide readers with sufficient information to understand the problem in detail. This makes the LP an easier problem to solve. The key properties of a LP problem can be summarized as (Griva et al., 2009)

1) If an optimum solution exists, then the optimum solution is located at corner points (vertex) where a corner point is an intersection between constraint boundaries. This simplifies an optimum solution searching procedure significantly.

2) If there is more than one optimum solution, then optimum solutions are in the form of a line or a plane.

3) The problem may be infeasible (no optimum solution) or unbounded (infinite optimum solution).

The general form of a LP problem is not convenient to implement. Hence, a generalised standard form is introduced. Considered a general problem with equality constraints ex and inequality constraints hx,

$$\max_{x} c^T x \tag{5-2a}$$

subject to

$$hx \leq b \tag{5-2b}$$

$$ex = l. \tag{5-2c}$$

The standard form, which employs slack variables s to convert inequality constraints in (5-2b) into compatible form of equality constraints and unrestricted variables x to express the differences between nonnegative variables, can be described as follows.

$$\min_{x}[c^T - c^T\ 0]\,x, \quad x = \begin{bmatrix} x^+ \\ x^- \\ s \end{bmatrix} \tag{5-3a}$$

subject to

$$\begin{bmatrix} h & -h & I \\ e & -e & 0 \end{bmatrix} x = \begin{bmatrix} b \\ l \end{bmatrix} \tag{5-3b}$$

$$x \geq 0. \tag{5-3c}$$

For a problem with ≥ inequalities constraints, surplus variables are used instead of slack variables. On the contrary, multiply the problem in (5-2a) with -1 for a maximization.

A LP problem can be solved by sketching the problem and its constraints provided the problem is feasible, bounded and an optimum solution exists. An optimum solution can be found by inspecting the corner points. This is called graphical method. The disadvantage of this method is only restricted to a small-scale problem with a relatively small variable dimension that allows for sketching. However, the graphical method provides an effective way to interpret the properties of a LP problem. An alternative method is to compute all corner points and evaluate the objective function with all corner points but this method is exhaustive when a large scale-problem is considered. Hence, it is called exhaustive search method.

5.6.1 Simplex Method

A popular method to solve a large-scale LP problem is Simplex method developed by George Dantzig in 1974 (Dantzig and Thapa, 2006). Simplex method converts a LP problem with n variables into a form of basic and nonbasic variables. Consider the problem in (5-3a) and rewrite the constraints in (5-3b) into

$$Gx = K \tag{5-4}$$

where $G \in \Re^{m\times n}$ with an assumption that G is a full rank matrix with $rank(G) = m$ and $m \leq n$. The problem in (5-3a) can be written as a combination of basic variables x_B and nonbasic variables x_N,

$$C^T x = C_B^T x_B + C_N^T x_N \tag{5-5a}$$

$$Gx = [G_N \; G_B] \begin{bmatrix} x_N \\ x_B \end{bmatrix} = K \tag{5-5b}$$

$$G_N x_N + G_B x_B = K \tag{5-5c}$$

where $G_B \in \Re^{m\times m}$, $G_N \in \Re^{m\times(n-m)}$, $x_B \in \Re^{m}$ and $x_N \in \Re^{(n-m)}$.

The impact of non-basic variables x_N on the objective function is assessed at every iteration. A basic variable x_B is replaced (called leaving basic variable) by a nonbasic variable (called entering basic variable) iteratively until an optimum solution is obtained. The leaving basic variable is decided by measuring a reduced cost r_c expressed as

$$r_c = (C_N - G_N^T G_B^{-1} C_B). \tag{5-6}$$

While, a minimum ratio test (MRT) that used to select an entering basic variable is described as

$$MRT = \frac{\widehat{K}_i}{-d_i} \qquad i \in B \tag{5-7a}$$

$$\widehat{K}_\iota = G_B^{-1} K \qquad d_i = G_B^{-1} g_e \tag{5-7b}$$

where g_e is a column of G that corresponds to the selected entering basic variable. MRT only takes $d_i < 0$ into consideration. If there is no $d_i < 0$, it means that the problem in (5-3) is unbounded.

Assuming an initial feasible solution is available, the procedure of Simplex method can be summarized as (Vanderbei, 2015)

Step 1 Compute the reduced cost r_c using (5-6) and select a nonbasic variable x_N that has most negative value as the entering basic variable and go to **Step 2**.

If the computed reduced cost $r_c \geq 0$, it means that an optimum condition is reached. Thus, the procedure is terminated.

Step 2 Compute the MRT using (5-7) and select a basic variable x_B with the smallest ratio as the leaving basic variable. Then, go to **Step 1**.

It is understood that Simplex method has a similar concept with Active Set method which is discussed in Section 5.9.3. This indicates the importance of understanding the Simplex method. In comparison, it can be said that basic variables represent active constraints that the current basic feasible solution intersects while, non-active constraints are represented by non-basic variables. At each iteration, Simplex method migrates from a current feasible solution to an improved adjacent feasible solution. This is achieved by sliding along the constraint boundaries. When the current feasible solution leaves one of its intersected constraints, it has to eliminate the constraint which is represented by a leaving basic variable. Next, an entering basic variable that represents the new constraint, which the current feasible solution migrates to, is added. Finally, the algorithm stops when there is no improvement of feasible solution can be found. The use of slack variables indicates the active constraints and it enables Simplex method to move along the constraints boundaries to search for an improved feasible solution. Based on author's experience, solve a simple LP problem using Simplex method with sketching provides a good understanding of the concept.

From the procedure above, it can be concluded that Simplex method is a version of Active Set method that tackles a LP problem despite their implementation methods are different. For a small-scale problem, Simplex method can be formulated and solved in a Tableau format (Griva et al., 2009). Although Simplex method is effective, there is degeneracy which causes a cycling phenomenon (not covered in this work), which prevents the algorithm to terminate by repeatedly visiting a number of feasible solutions. This can be avoided using Bland's rule (Bland, 1977).

5.6.2 Phase 1 Problem

The Simplex method described in previous section assumes that a selected initial basic solution is feasible. However, this is only applicable for a 'all slack variable basic' case where a LP problem in (5-2a) is of a form with the equality constraints in (5-2b) with nonnegative values on its right hand side, which allows all slack variables to be employed as basic variables x_B. Other than this case, Simplex method is usually implemented in two phases as it requires an initial basic feasible solution to commence (Guenin et al., 2014). To achieve this, the first phase is to formulate the constraints as an independent LP problem and it can be solved using Simplex method; thus, it is called Phase 1 problem. In other words, one has to solve two LP problems. Phase 1 forms a fundamental problem to find an initial feasible solution for optimization algorithms including Active Set method. In short, Phase 1 is to find a solution that satisfies all the constraints.

The strategy is to isolate each constraint with at least one basic variable with a +1 coefficient. This can be achieved by adding an artificial variable a_i to the following constraints: 1) Equality constraints without a slack variable; 2) ≥ inequality constraints with surplus variables. Then, the problem in Phase 1 is expressed as (Guenin et al., 2014)

$$minimize\ \ a_1 + a_2 + a_3 + \cdots + a_n \tag{5-8}$$

where n is the total number of artificial variables. If the solution in (5-8) is $a_1 = a_2 = \cdots = a_n \neq 0$, then the original LP problem is not feasible. This is because all artificial variables are nonnegative, which means that the summation in (5-8) is only zero if the artificial variables are all zero. Phase 1 problem can be integrated into Simplex method using the Tableau format. After an initial feasible solution is provided from Phase 1, the procedure proceeds to Phase 2 with the original LP problem.

5.7 Lagrange Method

Lagrange multiplier λ is a factor which describes the relationship between an optimization problem and gradients of its constraints. In other words, the tangent of an equality constraint (or an active inequality constraint) is parallel to the tangent of the level curve of an objective function for an optimum solution (Griva et al., 2009).

5.7.1 Problem with Equality Constraints

Lagrange multiplier λ is firstly used in handling equality constraints. Consider a general problem with two variables x_1, x_2

$$\max_{x_1,x_2} f(x_1, x_2) \tag{5-9a}$$

subject to an equality constraint,

$$e(x_1, x_2) = l. \tag{5-9b}$$

The problem in (5-9) can be interpreted as two gradient vectors are parallel if and only if there is a Lagrange multiplier λ such that

$$\nabla f(x_1^*, x_2^*) = \lambda \nabla e(x_1^*, x_2^*) \tag{5-10}$$

where $\nabla f(x)$ and $\nabla e(x)$ are the gradients of the problem and the constraint, respectively; (x_1^*, x_2^*) is an optimal point. Based on this, the formulation of Lagrange method is described as

$$\mathcal{L}(x_1, x_2, \lambda) = f(x_1, x_2) - \lambda(e(x_1, x_2) - l) \tag{5-11a}$$

with the necessary conditions for $(x_1^*, x_2^*, \lambda^*)$ being an optimum solution:

$$\frac{\partial \mathcal{L}}{\partial x_1}(x_1^*, x_2^*, \lambda^*) = \frac{\partial f}{\partial x_1}(x_1^*, x_2^*) - \lambda \frac{\partial e}{\partial x_1}(x_1^*, x_2^*) = 0 \tag{5-11b}$$

$$\frac{\partial \mathcal{L}}{\partial x_2}(x_1^*, x_2^*, \lambda^*) = \frac{\partial f}{\partial x_2}(x_1^*, x_2^*) - \lambda \frac{\partial e}{\partial x_2}(x_1^*, x_2^*) = 0 \tag{5-11c}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda}(x_1^*, x_2^*, \lambda^*) = e(x^*, y^*) - l = 0. \tag{5-11d}$$

It is noticeable that in (5-11b) and (5-11c), Lagrange multiplier λ describes the relationship between the gradients of the problem and the gradients of constraints with respect to each variable. The equation in (5-11d) is to satisfy the equality constraint. This technique is called Lagrange method which converts a constrained optimization problem into an unconstrained one, with an important condition $\nabla \mathrm{e}(x_1^*, x_2^*) \neq 0$.

5.7.2 Problem with Inequality Constraints

The Lagrange method can be extended for the case of inequality constraints. Consider the problem in (5-9a) subject to an inequality constraint,

$$h(x_1, x_2) \leq b \tag{5-12}$$

and, there exist two possible situations at an optimal point x_1^*, x_2^*:

1) $h(x_1^*, x_2^*) - b = 0$ – Binding Case: $\nabla f = \lambda \nabla h$, $\lambda^* \geq 0$
2) $h(x_1^*, x_2^*) < b$ – Non-binding Case: $\nabla f = 0$, $\lambda^* = 0$

Using the non-binding case, it is possible to stipulate $\lambda^* = 0$. Therefore, this yields two possible outcomes: 1) $\lambda^* = 0$, 2) $h(x_1^*, x_2^*) - b = 0$. This condition is called complementary slackness. Reconsider the problem in (5-9a) subject to an inequality constraint in (5-12), the Lagrange method can be written as

$$\mathcal{L}(x_1, x_2, \lambda) = f(x_1, x_2) - \lambda(h(x_1, x_2) - b) \tag{5-13a}$$

with the necessary conditions:

$$\frac{\partial \mathcal{L}}{\partial x_1}(x_1^*, x_2^*, \lambda^*) = 0 \tag{5-13b}$$

$$\frac{\partial \mathcal{L}}{\partial x_2}(x_1^*, x_2^*, \lambda^*) = 0 \tag{5-13c}$$

$$\lambda^*[h(x_1^*, x_2^*) - b] = 0 \qquad \text{(5-13d)}$$

$$\lambda^* \geq 0 \qquad \text{(5-13e)}$$

$$h(x_1^*, x_2^*) \leq b \qquad \text{(5-13f)}$$

Equation (5-13d) expresses the complementary slackness condition, while the nonnegative condition of the Lagrange multiplier is described in (5-13e). In order to satisfy the inequality constraints, equation in (5-13f) is included. This form of Lagrange method is also used to check the optimum condition for a QP problem in later section.

5.7.3 Procedure and Remarks

The importance of Lagrange method will be seen in later sections (QP and NLP) where it is used to augment an optimization problem. A Lagrange method converts an original optimization problem using the Lagrange function in (5.11a) and (5.13a) with respect to the constraints. Then, compute the gradients using partial derivatives and set them to zero. Finally, solve the formulated problem with the necessary conditions as in (5-11b - 5.11d) and (5-13b – 5.13f), respectively.

Although the conditions for both equality and inequality constraints are derived based on an optimum solution, it is important to stress that solving the equations for both conditions with respect to the optimization problem only yields a stationary point. It is also called critical point. This means that the Lagrange methods in first order form described in Section 5.7.1 and 5.7.2 only provide necessary conditions for an optimum solution. Another key point is the formulated problem may be non-linear, which means a numerical method such as Newton method is required.

It is necessary to state that both conditions for equality constraints in (5.11) and inequality constraints (5.13) can be merged for an optimization problem with both equality and inequality constraints, which forms the basis of Karush-Kuhn-Tucker conditions in next section.

5.8 Optimality Conditions

In general, optimality conditions are divided into necessary and sufficient conditions. Necessary conditions are satisfied by a local optimum solution while, satisfying sufficient conditions provides a local optimum solution. In this section, consider a general problem (Shevade, 2012)

$$\min f(x) \tag{5-14a}$$

subject to

$$h_j(x) \leq 0, \qquad j = 1, \dots, l \tag{5-14b}$$

$$e_i(x) = 0, \qquad j = 1, \dots, m \tag{5-14c}$$

$$x \in \mathbb{R}^n$$

$$\text{Let } X = \left\{x: h_j(x) \leq 0, e_i(x) = 0, j = 1, \dots, l; i = 1, \dots, m\right\} \tag{5-14d}$$

and active constraints are denoted as

$$A(x) = \left\{j: h_j(x) = 0\right\}. \tag{5-14e}$$

5.8.1 Regular Points

Regular points for inequality and equality constraints are essential in Karush-Kuhn-Tucker conditions (Chachuat, 2007, Shevade, 2012). Consider the problem in (5-14a) with inequality constraints in (5-14b), a point $x^* \in X$ is a regular point if the gradient vectors of inequality constraints $\nabla h_j(x^*)$, $j \in A(x^*)$ (active constraints) are linearly independent. For an equality constraints case, consider the problem in (5-14a) with equality constraints in (5-14c), a point $\bar{x} \in X$ is said to be a regular point if gradient vectors of equality constraints $\nabla e_i(\bar{x})$ are linearly independent.

5.8.2 Karush-Kuhn-Tucker Conditions

Karush-Kuhn-Tucker conditions (KKT) are deemed one of greatest development as it is used as the basis for many optimization algorithms of LP, QP or NLP. It was claimed that the conditions were firstly derived by William Karush in 1939 but it was not published and Harold W. Kuhn and Albert W. Tucker derived the conditions independently and published it in 1951 (Cottle, 2012). KKT conditions are computed to solve an optimization problem with inequality constraints using Lagrange method as described in Section 5.7.2. Furthermore, KKT conditions are used to identify a critical point for constrained QP and NLP; this type of critical point is called KKT point. In general, there are two categories of KKT conditions: 1) First order conditions; 2) Second order conditions. Readers are referred to (Boyd and Vandenberghe, 2004, Shevade, 2012) for more details where the materials are obtained for the following sections.

5.8.2.1 KKT First Order Necessary Conditions

The KKT first order necessary conditions are the generalised form of the extended Lagrange method with active inequality constraints in Section 5.7.2. Consider the problem in (5-14), if $x^* \in X$ is a local minimum and a regular point, then there exists a unique vector $\lambda^* \in (\lambda_1^*, \dots, \lambda_l^*)^T$ such that

$$
\begin{aligned}
&\nabla f(x) + \sum_{j=1}^{l} \lambda_j^* \nabla h_j(x^*) = 0 \\
&\lambda_j^* h_j(x^*) = 0 \qquad \forall j = 1, \dots, l \\
&\lambda_j^* \geq 0 \qquad\qquad \forall j = 1, \dots, l
\end{aligned}
\tag{5-15}
$$

- KKT point: $(x^*, \lambda^*), x^* \in X, \lambda^* \geq 0$
- Lagrange Function: $\mathcal{L}(x, \lambda) = f(x) + \sum_{j=1}^{l} \lambda_j \nabla h_j(x)$
- $\nabla \mathcal{L}_x(x^*, \lambda^*) = 0$
- λ_j: Lagrange multipliers, $\lambda_j \geq 0$ for inequality constraints

- $\lambda_j^* h_j(x^*) = 0$ Complementary slackness condition
- $\lambda_j^* = 0 \quad \forall j \notin A(x^*)$ Non-active constraints

For equality constraints, the condition can be stated as let $x^* \in X$ be a regular point and local minimum. Then $\exists\ \mu^* \in \mathbb{R}^m$ such that

$$\nabla f(x^*) + \sum_{i=1}^{m} \mu_i^* \nabla e_i(x^*) = 0. \tag{5-16}$$

5.8.2.2 Generalised Form of KKT First Order Necessary Conditions

By merging the conditions for equality and inequality constraints in previous section, a generalised form of KKT first order necessary condition can be stated as follows. Consider the problem in (5-14), if $x^* \in X$ is a local minimum and a regular point, then there exist unique vectors $\lambda^* \in \mathbb{R}^l$ and $\mu^* \in \mathbb{R}^m$ such that

$$\begin{gathered}\nabla_x \mathcal{L}(x^*, \lambda^*, \mu^*) = \nabla f(x) + \sum_{j=1}^{l} \lambda_j^* \nabla h_j(x^*) + \sum_{i=1}^{m} \mu_i^* \nabla e_i(x^*) \ = 0 \\ \lambda_j^* h_j(x^*) = 0 \quad \forall j = 1, \dots, l \\ \lambda_j^* \geq 0 \qquad \forall j = 1, \dots, l\end{gathered} \tag{5-17}$$

- KKT point: $(x^*, \lambda^*, \mu^*), x^* \in X, \lambda^* \geq 0, \mu^* \in \mathbb{R}^m$ satisfying above conditions.

5.8.2.3 KKT Second Order Necessary Conditions

Based on the Lagrange function in (5-17), if $x^* \in X$ is a local minimum and a regular point, then there exists unique vectors $\lambda^* \in \mathbb{R}^l$ and $\mu^* \in \mathbb{R}^m$ such that

$$\nabla_x \mathcal{L}(x^*, \lambda^*, \mu^*) = 0$$

$$\lambda_j^* h_j(x^*) = 0 \qquad \forall j = 1, \dots, l \tag{5-18a}$$

$$\lambda_j^* \geq 0 \qquad \forall j = 1, \dots, l$$

and the hessian,

$$d^T \nabla_x^2 \mathcal{L}(x^*, \lambda^*, \mu^*) d \geq 0 \quad \text{Positive Definite} \tag{5-18b}$$

for all $d \ni \nabla h_j(x^*)^T d \leq 0,\ j \in I$ and $\nabla e_i(x^*)^T d = 0, i \in \varepsilon$, where I and ε represent the index of active constraints for $h_j(x^*)$ and $e_i(x^*)$, respectively.

5.8.2.4 KKT Second Order Sufficient Conditions

If there exist $x^* \in X$, $\lambda^* \in \mathbb{R}^l$ and $\mu^* \in \mathbb{R}^m$ such that

$$\nabla_x \mathcal{L}(x^*, \lambda^*, \mu^*) = 0$$

$$\lambda_j^* h_j(x^*) = 0 \qquad \forall j = 1, \dots, l \tag{5-19a}$$

$$\lambda_j^* \geq 0 \qquad \forall j = 1, \dots, l$$

and

$$d^T \nabla_x^2 \mathcal{L}(x^*, \lambda^*, \mu^*) d \geq 0 \quad \text{Positive Definite} \tag{5-19b}$$

for all $d \neq 0$ such that

$$\nabla h_j(x^*)^T d = 0, j \in I \text{ and } \lambda_j^* > 0$$

$$\nabla h_j(x^*)^T d \leq 0, j \in I \text{ and } \lambda_j^* = 0 \tag{5-19c}$$

$$\nabla e_j(x^*)^T d \leq 0, i \in \varepsilon$$

Then, x^* is a strict local minimum of NLP.

5.9 Quadratic Programming

QP is regarded as a special case of NLP. A QP problem consists of a quadratic objective function subject to linear equality and inequality constraints. It is an easier problem to solve compared to NLP. More importantly, it provides the basis for Sequential Quadratic Programming to solve NLP problems. In an ideal case of QP, a global optimum solution is guaranteed due to its convex or concave shapes. One of the advantages of a QP problem is its characteristic can be determined by examining the definiteness of its hessian matrix.

5.9.1 Convex Function

A convex function is included to show its relationship to QP. For details information on its background, readers are referred to see (Boyd and Vandenberghe, 2004) which includes line function, affine set and convex set. A convex function that governed by a line segment equation is described as

$$f(\theta x_1 + (1-\theta)x_2) \leq \theta f(x_1) + (1-\theta)f(x_2) \qquad \theta \in [0,1] \tag{5-20}$$

where $x_1, x_2 \in \mathbb{R}^m$. From (5-20), it is expressed as a line, which joins the two points $f(x_1)$ and $f(x_2)$ (right hand side) within the function, is always greater than a line joint by x_1 and x_2 (left hand side). This condition implies that the function is a convex shape as illustrated in Fig. 5-3. This property guarantees two important conditions with $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*) \geq 0$.

The form of a quadratic optimization problem is as below,

$$\min_x f(x) = \frac{1}{2}x^T H x + c^T x \tag{5-21a}$$

subject to linear inequality constraints

$$hx \leq b \tag{5-21b}$$

where $H = H^T \geq 0$ which is a symmetric positive definite matrix. The definiteness of matrix can be found by evaluating its eigenvalues.

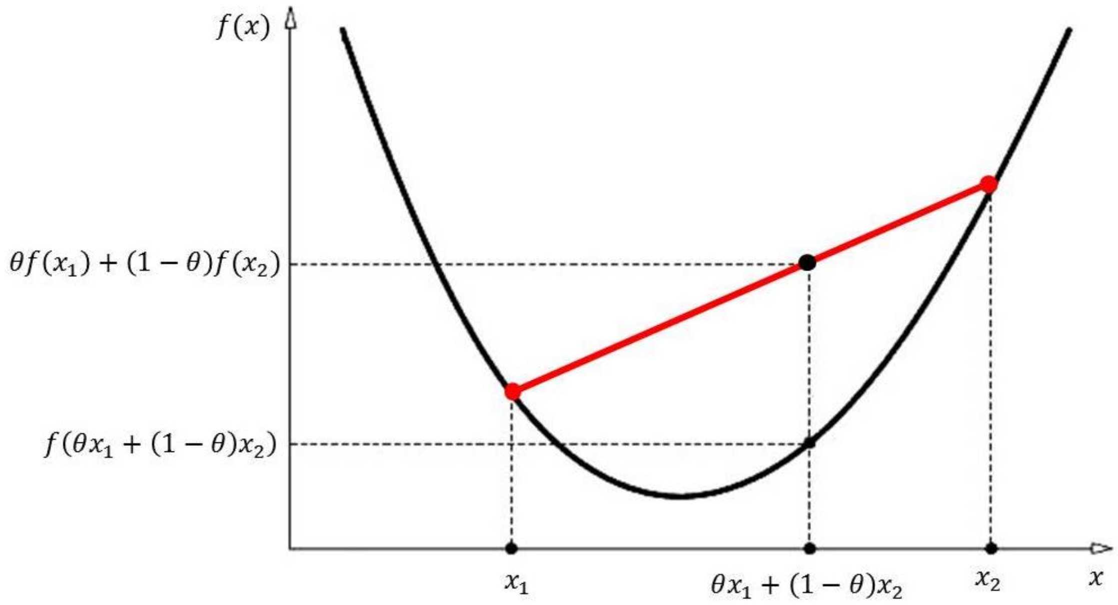
$f(x)$
$\theta f(x_1) + (1-\theta)f(x_2)$
$f(\theta x_1 + (1-\theta)x_2)$
x_1
$\theta x_1 + (1-\theta)x_2$
x_2
x

Fig. 5-3 Convex function.

5.9.2 Unconstrained Problem

Understanding an unconstrained QP problem is important as it forms the optimality conditions for constrained problems. By observing the function in (5-21a), if the hessian H is not positive semi-definite, $x^T Hx < 0$, then $f(x)$ approaches to a negative infinity, which means no minimum point is available (refer to KKT second order necessary conditions in Section 5.8.2.3). To solve an unconstrained quadratic problem in (5-21a), computes the gradient and takes it equals to zero which yields an local minimum $x^* = -H^{-1}c$. A global minimum is guaranteed for a positive definite H case. This implies that it has an unique stationary point x^*. On the hand, in a semi-positive definite H case, the formulated problem might be inconsistent. Therefore, a solution is not guaranteed.

5.9.3 Constrained Problem – Active Set Method

For a constrained problem with equality constraints case, the hessian matrix H is only needed to be positive definite on the subspace to obtain a minimum point. One of the popular methods to solve a constrained QP problem is Active Set method (ASM), which forms a foundation of step-wise line search method for Sequential Quadratic Program algorithm in the next section. Thus, the understanding of this method is of great importance. The development of ASM is based on the Lagrange Method in (5-11) which can be used to solve a constrained QP problem with equality constraints. In a case with inequality constraints, it can be formulated as an optimization problem with a series of equality constraints that represent active inequality constraints using the KKT first order necessary conditions in (5-15); this is also a binding case as described in Section 5.7.2.

A constraint is said to be active if a current feasible solution binds to the constraint boundary. The concept of ASM is to search an optimum solution by minimizing an objective function iteratively while remaining on the boundaries of constraints. An initial feasible solution is required and this can be achieved by solving the Phase 1 problem that described in Section 5.6.2. Given an initial feasible solution to commence, the first step is to search for a descent direction d_k which minimizes the objective function while keeping the current solution remains on constraint boundaries. A basis Z_k that satisfies $A_k Z_k = 0$ is computed from the QR decomposition of an active set $A_k^T \in \mathbb{R}^{m \times n}$ (Kalpesh, 2004),

$$\begin{gathered} qr(A_k^T) = \left[\hat{Q}\ Q_N\right]\begin{bmatrix} R \\ 0 \end{bmatrix} \\ Q_N = Z_k \end{gathered} \tag{5-22}$$

This makes Z_k provides a basis that corresponding to the current active constraints. Thus, d_k can be determined as a linear combination of some vector p in the $range(Z_k)$, which yields $d_k = Z_k^T p$. The vector p is computed by solving an unconstrained quadratic optimization with respect to p,

$$\begin{gathered} q(p) = \frac{1}{2} p^T Z_k^T H Z_k p + c^T Z_k p \\ p = -(Z_k^T H Z_k) \backslash (Z_k^T c). \end{gathered} \tag{5-23}$$

After d_k is found, next is to determine a maximum distance of which the current solution x_k allowed to travel on the constraints boundaries without violating the constraints; this distance is defined as a step length α. The next feasible solution is expressed as

$$x_{k+1} = x_k + \alpha d_k \tag{5-24}$$

If a unit length α along d_k can be taken while remaining feasible, then this is the ultimate optimum solution to the QP problem. Otherwise, one has to calculate a α that current feasible solution is allowed to travel to nearest non-active constraints bindingly,

$$\begin{gathered} A_i(x_k + \alpha d_k) \leq b_i \qquad i = 1, \dots, m \\ \alpha = \min_{i=1,\dots m} \frac{-(A_i x_k - b_i)}{A_i d_k}. \end{gathered} \tag{5-25}$$

The following step is to compute the Lagrange multipliers λ_k provided the linear equations are non-singular,

$$\begin{gathered} A_k^T \lambda_k = c \\ \lambda_k = (-R \backslash (Qc)). \end{gathered} \tag{5-26}$$

If all components of $\lambda_k > 0$, it means that the optimality in the KKT first order necessary conditions in (5-15) are satisfied, implying that an ultimate optimum solution is achieved. Otherwise, a constraint that corresponds to the most negative Lagrange multiplier λ_k (if there is more than one negative element), which the current solution is set to leave, is removed from the active set A_k. After this, the algorithm proceeds to next iteration. The procedure is illustrated in Fig. 5-4. Alternatively, for a small-scale problem, the ASM method can be executed in solving a set of linear equation (non-singular) formulated by KKT first order necessary conditions as described in Section 5.8.2.1.

It is noticeable that ASM adopts a similar concept with Simplex method as ASM also tracks an optimum solution along constraint boundaries iteratively. The difference is ASM has a step length α in computing the next feasible solution x_{k+1} because the optimum solution for a QP problem is not necessary a corner point. ASM plays an important role in Sequential Quadratic Programming (SQP) algorithm for NLP which is discussed in next section.

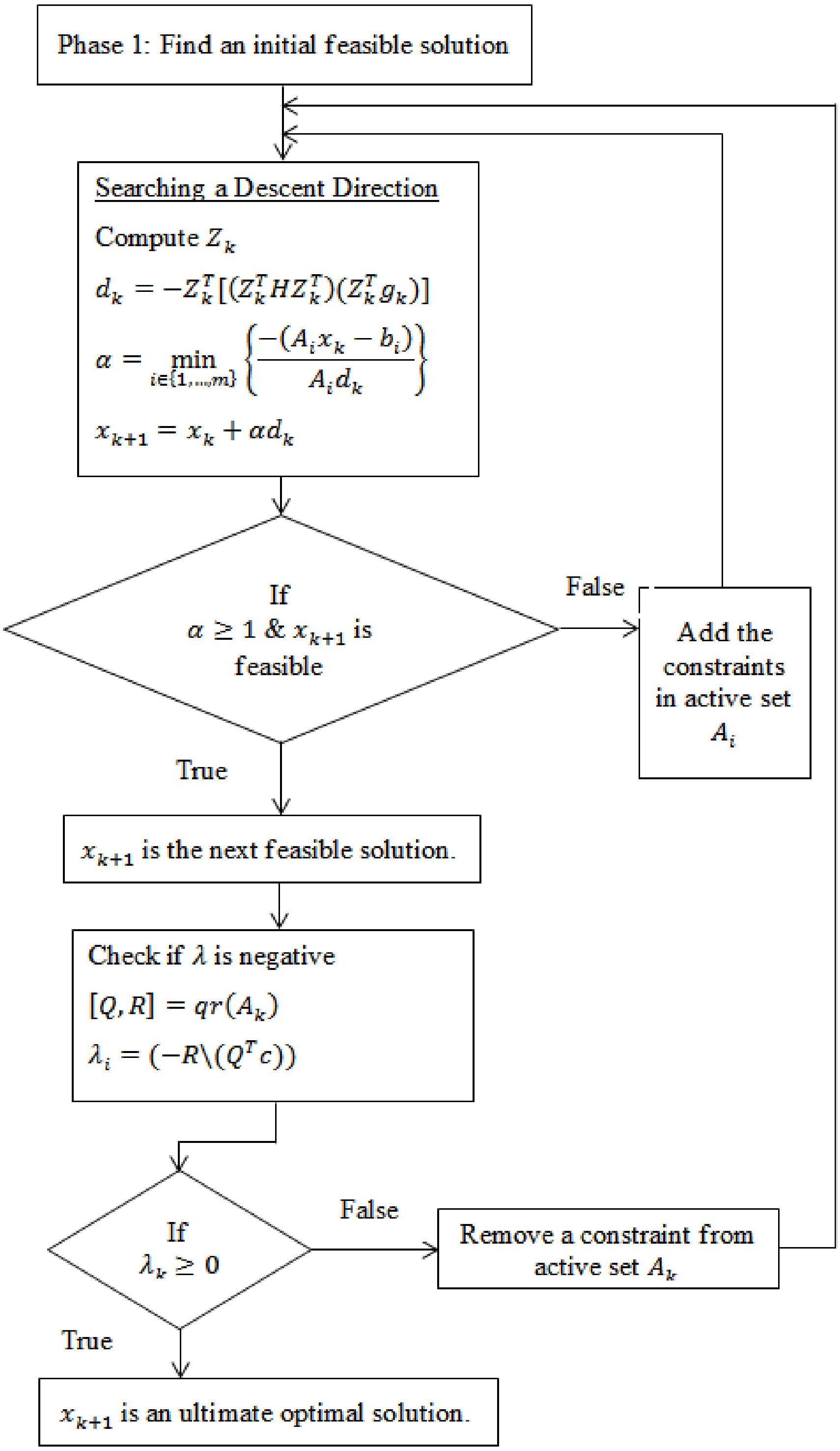
Phase 1: Find an initial feasible solution
Searching a Descent Direction
Compute Z_k
$d_k = -Z_k^T[(Z_k^T H Z_k^T)(Z_k^T g_k)]$
$\alpha = \min_{i \in \{1,\ldots,m\}} \left\{ \frac{-(A_i x_k - b_i)}{A_i d_k} \right\}$
$x_{k+1} = x_k + \alpha d_k$
If
$\alpha \geq 1$ & x_{k+1} is
feasible
False
Add the constraints in active set A_i
True
x_{k+1} is the next feasible solution.
Check if λ is negative
$[Q, R] = qr(A_k)$
$\lambda_i = (-R \backslash (Q^T c))$
If
$\lambda_k \geq 0$
False
Remove a constraint from active set A_k
True
x_{k+1} is an ultimate optimal solution.

Fig. 5-4 Flow chart of Active Set method.

5.10 Non-linear Programming

The difficulties in mathematical optimization arise when a non-linear problem is considered which is called Non-linear Programming (NLP). Non-linear characteristics have led to few severe consequences that summarized in (Chinneck, 2006). Limited information of the problem has restricted the recognition of a global optimum. Instead, a local optimum solution can be recognised as the information of the problem at current point is available for assessment. Dissimilar from LP and QP, optimum solutions of NLP problems are not restricted to corner points or constraint boundaries, increasing the difficulties in exploring the optimum solutions. Moreover, disconnected feasible regions may be encountered for problems that encounter severe non-linearity. These consequences have shown that the uncertainties in NLP problems are very high, leading to the employment of numerical methods. It is understood constraints help narrow the space of tracking an optimum solution in some cases, resulting in a shorter time frame needed in solving the problem compared to that of an unconstrained problem.

5.10.1 Unconstrained NLP

Similar to LP and QP, the understanding of unconstrained problems in NLP is equally important. Line search methods are the fundamental concept for both unconstrained and constrained NLP problems. A simple case for an unconstrained NLP problem with single variable dimension can be solved using Bisection Search method (BS) which a local optimum solution is searched along a line using the first derivatives information of the problem. Thus, in order to use BS method, the problem has to be differentiable.

For a multi-dimension problem, the concept of BS method is extended into methods called steepest descent (minimization) and steepest ascent (maximization); their difference with BS method is in the derivatives information where a gradient vector (first derivative) and hessian matrix (second derivative) are needed. The definiteness of a hessian matrix, which is determined by its determinant, is used to decide the characteristic of a stationary point. A step length along a descent direction is employed in both steepest descent and steepest ascent methods. In order to provide a clear relationship between unconstrained algorithms, their descent directions are briefly discussed. A descent direction $d_k \in \mathbb{R}^n$, which defines a direction that minimizes an objective function at every iteration $k,$ is expressed as

$$g_k^T d_k < 0 \tag{5-27}$$

where g_k is a gradient of the objective function at every iteration k. Equation in (5-27) requires an angle to be maintained between the gradient g_k and descent direction d_k to achieve a decrease in the objective function, as illustrated in Fig. 5-5. For global convergence theorem, readers are referred to (Yuan, 1999). Different optimization algorithms compute the descent directions differently. Every algorithm has very a distinctive characteristic. The discussion concentrates on the features and relationships between every technique (Shevade, 2012), as summarized in Fig. 5-6.

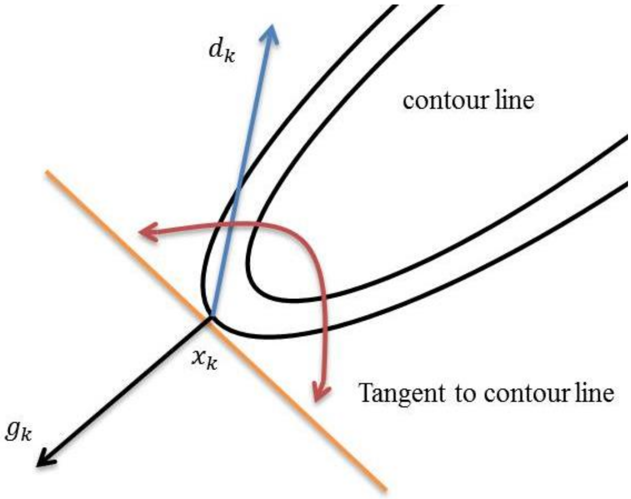

Fig. 5-5 Descent direction.

For steepest descent method (SD), its objective is to obtain a decrease of objective function maximally with $d_k = -g_k$. SD method is computationally cheap but it has a poor convergence rate except for a circular convex problem where an optimum solution can be achieved in one iteration. In comparison, classical Newton method has a better convergence rate especially for a convex problem regardless of its shape but it is computationally expensive as it has to compute an inverse of hessian matrix H^{-1} for its descent direction $d_k = -(H_k)^{-1} g_k$. Moreover, it is sensitive to the initial starting point for non-quadratic problems as its convergence is guaranteed only if a starting point is sufficiently close to the optimum solution. The drawback of both SD and classical Newton methods is they are positive definite hessian matrix based.

For non-positive definite problems, a modified Newton method is with a descent direction $d_k = -(H_k + \zeta_k I)^{-1} g_k$ is introduced. Its strategy is to modify the hessian matrix to achieve positive definiteness using an additional term ζ_k. A small ζ_k means that it has a good approximation of the positive definite hessian matrix. However, if ζ_k is large and becomes dominant, the algorithm may fail to converge. The main drawback of both classical and modified Newton methods is the requirement in computing the hessian matrix which may not be available in some cases (Ding et al., 2010).

Another advanced method that based on the Newton method, Quasi-Newton method, approximates the inversion of a hessian matrix B_k using rank-one correction method. This helps reduce the computational requirement. The resulting descent direction is $d_k = -B_k g_k$. The drawback of this method is the positive definiteness of a hessian matrix is not guaranteed at every iteration, which leads to improved methods such as Davidon-Fletcher-Powell (DFP) and Broyden-Fletcher-Goldfarb-Shannon (BFGS) methods. The difference between both methods is DFP method updates B_k while BFGS method updates $G_k = B_k^{-1}$. The update of DFP method is achieved by satisfying Quasi-Newton condition (secant condition),

$$\begin{gathered} B_{k+1}\gamma_k = \delta_k \\ g_{k+1} - g_k = \gamma_k, \qquad x_{k+1} - x_k = \delta_k. \end{gathered} \tag{5-28}$$

While, BFGS method is to satisfy (Dai, 2013)

$$G_{k+1}\delta_k = \gamma_k. \tag{5-29}$$

Among two methods, BFGS method is claimed to be more efficient as it has a self-correcting properties (Ding et al., 2010).

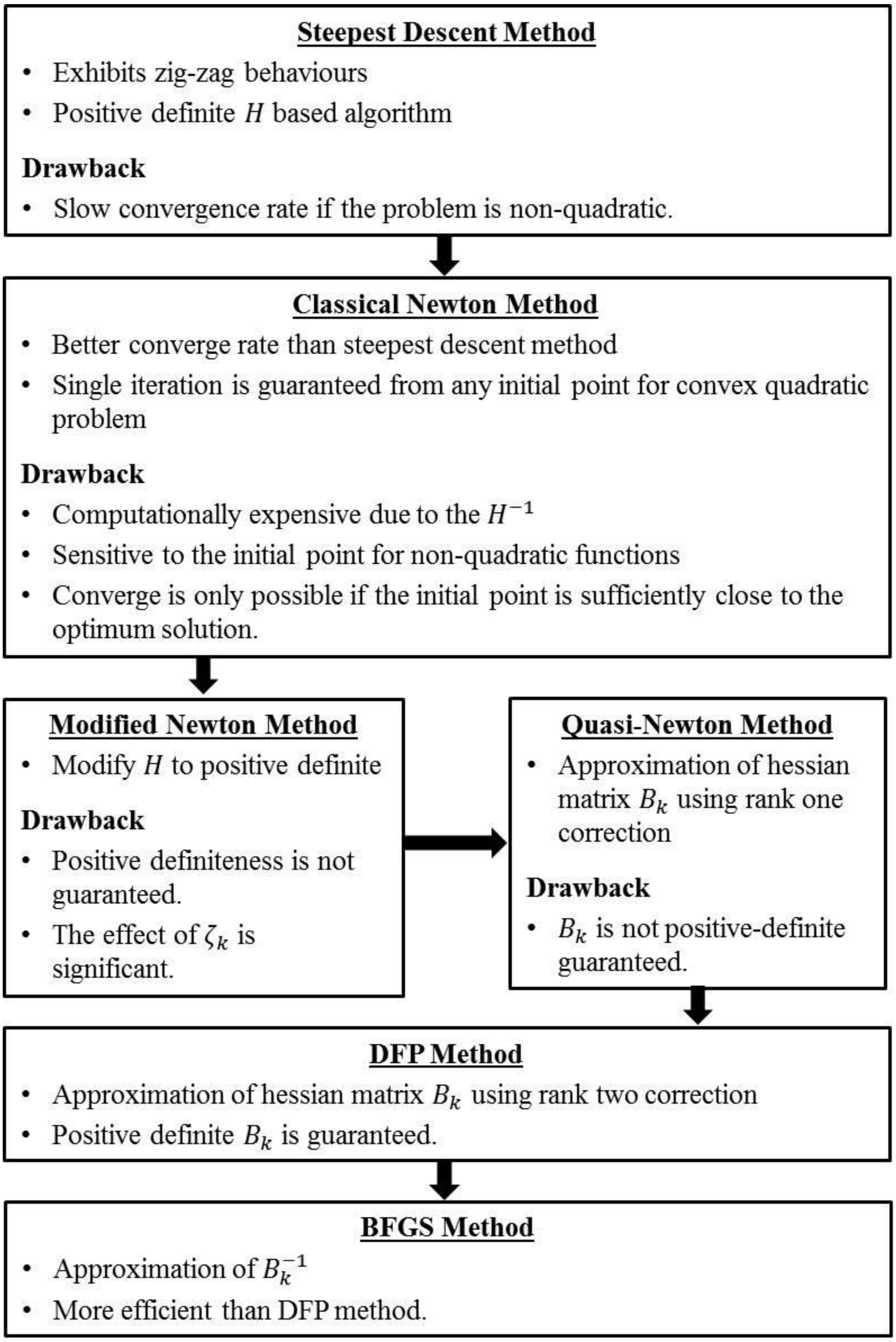
Steepest Descent Method
• Exhibits zig-zag behaviours
• Positive definite H based algorithm
Drawback
• Slow convergence rate if the problem is non-quadratic.
Classical Newton Method
• Better converge rate than steepest descent method
• Single iteration is guaranteed from any initial point for convex quadratic problem
Drawback
• Computationally expensive due to the H^{-1}
• Sensitive to the initial point for non-quadratic functions
• Converge is only possible if the initial point is sufficiently close to the optimum solution.
Modified Newton Method
• Modify H to positive definite
Drawback
• Positive definiteness is not guaranteed.
• The effect of ζ_k is significant.
Quasi-Newton Method
• Approximation of hessian matrix B_k using rank one correction
Drawback
• B_k is not positive-definite guaranteed.
DFP Method
• Approximation of hessian matrix B_k using rank two correction
• Positive definite B_k is guaranteed.
BFGS Method
• Approximation of B_k^{-1}
• More efficient than DFP method.

Fig. 5-6 Unconstrained NLP methods.

5.10.2 Constrained NLP: Penalty Method

Penalty method employs a penalty function to assess the infeasibility of an optimization problem where unsatisfied constraints are penalised with a cost. This method initiates from an infeasible solution, then reduces infeasibilities and terminates with feasible solutions eventually provided that convergence is achieved. This means the optimum and feasible solutions are computed simultaneously. Due to its feature of driving solutions into feasibility, it can be used to find a feasible starting point for NLP algorithms. Consider a NLP problem

$$\min f(x) \tag{5-30a}$$

subject to

$$h_j(x) \leq 0, \;\; j = 1, \dots, l \tag{5-30b}$$

$$e_i(x) = 0, \;\; j = 1, \dots, m \tag{5-30c}$$

Then, converts a constrained problem into an unconstrained problem by adding a penalty function to the objective function in (5-30a) (Di Pillo and Grippo, 1989),

$$\min \{f(x) + cP(x)\} \tag{5-31}$$

where c is a positive parameter. P is a function that penalises unsatisfied constraints which can be defined as

$$P(x) = \frac{1}{2}\sum_{j=1}^{l}\left[\max(0, h_j(x)\right]^2 + \frac{1}{2}\sum_{i=1}^{m} e_i^2(x) \tag{5-32}$$

and the augmented objective function can be rewritten as

$$q(x, c) = f(x) + cP(x). \tag{5-33}$$

The magnitude of parameter c has an important effect. A suitably large c relative to $f(x)$ will have a heavy penalise cost for any constraint violation. Using this penalty conversion strategy, an unconstrained problem can be solved using unconstrained search methods iteratively with an assumption of c_k approaching infinity,

$$c_k > 0 \text{ and } c_{k+1} > c_k, \quad c_k = 1,2, \dots, \infty. \tag{5-34}$$

With this assumption, the convergence is as below

$$x_k \rightarrow x^* \text{ as } c_k \rightarrow +\infty. \tag{5-35}$$

It is recommended to start c with a small value in order to ensure a convergence is achieved and a sequential unconstrained minimization technique can be used to implement the Penalty method (Bryan and Shibberu, 2005).

On the contrary, Barrier method is employed for infeasibility prevention which is useful when the search for an optimum solution is conducted in feasible regions. Barrier method penalises constraints that the current solution moves into by imposing a barrier function to prevent current solution to go infeasible. Similar to the Penalty method, Barrier method also transforms a constrained problem into an unconstrained problem (Byrne, 2013). However, this method is proved to be effective in dealing with problems with inequality constraints only (Chinneck, 2006).

5.10.3 Sequential Quadratic Programing

Sequential quadratic programming (SQP) is one of the effective techniques to solve a constrained NLP problem but it is also regarded as one of high complexity algorithms. SQP is based on a line search method that founded by other optimization algorithms such as the ASM described in Section 5.9.3 (Leyffer and Mahajan, 2010). The difference in SQP is it employs linear approximations to the objective function and constraints due to the non-linear characteristics of the problem. BFGS method is employed to approximate the positive definite Hessian matrix. Consider the general NLP problem in (5-30) augmented with the Lagrange function in (5-17), a quadratic approximation of the objective function is described as (Boggs and Tolle, 1995)

$$q_k(d) = \nabla f(x_k)^T d + \frac{1}{2} d^T \nabla_{xx}^2 \mathcal{L}(x_k, \lambda_k) d \tag{5-36a}$$

and the linear approximation of constrains

$$\nabla e_i(x_k)^T d + e_i(x_k) = 0, \qquad i = 1, \dots, m_e \tag{5-36b}$$

$$\nabla h_i(x_k)^T d + h_i(x_k) \leq 0, \qquad i = m_e + 1, \dots, m. \tag{5-36c}$$

Notice that μ^* in (5-17) is integrated into λ_k using the notation $i = 1, \ldots m_e$ in (5-36b). The linearized QP problem is solved using ASM described in Section 5.9.3 to obtain a descent direction d_k for next iterate

$$x_{k+1} = x_k + \alpha_k d_k. \tag{5-37}$$

One of the main stages in SQP approach is to approximate the hessian matrix $\nabla_{xx}^2 \mathcal{L}(x_k, \lambda_k)$. An approximation of the hessian matrix B_k is updated using BFGS method at every iteration

$$B_{k+1} = B_k + \frac{\gamma_k \gamma_k^T}{\gamma_k^T \delta_k} - \frac{B_k \delta_k \delta_k^T B_k}{\delta_k^T B_k \delta_k} \tag{5-38}$$

$$\delta_k = x_{k+1} - x_k, \qquad \gamma_k = \nabla_x \mathcal{L}(x_{k+1}, \lambda_k) - \nabla \mathcal{L}(x_k, \lambda_k).$$

However, the positive definiteness in B_k is not assured. When this happens, γ_k is modified so that $\gamma_k^T \delta_k > 0$ (Boggs and Tolle, 1995).

Another important step is to decide a step length α_k along a descent direction d_k from the quadratic problem. Different from the unconstrained algorithms, a taken step length α_k not only required to decrease the objective function value, but also attempts to satisfy the constraints. These two requirements can be achieved using the Penalty method,

$$P(x) = f(x) + \sum_{i=1}^{m_e} v_i \cdot g_i(x) + \sum_{i=1}^{m_e} v_i \max[0, g_i(x)] \tag{5-39}$$

where $v_i > 0$ are penalty parameters and the penalty parameter is set to

$$v_i = (v_{k+1})_i \max_i \left\{ \lambda_i, \frac{(v_k)_i + \lambda_i}{2} \right\}, \qquad i = 1, \ldots, m. \tag{5-40}$$

In order to implement the Penalty method, a sequential unconstrained minimization technique is recommended (Chinneck, 2006) where a series of formulated unconstrained problems is solved. From the procedure above, it shows that SQP algorithm involves a wide area of techniques which includes ASM, Quasi-Newton method, and Penalty method.

5.11 Selection of Optimization Algorithm

As discussed in previous sections, there are three general problems, namely LP, QP and NLP. Respective algorithms are used to tackle different optimization problems. In this section, Matlab functions for optimization algorithms based on Matlab 2009b are discussed. It is no doubt that LP is the easier problem to recognise and it can be solved using the popular Simplex method.

For constrained QP problems, the advantage of ASM is the feasibility is maintained throughout the procedure as it starts with a feasible solution by solving a Phase 1 problem. This is useful for the MPC because if ASM fails to terminate with an optimum solution, a feasible solution is still available. This is important for processes with strict safety operating regions. However, this feature can be disadvantageous at the same time as an initial feasible solution may not be easy to compute. Moreover, a QP problem in MPC only occurs when linear models such as state-space representation is used. ASM is described in *active-set quadprog* and *fmincon Active Set* as Matlab functions.

In this research, a RBF network is employed as the internal model for the MPC, leading to NMPC. An NLP algorithm, SQP, is used as the non-linear optimiser for MPC. The Matlab function of SQP algorithm that used in this work is *fmincon SQP* algorithm as in (Wang et al., 2006, Yu et al., 2006). The advantage of SQP is feasibility is not required, which means that there is no Phase 1 problem. This is because computing an initial feasible solution for a NLP problem may require huge efforts and it can be a very time-consuming procedure. This is one of the main reasons that SQP being employed in this work. As discussed previously, maintaining feasibility is beneficial to MPC but it is difficult to compute an optimum solution while remaining feasible due to the non-linear characteristic of NLP problems. However, with the receding horizon approach that adopted by MPC, the solution in SQP is used as a starting point for every subsequent iteration, providing a warm start approach which improves the searching of an optimum solution. SQP algorithm is more efficient in term of execution time for constrained problems because of the limits on feasibility regions as stated in Matlab support documentation. Thus, SQP algorithm is useful for constrained control problems in the thesis.

5.12 Summary

In brief, MPC adopts a receding horizon strategy to formulate a control objective with the consideration of constraints for inputs and outputs into an optimization problem. The main idea of the receding horizon approach is clearly described together with its objective function. The benefits of MPC are the process interactions and process constraints are addressed. The interaction effects between every tuning parameter have direct impacts on the control performance. The difficulties arise in solving the optimization problem, particularly for NMPC.

Three optimization algorithms, Simplex method, ASM and SQP, to tackle the problems of LP, QP and NLP, respectively, are discussed. Although every method has a different strategy, their concepts are related. For a large-scale problem, Simplex method is proved to be more effective than the exhaustive search method. The understanding of Simplex method helps to visualize a tracking procedure of an optimum solution. The discussion concludes that the tracking procedure of Simplex method has a similar concept to the ASM. Furthermore, the Phase 1 problem is equally important for Simplex method and ASM to find an initial feasible solution.

The concept underlying the Lagrange multiplier forms the basis for KKT conditions for constrained problems. KKT conditions are regarded as the generalised form of Lagrange method to check optimality conditions. Thus, the KKT conditions are important for the ASM and SQP. The formulation of the binding and non-binding case for inequality constraints plays a vital role in the ASM.

Convex function is the core of QP, which an optimum solution is guaranteed. An obvious similarity between Simplex method and ASM is the tracking process is executed by staying on constraints boundaries. The concluded difference between these two methods is ASM has a step length as an optimum solution of QP can lie at any section of constraint boundaries. ASM demonstrates a comprehensive line search mechanism which provides a good foundation for SQP algorithm.

SQP algorithm adopts a step-wise line search based strategy to solve NLP problems. The linearization yields a QP sub-problem to find a descent direction, which is solved using the ASM. In order to ensure the positive definiteness, BFGS method is employed to approximate the hessian matrix. Then, a penalty method is used to find a step length in relation to the computed descent direction, producing a solution for next iteration.

Lastly, the features of algorithms are briefly discussed in relation to the MPC and their respective Matlab functions are mentioned. The studies for optimization algorithms are constructed in the order based on author's experiences. The main purpose is to build a foundation of knowledge for optimization techniques that related to MPC.

Chapter 6

FACTORIZED RBF NETWORK MODEL

6.1 Introduction

NARX-RBF model is often used as an internal model in MPC (RBF-MPC) As reviewed in Section 2.3.4, the NARX-RBF model lacks of efficiency in long range predictions (Su and McAvoy, 1997, Bhartiya and Whiteley, 2001). Another drawback is that the network cannot be factorized according to the past and future information.

This chapter presents a new factorized f-step model-based RBF network (f-step RBF). The objective of this work is twofold. The first objective is to introduce a f-step predictor and implement it with a RBF network. Secondly, the novelty in this work is the factorization of the f-step RBF network. As a result, an explicit objective function of the MPC is derived to reduce the computing load in solving the on-line optimization problem. The proposed model is evaluated in term of the modelling performance, model compactness, and computational requirement using a continuous-stirred tank reactor (CSTR) plant.

Both the prediction performance and the computing time for long range prediction of the developed model are evaluated by comparing the results with that of two existing models. By applying to the MPC of the CSTR plant, the simulation results show that not only the developed model outperforms the other two models but more importantly, the developed model also uses a more compact structure. Secondly, the control performance and computational requirement in solving the optimization problem of the proposed network is verified by comparing it with existing control approaches. The proposed network is proved to be more computationally efficient while achieving a good control performance.

The chapter is presented in the following structure. Section 6.2 describes the development of f-step prediction model. Section 6.3 explains the training algorithm of f-step RBF model; it also presents the prediction performance of proposed f-step RBF model and the comparison with other two models. Section 6.4 describes the factorization of f-step RBF model. The evaluation of control performance and computational requirement of the proposed model-based MPC and comparison studies are presented in Section 6.5.

6.2 f-Step Prediction Model

A p-Step control (PSC) model was developed in (Bhartiya and Whiteley, 2001) and it was implemented with a RBF network. It was reported that the PSC model was capable of predicting future process outputs over a prediction horizon without requiring future unknown process outputs. However, the flaw of PSC model is that an unrealistic massive network structure is required in order to achieve a satisfactory prediction performance. This has incurred a big computational burden load to prevent its application in fast dynamic systems. Moreover, there is a limitation in selecting the network model that $n_u \geq 2$ is required. To address these problems, an improved model, f-step prediction model, is developed in this thesis and is implemented with a RBF network. In this section, the derivation of the f-step model is described.

In the applications of RBF networks (Wang et al., 2006, Yu and Yu, 2007), a continuous-time non-linear system is represented by a NARX model in (6-1).

$$y_k = f\left[y_{k-1}, \dots, y_{k-n_y}, u_{k-1}, \dots, u_{k-n_u}\right] + e_k \tag{6-1}$$

where $u \in \Re^m$ and $y \in \Re^n$ are the input and output vectors with n_u and n_y being the output and input orders respectively and e is the error. $f[*]$ is a vector-valued non-linear function. There are two types of predictor structure as depicted in Fig. 6-1, where a RBF network can be trained as an one-step-ahead (OSA) predictor or a multistep-ahead (MSA) predictor (also see Section 3.3).

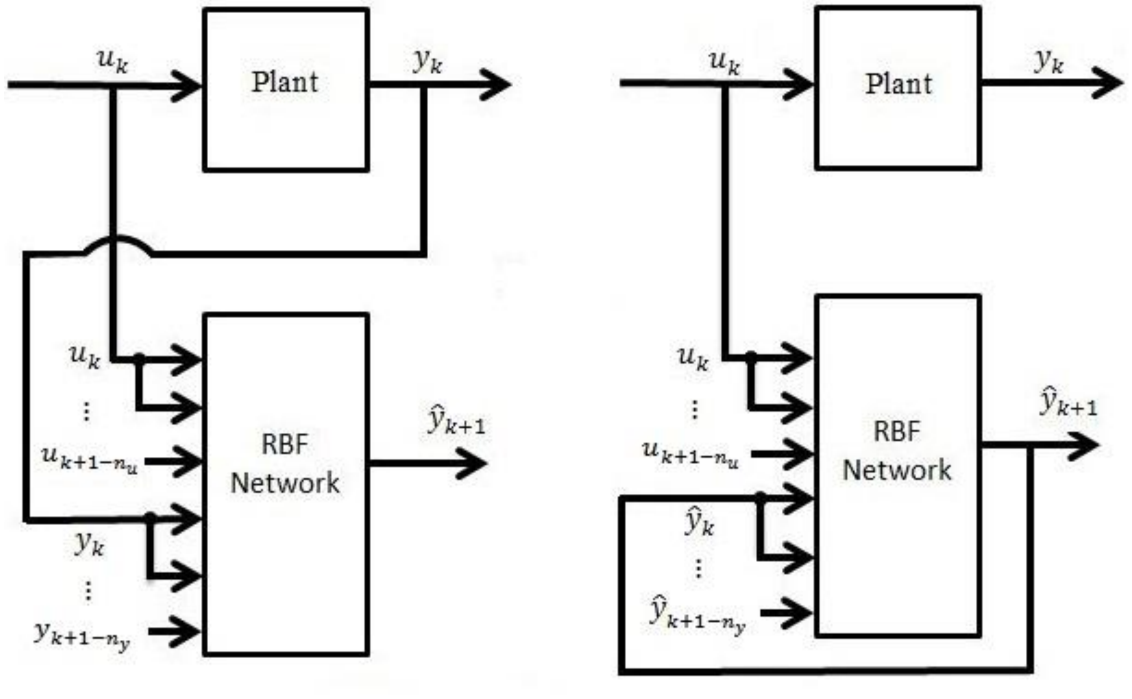

Fig. 6-1 Block diagrams of OSA and MSA predictors.

From Fig. 6-1, it is observed that the OSA predictor is trained using the plant inputs $u_k, \ldots, u_{k+1-n_u}$ and outputs $y_k, \ldots, y_{k+1-n_y}$ at sample time k to estimate the one-step-ahead prediction $\hat{y}_{k+1}$. For the OSA predictor, the future predictions $\hat{y}_{k+1}, \ldots., \hat{y}_{k+f}$ over a prediction horizon H_p are described as

$$\hat{y}_{k+1} = f\left[y_k, \ldots, y_{k+1-n_y}, u_k, \ldots, u_{k+1-n_u}\right]$$

$$\hat{y}_{k+2} = f\left[y_{k+1}, \ldots, y_{k+2-n_y}, u_{k+1}, \ldots, u_{k+2-n_u}\right]$$

$$\vdots$$

$$\hat{y}_{k+f} = f\left[y_{k+H_p-1}, \ldots, y_{k+H_p-n_y}, u_{k+H_p-1}, \ldots, u_{k+H_p-n_u}\right] \qquad (6\text{-}2)$$

In contrast, in the MSA predictor the predicted outputs $\hat{y}_{k+1}, \dots, \hat{y}_{k+H_p-1}$, instead of the plant outputs, are iteratively used in the future sample predictions $\hat{y}_{k+2}, \dots, \hat{y}_{k+f}$ across a prediction horizon H_p as described as

$$\hat{y}_{k+1} = f\left[y_k, \dots, y_{k+1-n_y}, u_k, \dots, u_{k+1-n_u}\right]$$

$$\hat{y}_{k+2} = f\left[\hat{y}_{k+1}, y_k, \dots, y_{k+2-n_y}, u_{k+1}, \dots, u_{k+2-n_u}\right]$$

$$\vdots$$

$$\hat{y}_{k+f} = f\left[\hat{y}_{k+H_p-1}, \dots, \hat{y}_{k+H_p-n_y}, u_{k+H_p-1}, \dots, u_{k+H_p-n_u}\right]. \tag{6-3}$$

From (6-2), it is understood that the MSA predictor is less accurate compared with the OSA because the MSA uses the predicted outputs iteratively as its inputs, which introduces accumulated modelling error to the future predictions. However, though the MSA predictor is not as accurate as the OSA predictor, its ability to predict the multistep-ahead behaviour is essential for being used in the MPC (Yu et al., 2004, Wang et al., 2006).

In order to improve the inaccurate prediction and at the same time to reserve the multistep-ahead prediction ability of the MSA predictor, a new FS prediction model is proposed here. The f-step model is designed to make predictions over a prediction horizon H_p without using the future process outputs. It therefore makes use of the advantages of both the OSA and MSA predictors. The derivation of f-step model commences with a NARX model in (6-1). To illustrate the concept of proposed f-step model, consider an example with output orders $n_u = 2$ and input orders $n_y = 2$ to make predictions across a prediction horizon $H_p = 3$. Using this example, the NARX model in (6-1) can be expressed as

$$\hat{y}_k = F[y_{k-1}, y_{k-2}, u_{k-1}, u_{k-2}] \tag{6-4}$$

The outputs $y_{k-1}, \dots, y_{k-2}$ can be described in prediction forms of

$$\hat{y}_{k-1} = F[y_{k-2}, y_{k-3}, u_{k-2}, u_{k-3}]$$

$$\hat{y}_{k-2} = F[y_{k-3}, y_{k-4}, u_{k-3}, u_{k-4}] \tag{6-5}$$

Now, using (6-5) to approximate y_{k-1}, y_{k-2} in (6-4), it becomes

$$\begin{aligned}\hat{y}_k &= F[F[y_{k-2}, y_{k-3}, u_{k-2}, u_{k-3}], F[y_{k-3}, y_{k-4}, u_{k-3}, u_{k-4}], u_{k-1}, u_{k-2}] \\ &= F[F[F[y_{k-3}, y_{k-4}, u_{k-3}, u_{k-4}], y_{k-3}, u_{k-2}, u_{k-3}], \\ &\quad F[y_{k-3}, y_{k-4}, u_{k-3}, u_{k-4}], u_{k-1}, u_{k-2}]\end{aligned} \tag{6-6}$$

Using a function G to represent the composite function F in (6-6),

$$\hat{y}_{k|k-3} = G[y_{k-3}, y_{k-4}, u_{k-1}, u_{k-2}, u_{k-3}, u_{k-4}] \tag{6-7}$$

Using (6-7) for predictions over $H_p = 3$, they are as follows,

$$\begin{aligned}\hat{y}_{k+1|k-2} &= G[y_{k-2}, y_{k-3}, u_k, u_{k-1}, u_{k-2}, u_{k-3}] \\ \hat{y}_{k+2|k-1} &= G[y_{k-1}, y_{k-2}, u_{k+1}, u_k, u_{k-1}, u_{k-2}] \\ \hat{y}_{k+3|k} &= G[y_k, y_{k-1}, u_{k+2}, u_{k+1}, u_k, u_{k-1}]\end{aligned} \tag{6-8}$$

Equation (6-7) can be extended to a general form,

$$\hat{y}_{k|k-f} = G\left[y_{k-f}, \dots, y_{k-f+1-n_y}, u_{k-1}, .., u_{k-f+1-n_u}\right] \tag{6-9}$$

or alternatively,

$$\hat{y}_{k+f|k} = G\left[y_k, \dots, y_{k+1-n_y}, u_{k-1+f}, .., u_{k+1-n_u}\right]. \tag{6-10}$$

From (6-10), it can be observed that the f-step prediction $\hat{y}_{k+f|k}$ requires process outputs up to k^{th} sample time, which are all available at current sample period k. In other words, only process output measurements up to k^{th} sample time are required for the prediction of outputs up to $(k+f)^{\text{th}}$ sample time. This means that the dependency on the future predicted outputs $y_{k+1}, \dots, y_{k+f-1}$ over a prediction horizon H_p is eliminated. Therefore, it improves the prediction accuracy.

6.3 Modelling a CSTR

To demonstrate the effectiveness of the developed f-step RBF model for multistep ahead prediction, the f-step RBF model is developed for a CSTR and compared with other two frequently used RBF models. The structure of the developed f-step RBF model is the same as that of normal RBF network with only difference being different input vectors. Therefore, except for forming input vector by different input/output sample values, the training of the network including find centre vector and width for the Gaussian functions in each hidden layer node and training of the connected weights between the hidden layer and the output layer, are all the same. As in (Lightbody and Irwin, 1997) and (Yu et al., 2006), the K-means clustering algorithm and p-nearest centres algorithm are used to determine the location and width of centres, respectively as described in in Section 3.2.1 and 3.2.2.. The RLS algorithm in Section 3.2.3 is used to optimize the connection weights.

6.3.1 CSTR Dynamics and Data Acquisition

A CSTR plant in (Lightbody and Irwin, 1997, Morningred et al., 1990) is selected as the example process for evaluation of modelling and control of the developed f-step RBF model as illustrated in Fig. 6-2. This process has non-linear dynamics not only in static gain and also in dynamic parameters. It is therefore often employed as a benchmark for non-linear control evaluation. The plant is described by the following non-linear differential equations,

$$\dot{C_a}(t) = \frac{q}{v}\left(C_{a0} - C_a(t)\right) - k_0 C_a(t) e^{-\frac{E}{RT(t)}} \tag{6-11a}$$

$$\begin{aligned}\dot{T}(t) = \frac{q}{v}\left(T_0 - T(t)\right) + k_1 C_a(t) e^{-\frac{E}{RT(t)}} \\ + k_2 q_c(t)\left(1 - e^{-\frac{k_3}{q_c(t)}}\right)\left(T_{c0} - T(t)\right)\end{aligned} \tag{6-11b}$$

The reactor is used to mix two chemicals to produce a product compound A. A type of exothermic reaction takes place in the reactor, which slows down the reaction resulting with non-linear dynamics. The objective of the control system is to control the concentration of product compound A, $C_a(t)$ with temperature $T(t)$ by manipulating the flow rate of coolant, $q_c(t)$. Therefore, the input is the flow rate of coolant q_c and the output is the concentration of compound A, $C_a(t)$. For the reactor the nominal values of the physical parameters are listed in Appendix A. The non-linearity of the plant is illustrated in Fig. 6-3, where the step response at different operating points (90, 106, 98, 105 l/min) is displayed. It can be observed that the dynamics are more underdamped when the concentration is higher.

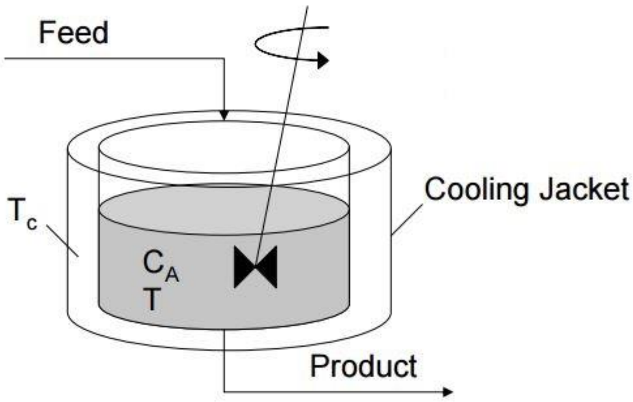

Fig. 6-2 The schemetic of the CSTR tank.

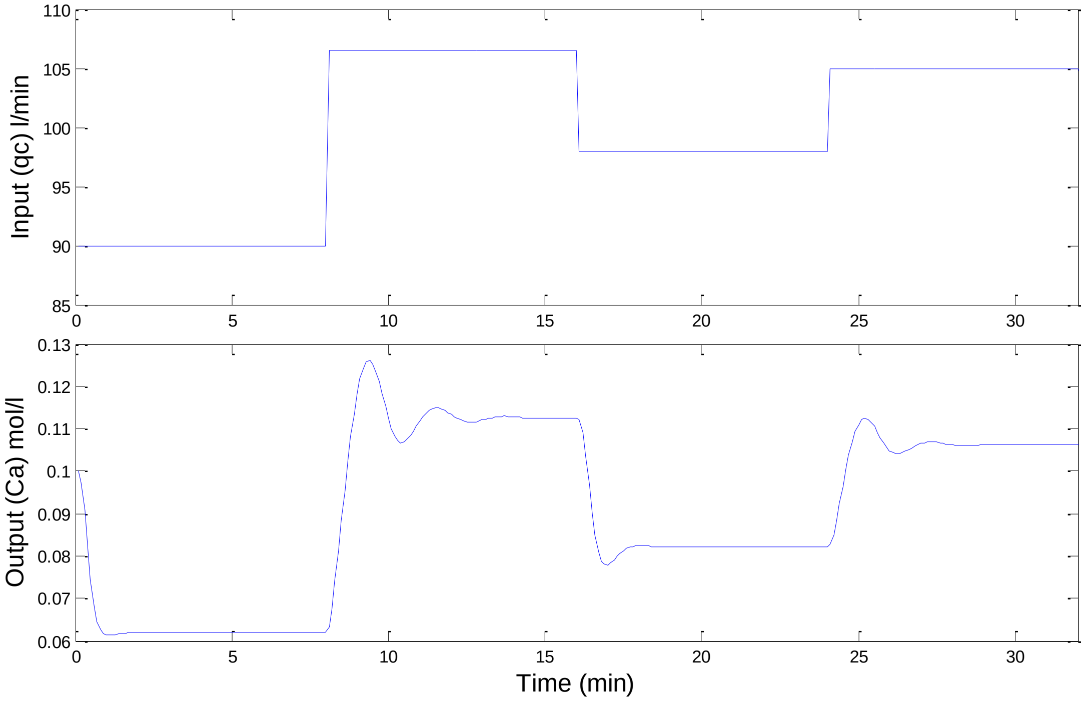

Fig. 6-3 Step responses of the CSTR plant.

The sampling period is chosen to be 0.1min. A set of persistently exciting input signal is designed as shown in Fig. 6-4 to generate a set of 1400 output data samples. The excitation input signal consists of a RAS of large amplitude superimposed on it with a RAS of small amplitude, to capture the dynamic behaviours of the plant at all frequencies and at all different levels of operating space. The collected input-output data points are halved into two sets - first 700 data samples are used as training data and the remainders are used as validation data. The orders of all variables in the network input vector are selected according to the orders of them in the reactor dynamic equations and are carefully tuned to give the best generalization result. In the meantime, the numbers of centres are decided considering a trade-off between the network size and the prediction error.

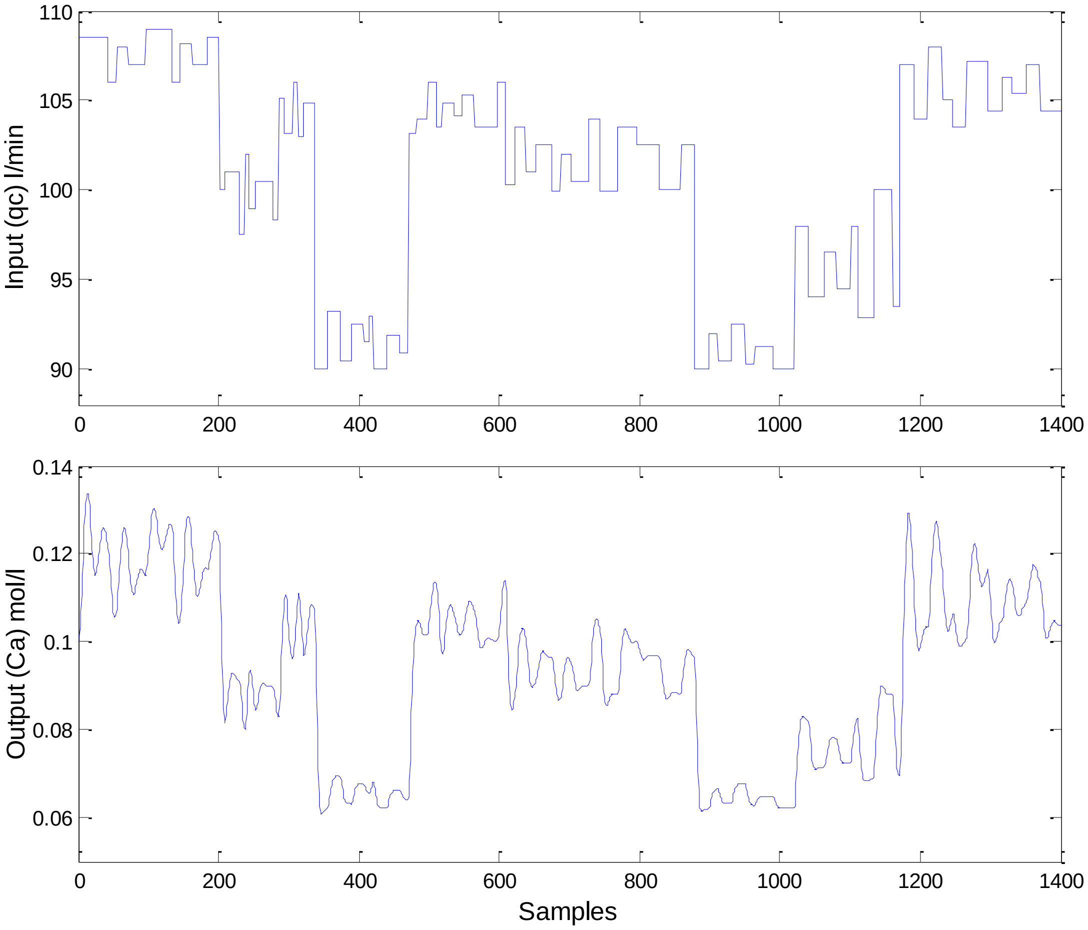

Fig. 6-4 Collected data samples.

6.3.2 f-step RBF Modelling and Comparison

In this section, the prediction performance and model compactness of the f-step RBF model are evaluated with the training and validation date sets, and compared with the NARX-RBF model and the PSC-RBF model. The NARX-RBF model is of the structures shown on right hand side in Fig. 6-1 and is able to predict for multistep ahead with the predicted outputs as model inputs. The PSC-RBF model can predict for multi-step ahead outputs (Bhartiya and Whiteley, 2001). To effectively evaluate the ability of accurate long range prediction, the proposed f-step RBF model and PSC-RBF model are tested with three different prediction horizons $H_p = 5, 10, 20$, respectively. Both training and validation data sets are scaled to [0 1] to minimize the error caused by the difference between ranges of different variables. The scaled output predictions are then scaled back after the model is used as described in Section 3.3.1.

All the three network models are trained with the RLS training algorithm in Section 3.2.3. For a fair comparison, the parameters in all networks are carefully tuned. The K-means clustering algorithm is used to find the position of centres and the radius of the Gaussian functions is calculated using P-nearest neighbour algorithm. The MAE described in Section 3.3.2 is used to measure the prediction errors. After training, the validation data set is applied to the three types of model and model prediction results are recorded in Table 6-1.

From the prediction results in Table 6-1 it is evidence that the following four points. First, both f-step RBF and PSC-RBF models have more accurate predictions than the NARX-RBF model for 5-step-ahead prediction. This is because the former two models used measured input/output data rather than the latter model used predicted output as model input to bring in accumulated error. Secondly, for long-range, $H_p = 10$ and $H_p = 20$, the f-step RBF predictions are more accurate than that of the PSC-RBF model, especially for the validation data. Thirdly, in addition to the more accurate prediction, the f-step RBF model has a much smaller size than the corresponding PSC-RBF model for all the three different prediction horizons. This point is important as the bigger size of the model will lead to a much increased computing load in MPC optimization. Fourthly, the f-step RBF model prediction has only a slight degradation with the increase of the prediction horizon compared with the PSC-RBF model. This point is vital when the model is used as the internal model in the non-linear MPC.

As can be seen, the PSC-RBF model has the largest network structures as listed in Table 6-1. With the prediction horizon increasing, the PSC-RBF model needs to use unrealistically large network structures to achieve satisfactory predictions. This prevents the PSC-RBF model to be used in MPC for systems with slow dynamics only, for example a temperature control system. Unlike the PSC model, the proposed f-step model is able to maintain its structure size while producing satisfactory predictions for long range.

Models	**NARX-RBF**	**PSC-RBF**			**f-step RBF**		
H_p	5	5	10	20	5	10	20
n_u	3	3	2	2	3	2	2
n_y	3	3	2	2	3	2	2
n_h	43	150	216	362	44	36	44
Training data MAE	0.0154	0.0086	0.0157	0.0156	0.0080	0.0104	0.0116
Validation data MAE	0.0121	0.0070	0.0125	0.0176	0.0056	0.0075	0.0081

Table 6-1 Performance comparison of different RBF-based models.

6.4 Factorization of f-step RBF Network for MPC

The purpose to factorize f-step RBF network is to derive an explicit objective function for MPC to reduce the computation burden in solving the on-line optimization problem. For a NARX model, it is impossible to factorize because its future predictions are dependent on unknown future process measurement. Although PSC model manages to offer the factorization form, an unrealistic large network structure is required to obtain a satisfied modelling performance as shown in Section 6.3.2. As the f-step RBF model is different from the PSC model, the derivation of factorization is completely different. Due to the factorability of the exponential function, the prediction output of a RBF network can be rewritten as

$$\hat{y}_{k+i} = \sum_{j=1}^{n_h} w_j \exp(Past + Future)$$

$$= \begin{bmatrix} w_1 \exp(Past) \\ \vdots \\ w_{n_h} \exp(Past) \end{bmatrix}^T \begin{bmatrix} \exp(Future) \\ \vdots \\ \exp(Future) \end{bmatrix}$$

$$= \hat{y}_p^T \hat{y}_f \qquad (6\text{-}12)$$

with $\hat{y}_p^T$ and $\hat{y}_f$ denote the past and future matrices, respectively.

Based on f-step model in (6-10), $u_{k-1+f}, \ldots, u_{k+1-n_u-1}$ and u_{k+1-n_u} can be called as past inputs and future inputs, respectively. In MPC, the future inputs $u_{k+1-n_u}$are the variables to be optimized. From Chapter 5, the objective function of MPC is described as

$$V_k = \sum_{i=H_w}^{H_p} \left\| \hat{y}_{k+i|k} - r_{k+i|k} \right\|_{Q_i}^2 + \sum_{i=0}^{H_u-1} \left\| \Delta\hat{u}_{k+i|k} \right\|_{R_i}^2 \qquad (6\text{-}13)$$

where R_i and Q_i are the penalty on the changes of inputs and the errors between output and desired set-point $r_{k+i|k}$, respectively. $\Delta\hat{u}_{k+i|k}$ represents the changes in input. H_w and H_u represent the window parameter and control horizon, respectively. From (6-13), it can be noticed that the objective function penalizes the changes in input $\Delta\hat{u}_{k+i|k}$.

Therefore, it is necessary to re-formulate f-step model in (6-9). The changes in input are described by

$$\Delta\hat{u}_k = \hat{u}_k - u_{k-1}$$

$$\hat{u}_k = \Delta\hat{u}_k + u_{k-1}$$

$$\hat{u}_{k+1} = \Delta\hat{u}_{k+1} + \hat{u}_k$$

$$= \Delta\hat{u}_{k+1} + (\Delta\hat{u}_k + u_{k-1})$$

$$\vdots$$

$$\hat{u}_{k+f-1} = \Delta\hat{u}_{k+f-1} + \Delta\hat{u}_{k+f-2} + \cdots + \Delta\hat{u}_{k+1} + \Delta\hat{u}_k + u_{k-1} \quad (6\text{-}14)$$

with $\hat{u}$ denotes the future prediction. Then, substitute (6-14) into (6-10), the f-step model becomes

$$\hat{y}_{k+1} = G\left(y_{k+1-f}, \dots, y_{k+2-f-N_y}, u_{k-2-f-N_u}, \dots, u_{k-1}, (u_{k-1} + \Delta u_k)\right)$$

$$\vdots$$

$$\hat{y}_{k+f} = G\Big(y_k, \dots, y_{k+1-N_y}, u_{k-1}, (u_{k-1} + \Delta u_k), \dots, \left(\Delta\hat{u}_{k+f-1} + \Delta\hat{u}_{k+f-2} + \cdots + \Delta\hat{u}_{k+1} + \Delta\hat{u}_k + u_{k-1}\right)\Big) \quad (6\text{-}15)$$

Note that the order of inputs is rearranged for factorization purpose. To illustrate the factorized f-step RBF model, an example with $H_p = 3$, $n_y = 2$, $n_u = 2$, and $n_h = 3$ is considered. The f-step predictions within prediction horizon in (6-10) are

$$\hat{y}_{5|1} = G(y_2, y_1, u_1, u_2, u_3, u_4)$$

$$\hat{y}_{6|2} = G(y_3, y_2, u_2, u_3, u_4, u_5)$$

$$\hat{y}_{7|3} = G(y_4, y_3, u_3, u_4, u_5, u_6) \quad (6\text{-}16)$$

and the changes in inputs are rewritten according to (6-14)

$$\hat{u}_4 = \Delta\hat{u}_4 + u_3$$

$$\hat{u}_5 = \Delta\hat{u}_5 + \hat{u}_4 = \Delta\hat{u}_5 + (\Delta\hat{u}_4 + u_3)$$

$$\hat{u}_6 = \Delta\hat{u}_6 + \hat{u}_5 = \Delta\hat{u}_6 + \Delta\hat{u}_5 + (\Delta\hat{u}_4 + u_3) \qquad (6\text{-}17)$$

By substituting (6-17) into (6-16), the predictions become

$$\hat{y}_{5|1} = G\big(y_2, y_1, u_1, u_2, u_3, (\Delta\hat{u}_4 + u_3)\big)$$

$$\hat{y}_{6|2} = G\big(y_3, y_2, u_2, u_3, (\Delta\hat{u}_4 + u_3), (\Delta\hat{u}_5 + \Delta\hat{u}_4 + u_3)\big)$$

$$\hat{y}_{7|3} = G\big(y_4, y_3, u_3, (\Delta\hat{u}_4 + u_3), (\Delta\hat{u}_5 + \Delta\hat{u}_4 + u_3), (\Delta\hat{u}_6 + \Delta\hat{u}_5 + \Delta\hat{u}_4 + u_3)\big) \qquad (6\text{-}18)$$

Using (6-12), the output predictions can be factorized as

$$\begin{aligned}
&\hat{y}_{5|1} \\
&= \begin{bmatrix} w_1 \exp\left(\dfrac{\big(c_{1,1} - y_2\big)^2 + \big(c_{1,2} - y_1\big)^2 + \big(c_{1,3} - u_1\big)^2 + \big(c_{1,4} - u_2\big)^2 + \big(c_{1,5} - u_3\big)^2}{\sigma^2}\right) \\ w_2 \exp\left(\dfrac{\big(c_{2,1} - y_2\big)^2 + \big(c_{2,2} - y_1\big)^2 + \big(c_{2,3} - u_1\big)^2 + \big(c_{2,4} - u_2\big)^2 + \big(c_{2,5} - u_3\big)^2}{\sigma^2}\right) \\ w_3 \exp\left(\dfrac{\big(c_{3,1} - y_2\big)^2 + \big(c_{3,2} - y_1\big)^2 + \big(c_{3,3} - u_1\big)^2 + \big(c_{3,4} - u_2\big)^2 + \big(c_{3,5} - u_3\big)^2}{\sigma^2}\right) \end{bmatrix}^T \\
&\times \begin{bmatrix} \exp\left(\dfrac{\big(c_{1,6} - u_3 - \Delta\hat{u}_4\big)^2}{\sigma^2}\right) \\ \exp\left(\dfrac{\big(c_{2,6} - u_3 - \Delta\hat{u}_4\big)^2}{\sigma^2}\right) \\ \exp\left(\dfrac{\big(c_{3,6} - u_3 - \Delta\hat{u}_4\big)^2}{\sigma^2}\right) \end{bmatrix}
\end{aligned} \qquad (6\text{-}19)$$

$$\hat{y}_{6|2} = \begin{bmatrix} w_1 \exp\left(\frac{\left(c_{1,1}-y_3\right)^2+\left(c_{1,2}-y_2\right)^2+\left(c_{1,3}-u_2\right)^2+\left(c_{1,4}-u_3\right)^2}{\sigma^2}\right) \\ w_2 \exp\left(\frac{\left(c_{2,1}-y_3\right)^2+\left(c_{2,2}-y_2\right)^2+\left(c_{2,3}-u_2\right)^2+\left(c_{2,4}-u_3\right)^2}{\sigma^2}\right) \\ w_3 \exp\left(\frac{\left(c_{3,1}-y_3\right)^2+\left(c_{3,2}-y_2\right)^2+\left(c_{3,3}-u_2\right)^2+\left(c_{3,4}-u_3\right)^2}{\sigma^2}\right) \end{bmatrix}^T \times \begin{bmatrix} \exp\left(\frac{\left(c_{1,5}-u_3-\Delta\hat{u}_4\right)^2+\left(c_{1,6}-u_3-\Delta\hat{u}_4-\Delta\hat{u}_5\right)^2}{\sigma^2}\right) \\ \exp\left(\frac{\left(c_{2,5}-u_3-\Delta\hat{u}_4\right)^2+\left(c_{2,6}-u_3-\Delta\hat{u}_4-\Delta\hat{u}_5\right)^2}{\sigma^2}\right) \\ \exp\left(\frac{\left(c_{3,5}-u_3-\Delta\hat{u}_4\right)^2+\left(c_{3,6}-u_3-\Delta\hat{u}_4-\Delta\hat{u}_5\right)^2}{\sigma^2}\right) \end{bmatrix} \tag{6-20}$$

$$\hat{y}_{7|3} = \begin{bmatrix} w_1 \exp\left(\frac{\left(c_{1,1}-y_4\right)^2+\left(c_{1,2}-y_3\right)^2+\left(c_{1,3}-u_3\right)^2}{\sigma^2}\right) \\ w_2 \exp\left(\frac{\left(c_{2,1}-y_4\right)^2+\left(c_{2,2}-y_3\right)^2+\left(c_{2,3}-u_3\right)^2}{\sigma^2}\right) \\ w_3 \exp\left(\frac{\left(c_{3,1}-y_4\right)^2+\left(c_{3,2}-y_3\right)^2+\left(c_{3,3}-u_3\right)^2}{\sigma^2}\right) \end{bmatrix}^T \times \begin{bmatrix} \exp\left(\frac{\left(c_{1,4}-u_3-\Delta\hat{u}_4\right)^2+\left(c_{1,5}-u_3-\Delta\hat{u}_4-\Delta\hat{u}_5\right)^2+\left(c_{1,6}-u_3-\Delta\hat{u}_4-\Delta\hat{u}_5-\Delta\hat{u}_6\right)^2}{\sigma^2}\right) \\ \exp\left(\frac{\left(c_{2,4}-u_3-\Delta\hat{u}_4\right)^2+\left(c_{2,5}-u_3-\Delta\hat{u}_4-\Delta\hat{u}_5\right)^2+\left(c_{2,6}-u_3-\Delta\hat{u}_4-\Delta\hat{u}_5-\Delta\hat{u}_6\right)^2}{\sigma^2}\right) \\ \exp\left(\frac{\left(c_{3,4}-u_3-\Delta\hat{u}_4\right)^2+\left(c_{3,5}-u_3-\Delta\hat{u}_4-\Delta\hat{u}_5\right)^2+\left(c_{3,6}-u_3-\Delta\hat{u}_4-\Delta\hat{u}_5-\Delta\hat{u}_6\right)^2}{\sigma^2}\right) \end{bmatrix} \tag{6-21}$$

with $c_{i,j}$ representing the j^{th} element of the centre vector c_i. This example shows that the output predictions of f-step RBF model is factorized into the past and the future matrices form as in (6-12). On the right hand side in (6-19 – 6-21), the first factor represents $\hat{y}_p^T$ which consists of all known process measurements, whilst the second factor represents $\hat{y}_f$ which consists of the changes in inputs to be optimized in MPC. The general form of the factorized f-step RBF model is described by

$$\begin{bmatrix}\hat{y}_{k+1}\\ \vdots \\ \hat{y}_{k+f}\end{bmatrix}=\begin{bmatrix}\hat{y}_{past,k+1}^{T}\ \hat{y}_{future,k+1}\\ \vdots \\ \hat{y}_{past,k+f}^{T}\ \hat{y}_{future,k+f}\end{bmatrix} \tag{6-22}$$

$$\hat{y}_{past,k+1}=\begin{bmatrix} w_1 exp\left\{-\frac{1}{\sigma^2}\begin{bmatrix}\left(c_{1,1}-y_{k+1-f}\right)^2+\cdots+\left(c_{1,n_y}-y_{k+2-f-n_y}\right)^2+\\ \left(c_{1,n_y+1}-u_{k+2-f-n_u}\right)^2+\cdots+\left(c_{1,n_y+n_u+f-2}-u_{k-1}\right)^2\end{bmatrix}\right\}\\ \vdots \\ w_{n_h} exp\left\{-\frac{1}{\sigma^2}\begin{bmatrix}\left(c_{n_h,1}-y_{k+1-f}\right)^2+\cdots+\left(c_{n_h,n_y}-y_{k+2-f-n_y}\right)^2\\ +\left(c_{n_h,n_y+1}-u_{k+2-f-n_u}\right)^2+\ldots+\left(c_{n_h,n_y+n_u+f\,-2}-u_{k-1}\right)^2\end{bmatrix}\right\}\end{bmatrix}$$

$\vdots$

$$\hat{y}_{past,k+f}$$
$$=\begin{bmatrix} w_1 exp\left\{-\frac{1}{\sigma^2}\left[\left(c_{n_h,1}-y_{k+1-f}\right)^2+\cdots+\left(c_{n_h,n_y}-y_{k+2-f-n_y}\right)^2+\left(c_{n_h,n_y+1}-u_{k-1}\right)^2\right]\right\}\\ \vdots \\ w_{n_h} exp\left\{-\frac{1}{\sigma^2}\left[\left(c_{n_h,1}-y_{k+1-f}\right)^2+\cdots+\left(c_{n_h,n_y}-y_{k+2-f-n_y}\right)^2+\left(c_{n_h,n_y+1}-u_{k-1}\right)^2\right]\right\}\end{bmatrix} \tag{6-23}$$

$$\hat{y}_{future,k+1}=\begin{bmatrix} exp\left\{-\frac{1}{\sigma^2}\left[\left(c_{1,n_y+n_u+f-1}-u_{k-1}-\Delta\hat{u}_k\right)^2\right]\right\}\\ \vdots \\ exp\left\{-\frac{1}{\sigma^2}\left[\left(c_{n_h,n_y+n_u+f-1}-u_{k-1}-\Delta\hat{u}_k\right)^2\right]\right\}\end{bmatrix}$$

$\vdots$

$$\boldsymbol{\hat{y}_{future,k+H_u}}$$
$$=\begin{bmatrix} \boldsymbol{exp}\left\{-\boldsymbol{\frac{1}{\sigma^2}}\begin{bmatrix}\left(\boldsymbol{c_{1,n_y+2}}-\boldsymbol{u_{k-1}}-\Delta\boldsymbol{\hat{u}_k}\right)^{\mathbf{2}}+\cdots+\\ \left(\boldsymbol{c_{1,k+n_y}}-\Delta\boldsymbol{\hat{u}_{k+H_u-1}}-\Delta\boldsymbol{\hat{u}_{k+H_u-2}}-\cdots-\Delta\boldsymbol{\hat{u}_{k+1}}-\Delta\boldsymbol{\hat{u}_k}-\boldsymbol{u_{k-1}})\right)^{\mathbf{2}}\end{bmatrix}\right\}\\ \vdots \\ \boldsymbol{exp}\left\{-\boldsymbol{\frac{1}{\sigma^2}}\begin{bmatrix}\left(\boldsymbol{c_{n_h,n_y+2}}-\boldsymbol{u_{k-1}}-\Delta\boldsymbol{\hat{u}_k}\right)^{\mathbf{2}}+\cdots+\\ \left(\boldsymbol{c_{n_h,k+n_y}}-\Delta\boldsymbol{\hat{u}_{k+H_u-1}}-\Delta\boldsymbol{\hat{u}_{k+H_u-2}}-\cdots-\Delta\boldsymbol{\hat{u}_{k+1}}-\Delta\boldsymbol{\hat{u}_k}-\boldsymbol{u_{k-1}})\right)^{\mathbf{2}}\end{bmatrix}\right\}\end{bmatrix}$$

$\vdots$

$$\hat{y}_{future,k+f} = \begin{bmatrix} exp\left\{-\frac{1}{\sigma^2}\left[\begin{matrix}\left(c_{1,n_y+2}- u_{k-1}-\Delta\hat{u}_k\right)^2+\cdots+\\ \left(c_{1,k+n_y}-\Delta\hat{u}_{k+H_u-1}-\Delta\hat{u}_{k+H_u-2}-\cdots-\Delta\hat{u}_{k+1}-\Delta\hat{u}_k-u_{k-1})\right)^2\end{matrix}\right]\right\} \\ \vdots \\ exp\left\{-\frac{1}{\sigma^2}\left[\begin{matrix}\left(c_{n_h,n_y+2}- u_{k-1}-\Delta\hat{u}_k\right)^2+\cdots+\\ \left(c_{n_h,k+n_y}-\Delta\hat{u}_{k+H_u-1}-\Delta\hat{u}_{k+H_u-2}-\cdots-\Delta\hat{u}_{k+1}-\Delta\hat{u}_k-u_{k-1})\right)^2\end{matrix}\right]\right\} \end{bmatrix} \tag{6-24}$$

Notice all control inputs remain constant from H_u to f in MPC. With the factorization of f-step RBF model, the objective function in MPC in (6-13) becomes explicit in the changes of inputs,

$$V_k = \sum_{i=H_w}^{H_p} \left\|\hat{y}^T_{past,\ k+i}\,\hat{y}_{future,k+i}(\Delta\hat{u}_{k+i}) - r_{k+i|k}\right\|^2_{Q_i} + \sum_{i=0}^{H_u-1}\left\|\Delta\hat{u}_{k+i|k}\right\|^2_{R_i} \tag{6-25}$$

From (6-25), it is understood that unlike the NARX-RBF model, the factorized f-step RBF model provides an analytical form of objective function for MPC. $\hat{y}^T_{past,k+1}, \cdots, \hat{y}^T_{past,k+f}$ in (6-23) are only computed once at each sample time, which reduces the computational requirement when solving the on-line optimization problem.

6.5 MPC of the CSTR

To evaluate the effectiveness of the proposed f-step RBF model-based MPC (MPC-FS), the CSTR plant in Section 6.3.1 is considered. The CSTR plant possesses different non-linear characteristics in different level of the product concentration. Thus, the control performance is assessed in three levels of product concentration: high (0.11mol/l), middle (0.09mol/l), and low (0.065mol/l). Step changes of the set-point in two different levels are used to further test the dynamic response of the control approach. The upper and lower bound constraints for the control variable, coolant flow rate q_c , are set to 110 and 90, respectively. There is no constraint imposed on the outputs. The MPC approaches based on both the NARX-RBF model and the PSC-RBF model for the CSTR are developed. The control system performances and the computing loads with the two different models are then compared with the MPC based on the f-step RBF model.

6.5.1 Control Performance

To provide a fair comparison, the prediction horizon H_p and control horizon H_u for all control approaches are set to 10 and 5, respectively. It is realized that in MPC, the control performance is strongly based on the prediction accuracy of the internal model that represents the real process. Thus, based on the modelling performance, the model orders and number of centres are selected from Table 6-1 in Section 6.3.2. In practice, the parameters of three control approaches are well tuned and recorded in Table 6-2.

Control Strategy	**Parameters**
MPC-NARX	$R_i = 1.1, Q_i = 0.9$
MPC-FS	$R_i = 1.2, Q_i = 0.75$
MPC-PSC	$R_i = 1.15, Q_i = 0.8$

Table 6-2 Control parameters.

In the first control experiment a small step change in the middle level of concentration $C_a = 0.09mol/l$ as shown in Fig. 6-5 is used as the set-point. The set-point is a rectangular waveform where the product concentration level decreases from 0.1mol/l to 0.09mol/l, and then increases from 0.09mol/l to 0.1mol/l again. The control performances of all compared control strategies are shown in Fig. 6-5 and the MAE are recorded in Table 6-3. The MPC-NARX has the largest overshoot among three control strategies, for both drop and rise of the set-points. Meanwhile, the MPC-PSC has the largest steady-state error in both drop and rise scenarios. Conversely, the MPC-FS clearly has the least steady-state error. From the MAE presented in Table 6-3, the overall control performance of MPC-FS is the best among the group of three models.

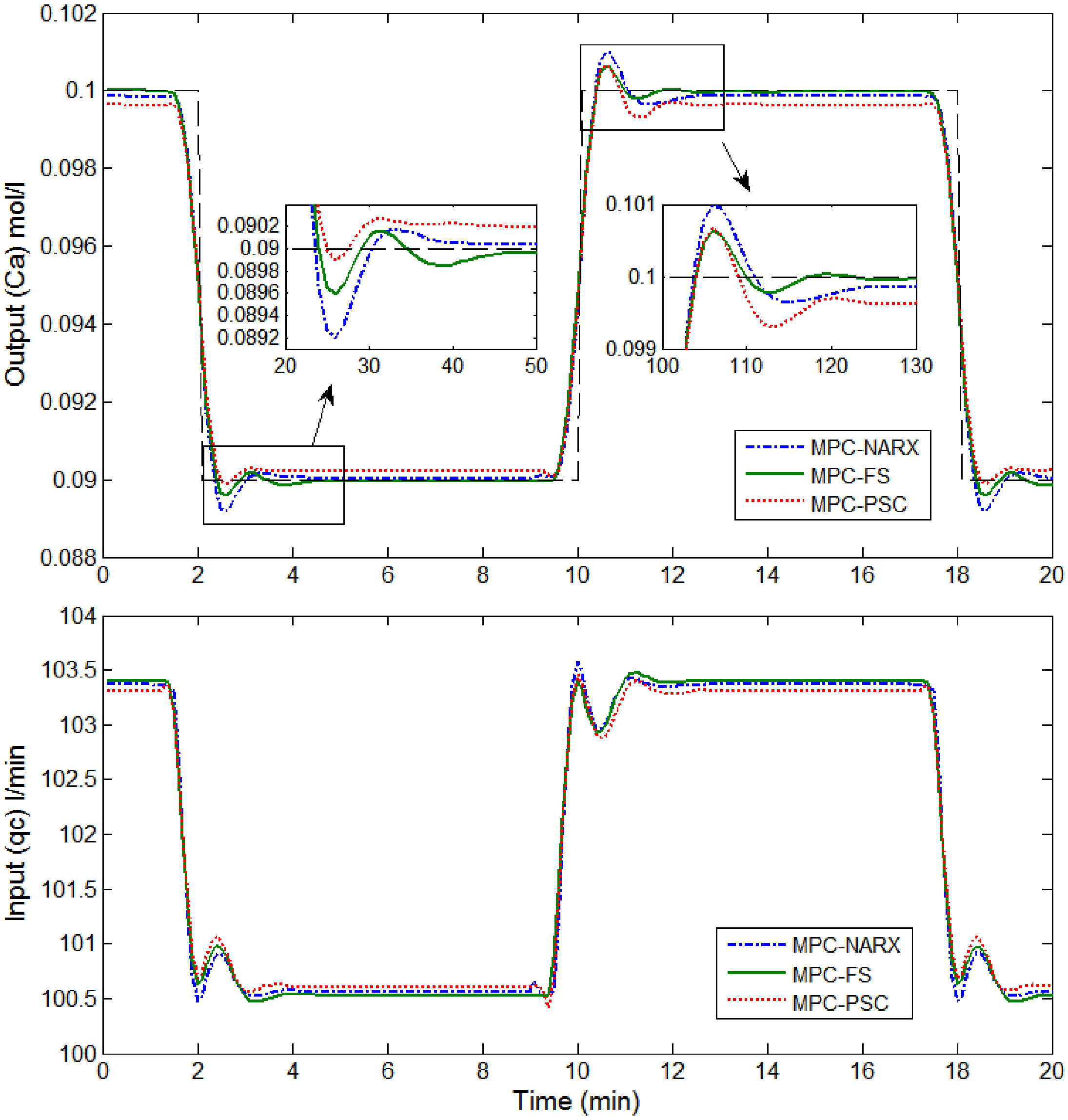

Fig. 6-5 Control performance in middle level of product concentration.

The second experiment is executed with a large step change in the set-point, which involves the low and high levels of product concentration, $C_a = 0.11$ and $C_a = 0.065$, respectively. The product concentration increases from 0.09mol/l to the highest level 0.11mol/l, and then decreases to the lowest level 0.065mol/l. The process responses with the manipulated variables are displayed in Fig. 6-6. Besides, the MAE is given in Table 6-3.

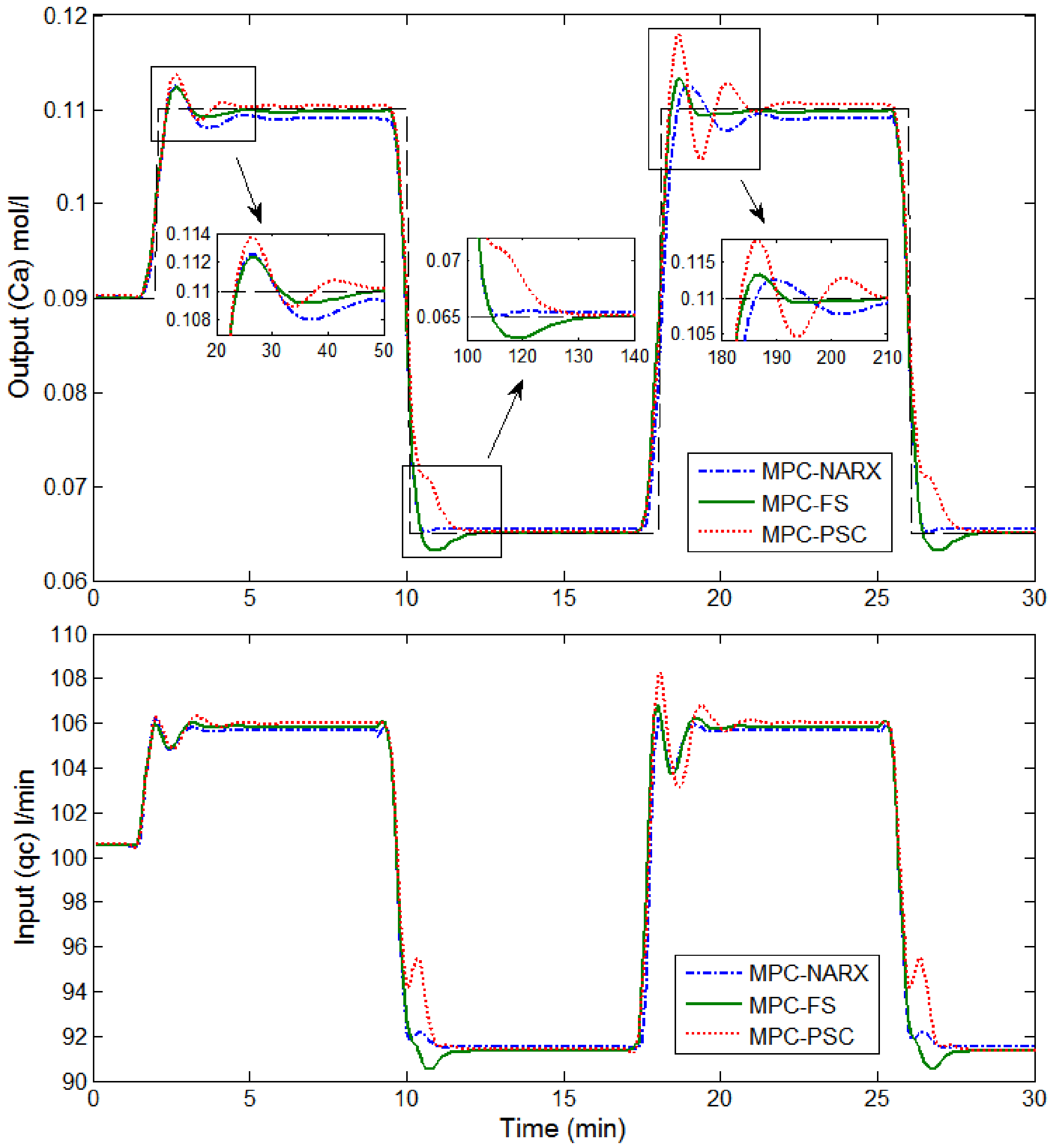

Fig. 6-6 Control performance in low and high level of product concentration.

In the both rise of step changes, the MPC-PSC has the largest overshoot. The MPC-NARX and MPC-FS have similar overshoot when the set-point jumps, but the MPC-NARX has severe oscillation. When the set-point drops, the MPC-PSC has the slowest response. Meanwhile, in the same scenario, there is a small acceptable overshoot for MPC-FS. In both rise and drop scenarios, the proposed MPC-FS has the smallest steady-state error, as illustrated in Fig. 6-6. Furthermore, from Table 6-3, the MAE clearly suggests that the proposed MPC-FS has the best overall control performance.

As a whole, in the evaluation of control performances, it is verified that the proposed MPC-FS has a better performance than that of MPC-NARX and MPC-PSC. In the control in the middle level of product concentration, it can be said that all the three control approaches have similar performance. For the control in low and high level of product concentration, the advantage of proposed MPC-FS over the other two approaches is more obvious. This is because the non-linearity of CSTR plant is much stronger on the high level of product concentration than the other regions of the operating space. In addition, as presented in Section 6.3.2, the MPC-FS has the best modelling performance among the three RBF models, which is the main reason for the better control performance.

Control Strategy	**MAE**	
	Middle level	**Low & High level**
MPC-NARX	0.00037880	0.0016
MPC-FS	0.00032564	0.0013
MPC-PSC	0.00051959	0.0017

Table 6-3 Control performances.

6.5.2 Computing Load

As mentioned previously, although a non-linear model appears to be a better internal model in MPC, one of disadvantages of NMPC is its large computational load in on-line optimization of control variables. Therefore, after the evaluation of modelling and control performance, it is imperative to evaluate the computational requirement of all the control strategies in previous section. In this work, MATLAB R2009a on an Intel Core i3 laptop with Windows 7 system is used to carry out this simulation. For a fair comparison, the optimization problem in all control strategies is solved using the Matlab function, *fmincon SQP*, as mentioned in Section 5.11. The computation time in solving the on-line optimization problem is measured using *tic-toc* command in Matlab.

The on-line computing times of all control algorithms at each sample time for the control in middle level and low/high level of product concentration are shown in Fig. 6-7 and 6-8, respectively. The total computation times are recorded in Table 6-4. It is observed that the computing times of all the three approaches are high when step changes occur. Then, the computing times gradually reduce as the outputs track the constant set-point as illustrated in both Fig. 6-7 and 6-8. This is because the control value used in this sample time is set as the initial value of the control for next sample time, so that the computing load is greatly reduced. For middle level control, the highest computation time of MPC-NARX, MPC-FS and MPC-PSC are approximately 6s, 3.26s and 9.327s, respectively. For low/high control, the highest computation time of MPC-NARX, MPC-FS and MPC-PSC are approximately 5s, 3.33s and 33s, respectively. Considering the computing time using Matlab is more than 10 times high compared that using the industrial C code, the highest computing time 6 seconds in MPC-NARX using Matlab is equivalent to 0.6 seconds using the C code, which is acceptable for the sampling time of 6 second. While the MPC-PSC for large set-point change uses 33 seconds to solve the optimization problem, which is much larger than the MPC-FS.

Furthermore, it is obvious the MPC-PSC has the largest total computation time as presented in Table 6-4. Although the MPC-PSC is a factorized approach, its unrealistic huge network structure to acquire a satisfying modelling performance has increased the size of factorized matrices in (6-23) and (6-24). Therefore, the computation load has inevitably become greater during the execution of optimized control variable.

On the other hand, it is noticeable that the computation time of MPC-FS for low/high level is not as efficient as for middle level due to the high non-linearity characteristic in this region. Despite this, the proposed MPC-FS is proved to have advantages over MPC-NARX with lower computation times every time step changes occur as shown in Fig. 6-7 and 6-8. Furthermore, Table 6-4 suggests that in overall, the MPC-FS is more efficient than the MPC-NARX. This is because unlike the MPC-NARX, the MPC-FS has the advantages using the factorized model in (6-23) where $\hat{y}_{past,k+1}^{T}, \dots ,\hat{y}_{past,k+f}^{T}$ are only needed to be computed once at each sample time instant, which reduces the computational requirement. In this simulation, it is proved that the MPC-FS is a more efficient control strategy in term of control results and computational requirement.

Control Strategy	**Total Computation Times (s)**	
	Middle level	**Low & High level**
MPC-NARX	155.4303	221.0456
MPC-FS	125.3157	213.2456
MPC-PSC	394.3540	1565.8

Table 6-4 Total computation times.

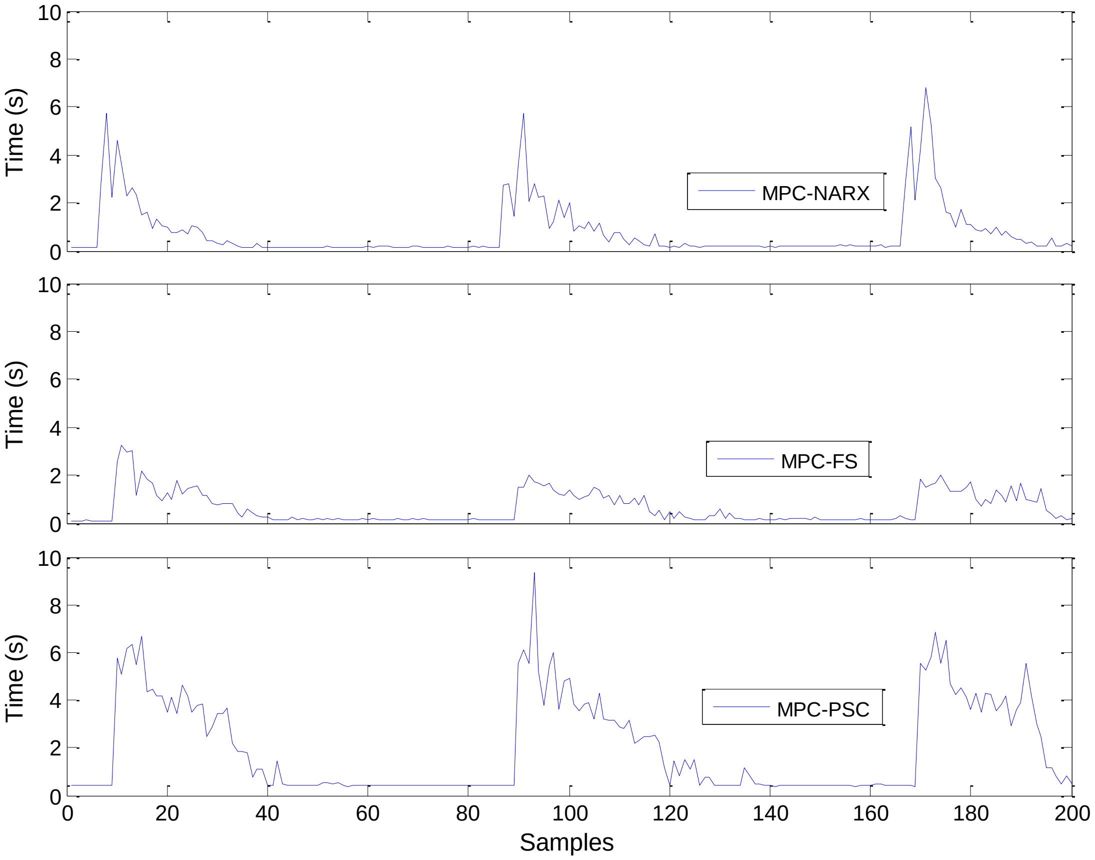

Fig. 6-7 Computing times of all control approaches at the control in middle level.

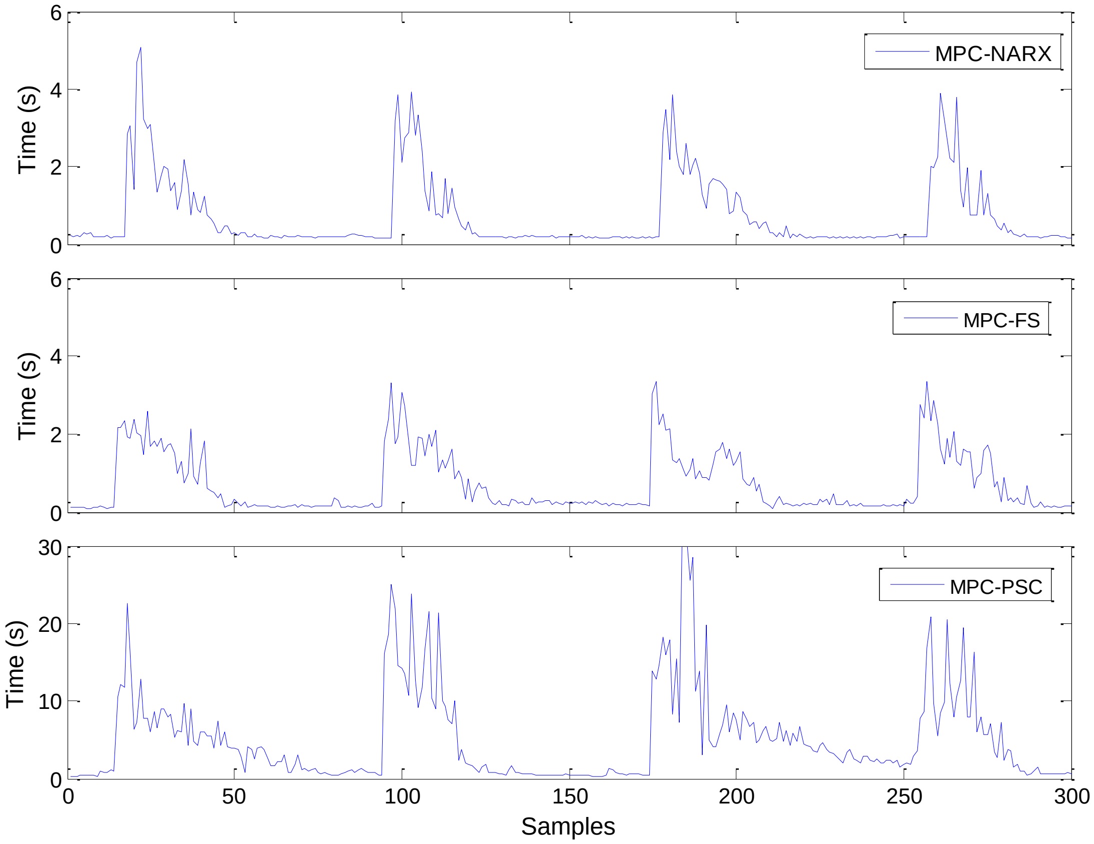

Fig. 6-8 Computing times of all control approaches at the control in low and high level.

6.6 Summary

In summary, a new f-step RBF model is developed for non-linear dynamics processes and its factorized form is developed for the application in MPC. A f-step prediction model is firstly developed and implemented with a RBF network. The effectiveness of the f-step RBF model is verified in modelling a CSTR plant. The comparison results demonstrate that the f-step RBF model outperforms the PSC-RBF model in term of prediction accuracy and model compactness. In addition, the proposed network model matches the NARX-RBF model in model compactness and it presents a better modelling performance. With these two advantages, the developed network model is more effective to be used in MPC for output prediction. After that, the factorization of proposed network is derived and an explicit MPC's objective function is obtained to reduce the computational load. The control performance and the computational load of the MPC based on the proposed model are evaluated by controlling the CSTR plant.

Comparing with existing control approaches, the results show that the proposed control approaches possesses a more efficient and better control performance. These advantages proved that the developed factorized f-step RBF model provided a better approach to be applied in MPC.

Chapter 7

f-STEP RBF MODEL-BASED MPC FOR PEMFC STACK SYSTEM

7.1 Introduction

In recent years, green technology has been widely promoted due to the climate changes and depletion of fossil fuel. Fuel cell, which is widely regarded as a potential alternative stationary and mobile power source, is an electrochemical device that converts chemical energy into electricity with hydrogen and oxygen being its fuel supplies. The proton exchange membrane fuel cell (PEMFC) stack system, as one of the popular fuel cells in automobile applications, is controlled via the f-step RBF network in this chapter.

PEMFC is a complex system, in which the relationship between the output stack voltage, net power and input variables is highly non-linear (Grötsch, 2010). Although the structure of a PEMFC stack system is simple, there are many internal factors, such as the effects of temperature (Rohendi et al., 2015) and the supply of hydrogen and air (Jo et al., 2010) and the amount of produced water (Hou et al., 2012), which need good control schemes in order to maintain the safe production and maximise the performance and lifetime of the system. In the past years, various control strategies have been implemented to provide control solutions in respect to the internal control problems of the PEMFC stack systems, such as cathode outlet flow (Feroldi et al., 2007) and water management (Damour et al., 2015). It was reported that the traditional control techniques were found not suitable in one way or another to be used in the PEMFC control system, and therefore some advanced control methods were investigated in (Rezazadeh et al., 2010), (Sedighizadeh et al., 2011), (Panos et al., 2011) and (Benchouia et al., 2015) to control the output voltage of the PEMFC stack systems, and the results showed that the performances were improved.

One of the major difficulties in controlling the PEMFC stack system is due to the sharp change in power demands as in the automobile application. This leads to the PEMFC stack system often operates under non-optimum conditions. In further, the sharp changes also cause oxygen excess ratio dramatically change. When the oxygen excess ratio is lower than a certain threshold, it will reduce the life time of the device and may damage the fuel cells, which is called oxygen starvation (Guo et al., 2013). The investigation in (Pukrushpan et al., 2002) and (Guo et al., 2013) reported that the control of air flow has an important role in maintaining the reliability and efficiency of the PEMFC stack systems against the problem of oxygen starvation. Several control strategies have been proposed for the oxygen starvation problem. Pukrushpan et al. (2002) demonstrated that a feed-forward plus feedback control can improve the oxygen response. However, the control method does not take constraints and parameter uncertainty into consideration and the improvement in oxygen response is weak. To overcome this problem, Vahidi et al. (2004) designed a model predictive control strategy and their results showed that the oxygen starvation could be avoided. However, this was achieved by using an auxiliary power source. Bordons et al. (2006) proposed a predictive controller. Their results showed that quick control response was achieved but the overall performance was not satisfactory. Recently, Abdullah and Idres (2014) employed the MPC strategy to prevent the oxygen starvation from happening. Guo et al. (2013) attempted to control the oxygen excess ratio by using a fuzzy logic control. Acceptable results were achieved in (Abdullah and Idres, 2014) and (Guo et al., 2013) but the computing load was not evaluated and was possibly large. Neural networks were also considered in modelling the non-linear dynamics of the PEMFC stack systems (Puranik et al., 2010, Grondin-Perez et al., 2014, da Costa Lopes et al., 2015) due to their strong ability in approximating non-linear maps for complex processes (Hao et al., 2011).

This chapter designs an MPC to control the oxygen excess ratio of the PEMFC stack system by regulating the air mass flow rate through tuning the air compressor motor voltage. The developed MPC employs a f-step RBF model that described in Chapter 6. The advantages of proposed f-step RBF model based MPC (MPC-FS) have been validated by simulation based on the non-linear model of the PEMFC stack system and comparison with the NARX-RBF model based MPC (MPC-NARX).

This chapter is organised as follows. Section 7.2 describes the PEMFC stack system model and oxygen starvation phenomenon. Section 7.3 introduces the f-step RBF model and presents its prediction performance for the PEMFC stack system. Section 7.4 describes the control performance and computing loads of the MPC-FS.

7.2 Fuel Cell System Model

The considered PEMFC stack system in this work, as shown in Fig. 7-1 (Pukrushpan et al., 2004), can be mainly categorized into four main components, namely air flow system, hydrogen flow supply, cooling system and humidification system. The dynamics of reactant flow, heat and temperature, water management, power management and fuel processor are taken into consideration.

The dynamic reactant flow can be separated into two models – the hydrogen flow in anode flow model and the air flow in cathode flow model. The hydrogen, stored in a high pressured tank, is supplied to the anode side of the fuel cell using a valve. In order to provide a sufficient supply of hydrogen, a proportional controller is employed for the anode pressure to match the cathode pressure. This is achievable due to the high pressure flow rate of hydrogen. The membrane is assumed to be fully humidified by setting $\lambda_m = 14$. As a result, it enables the focus on the control of air supply.

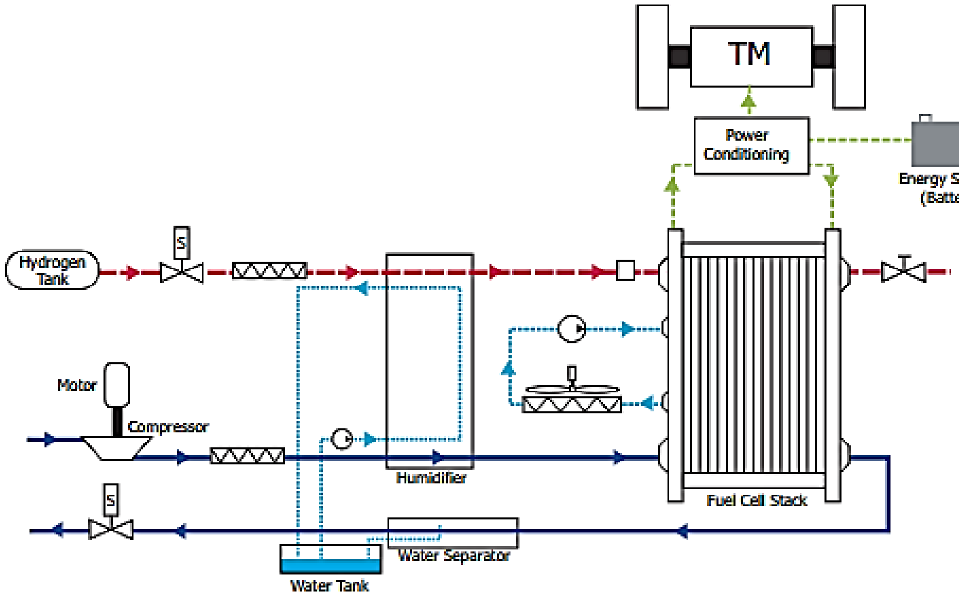

Fig. 7-1 Schematic of the PEMFC stack system (Pukrushpan et al., 2004).

In the cathode side, the air is supplied into the fuel cell stack using a compressor motor. The compressor model is separated into two parts. The first part represents a static compressor map which determines air flow rate through the compressor. The required compressor power is calculated using thermodynamic equations. The second part characterizes the compressor speed. The inlet air pressure $p_{cp,in}$, temperature $T_{cp,in}$, input voltage v_{cm} and downstream pressure are the inputs to the compressor model. The downstream pressure is determined by the supply manifold model. The dynamics of compressor speed is governed by compressor motor torque and the torque is required to drive the compressor. The compressor air flow rate is determined by manipulating the input voltage which is the input to the fuel cell system. Before entering into the fuel cell stack, the air flow is cooled and humidified to prevent damage to the fuel cell membrane. The dynamics of compressor, air cooler and humidified model can be found in (Pukrushpan et al., 2004).

7.2.1 Oxygen Starvation Phenomenon

Before examining the oxygen excess ratio (OER), it is important to observe the relationship between stack power P_{st}, net power P_{net} and compressor motor power P_{cm}. In a standalone PEMFC stack system, the power required by the compressor motor is drawn from the system. Due to the high power consumed by compressor motor compared to that of other auxiliary components, the power consumptions of other components are neglected. Thus, the net power P_{net} can be described as

$$P_{net} = P_{st} - P_{cm} \tag{7-1}$$

This relationship plays an important role in obtaining the trade-off for an optimum OER value. The OER, which is used to measure the excess amount of oxygen that being supplied to the fuel cell system in the cathode side, can be expressed as

$$\lambda_{O_2} = \frac{W_{O_2,in}}{W_{O_2,react}} \tag{7-2}$$

where $W_{O_2,in}$ and $W_{O_2,reac}$ are the supplied and reacted oxygen, respectively.

It can be seen in (7-2) that $\lambda_{O_2} > 1$, the amount of reacted oxygen is always less than supplied. A too high OER, such as greater than 2, means a high oxygen supply, which requests a high air flow rate and results in a high compressor energy consumption. On the other hand, a low QER, such as between 1 and 2 or lower than 1 instantly, will result in deficiency of oxygen supply, and causes oxygen starvation. As pointed out in (Pukrushpan et al., 2004), the oxygen starvation may cause a rapid decrease in cell voltage which affects the stack current. Moreover, it may also damage the fuel cell membrane due to a hot spot on the surface of membrane. Therefore, an optimum value of λ_{o_2} was studied in (Pukrushpan et al., 2004) and it was simplified to be a constant value of 2 in order to maximise the efficiency of the P_{net}. Using this criterion, the control problem can be formulated as the compressor motor voltage v_{cm} is manipulated to achieve the optimum value of λ_{o_2}, subject to an input disturbance which is the stack current I_{st}, as depicted in Fig. 7-2.

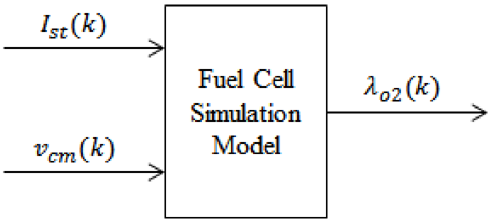

Fig. 7-2 Input/output of oxygen feeding dynamics.

7.3 f-Step RBF Model for the OER

In this section, a newly developed f-step RBF network model described in Chapter 6 is selected as the prediction model for the PEMFC stack system. Normally, when a RBF network is used to model a dynamic system, the modelling is achieved by using the network to approximate the non-linear mapping in the NARX model presented in equation (6-1), where the NARX model is used to present the non-linear dynamics of the process to be modelled.

For the f-step RBF model, the RBF network structure is not changed as presented in Section 3.1. The f-step RBF model uses a RBF network to implement an f-step predictive model equation given in (6-10). The f-step prediction model is used to present the non-linear dynamics of the system to be modelled. From the f-step prediction equation in (6-10), it can be seen that the f-step ahead outputs are predicted from the currently available process output data with the expectation of higher prediction accuracy as demonstrated in Section 6.3.2.

7.3.1 Data Collection

Due to strong non-linearity in different operating regions of the PEMFC stack system, the collected data set has been selected carefully to reflect the interested regions which will be used for control purpose in the following section. The compressor motor voltage v_{cm} is the input while the output is the OER. A set of excitation input signal, with a RAS of large amplitude superimposed with a series of small amplitude RAS at all different levels within a designated operating space in a range between 100 and 235 volts, is designed for the compressor motor voltage v_{cm}. The collected data set is subject to a disturbance which is the stack current of a range of 120 to 280 A. The sampling time is set to 0.2 sec and a set of 6600 data samples of the OER is collected for network training and validation purpose. All input/output data are scaled to [0, 1] using a linear scaling and are shown in Fig. 7-3. The scaled prediction results are then scaled back for control purpose in the next section.

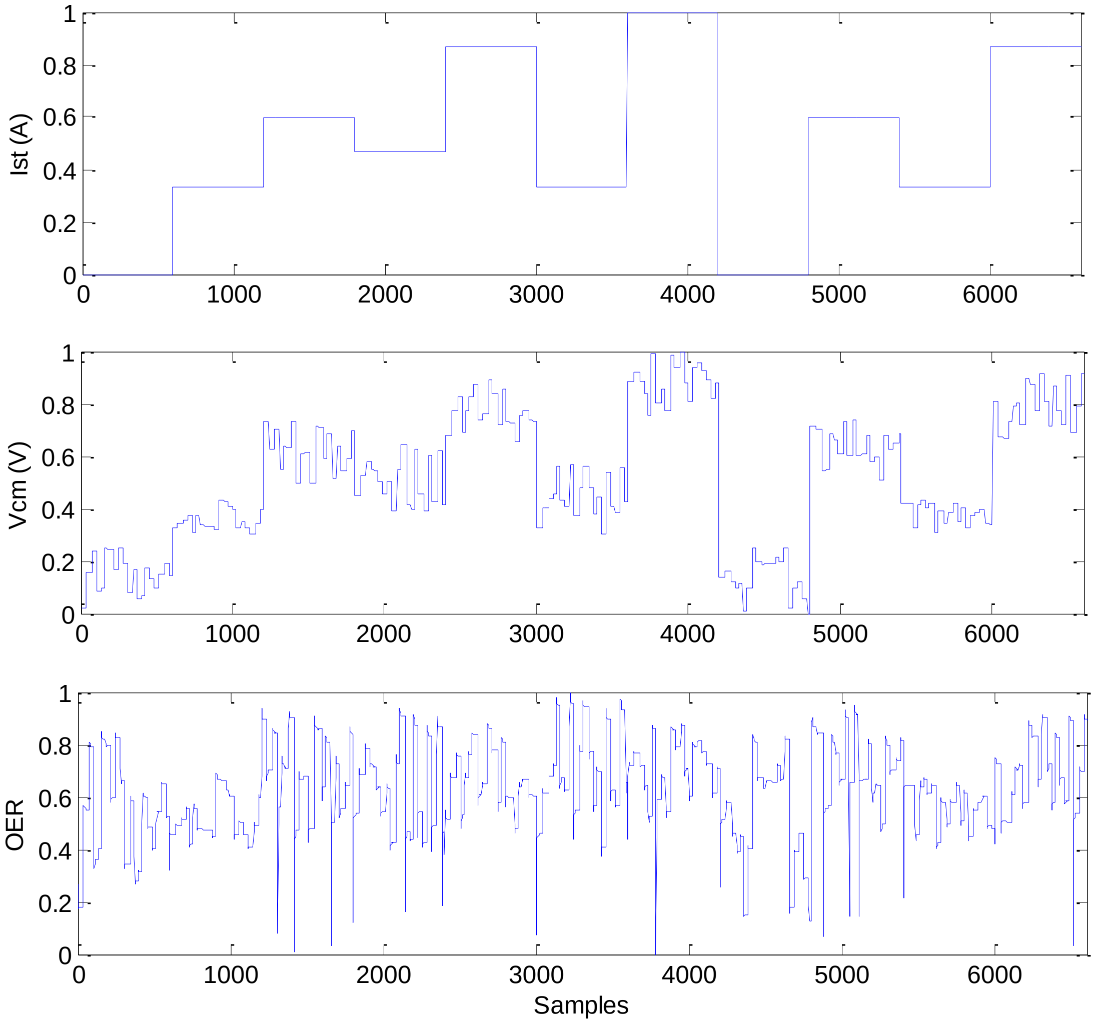

Fig. 7-3 Scaled input and output data.

7.3.2 Fuel Cell Stack System Modelling

In this sub-section, the f-step RBF model is employed to model the OER of the PEMFC stack system. The modelling performance and model compactness of the f-step RBF model are evaluated and compared with that of a NARX-RBF model. As mentioned in the previous section, three variables are chosen to be network inputs which are compressor motor voltage v_{cm}, stack current I_{st} and OER λ_{o_2}. To achieve a good performance, the orders of model for both network inputs are determined empirically and carefully tuned. Considering a trade-off between the network compactness and modelling residual, the number of the hidden neurons is decided experimentally. The collected input-output data set in previous section is divided into two sets - first 5000 data samples and remainders are used as the training data and validation data sets, respectively.

Both RBF networks are trained with the recursive least squares algorithm in Section 3.2.3. The position of centres and the radius of the Gaussian functions are computed using the K-means clustering algorithm and P-nearest neighbour algorithm, respectively, as described in Section 3.2.1 and 3.2.2. The prediction error is measured using the MAE. The criteria and modelling performances for both networks are recorded in Table 7-1.

Models	**NARX-RBF**	**f-step RBF $(H_p = 5)$**
n_u	1	1
n_{st}	2	1
n_y	3	1
n_h	42	31
Training data MAE	0.0137	0.0124
Validation data MAE	0.0149	0.0131

Table 7-1 Performance comparison of the two RBF models.

The f-step RBF model is assessed in term of modelling performance and model compactness. A good prediction performance is crucial as the control performance of the MPC is highly dependent on the accuracy of the future outputs prediction by the model. The model compactness, which contributes to the computing load of the on-line optimization in MPC, is an equally important assessment.

The results in Table 7-1 clearly indicate the advantages of f-step RBF in both modelling performance and model compactness. Firstly, the f-step RBF has a smaller modelling error. This is because, unlike the NARX-RBF, the prediction model in (6-10) allows f-step RBF network to make a series of predictions in a specified prediction horizon range at each sample period using available output variables, which successfully avoids the accumulated prediction errors. This feature also enables the factorization of f-step RBF in the application of the MPC as developed in Section 6.4. On top of that, the f-step RBF also has a more compact network structure as suggested in Table 7-1. These notable advantages of f-step RBF network model are very useful for the MPC, which will be shown in the next section.

7.4 MPC of the OER with f-step RBF Model Factorization

In this section, the MPC strategy is applied to the control of the OER in the PEMFC stack system, in which the developed f-step RBF model is used as the internal prediction model. Furthermore, the factorization algorithm in Section 6.4 is employed for the MPC, so that the time used for on-line optimization in the MPC is also significantly reduced. The factorized model is adapted with the input/output data using the RLS algorithm under the framework of the MPC, so that the robustness of the developed MPC against model uncertainty and external disturbance is also enhanced.

The strategy of the f-step RBF model-based MPC for the PEMFC stack system is displayed in Fig. 7-4. The f-step RBF network model has three input variables which are the compressor motor voltage v_{cm}, stack current I_{st} and OER λ_{o2}. The trained f-step RBF network model obtained in Section 7.3.2 is employed to make a succession of future predictions $\hat{\lambda}_{o2}(k), \dots, \hat{\lambda}_{o2}(k + H_p - 1)$ within a defined prediction horizon H_p. The control approach is as follows. Based on the information of predicted outputs $\hat{\lambda}_{o2}(k), \dots, \hat{\lambda}_{o2}(k + H_p - 1)$ and the predefined set point r, the cost function in (6-13) is minimized using a non-linear optimiser to obtain a series of optimal control variables

$v_{cm}(k), \dots v_{cm}(k+H_u)$ at each sample period. Then, the first control variable $v_{cm}(k)$ is applied to fuel cell simulation model to generate an output $\lambda_{o2}(k)$. At next sample period, the whole procedure is repeated with the newly computed $v_{cm}(k)$ and $\lambda_{o2}(k)$.

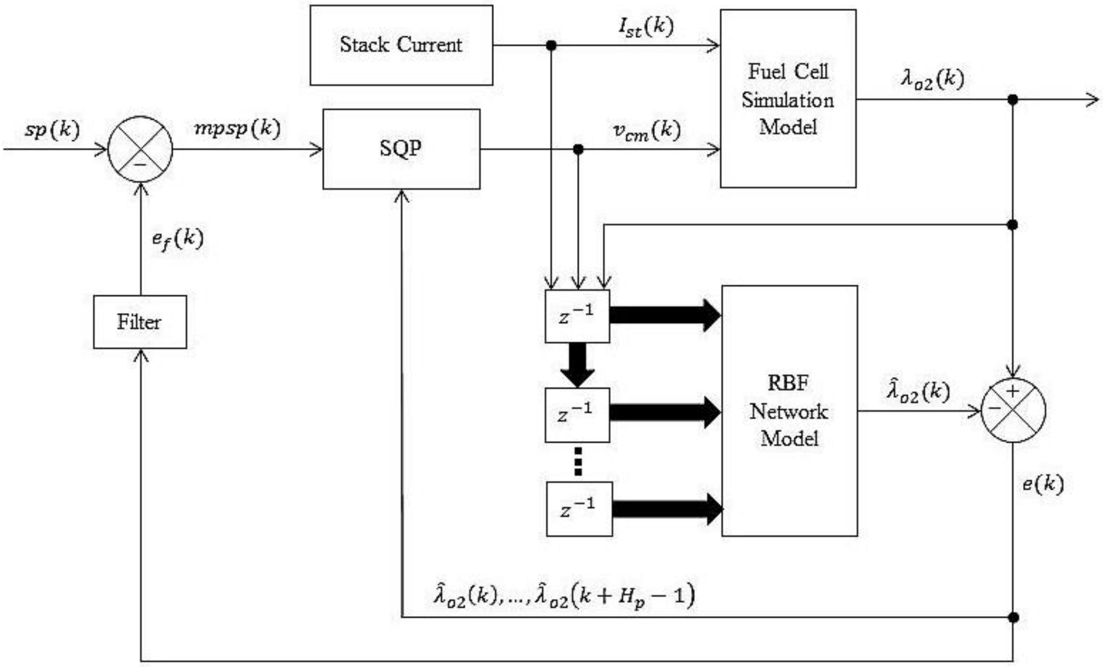

Fig. 7-4 Control scheme of PEMFC stack system.

Different with the NARX-RBF model-based MPC, the future predictions $\hat{\lambda}_{o2}(k), \dots, \hat{\lambda}_{o2}(k+H_p-1)$ in the f-step RBF model-based MPC are being predicted using known system measurements $\lambda_{o2}(k-H_p), \dots, \lambda_{o2}(k-1)$, which is clearly shown in the derivation of f-step prediction model in Section 6.2. This feature has not only enabled the factorization of network model as shown in Section 6.4 but also simplifies the objective function for the MPC.

7.4.1 Simulation

The proposed f-step RBF model-based MPC (MPC-FS) is applied to an industrial benchmark Simulink model of a PEMFC stack system developed at Michigan University (Pukrushpan et al., 2004) to evaluate its effectiveness and lower computing load. Due to the strong non-linear characteristics of the OER in the PEMFC stack system at different levels of stack current, various step changes are selected in the stack current over the operating space between 100 and 300 amps. The sampling time is chosen as 0.2s. The optimum value for OER, as mentioned in Section 7.2.1, is selected as a constant set-point $r = 2$. The upper and lower bound constraint for the control variables, compressor motor voltage v_m are set to 100 and 235 volts, respectively and there is no constraint imposed on the output variable, OER.

As mentioned in previous section, the same MPC strategy but based on the NARX-RBF model is also applied to the same Simulink model of the system, and both the control performance and the computing load are compared with the developed model with factorization. It is understood that the control performance of MPC pivots on the prediction accuracy of its internal prediction model. Hence, the model orders and number of centres for both networks are determined based on the modelling performance in Section 7.3.2. In practice, the parameters for both control approaches are chosen as $H_p = 5$, $H_u = 1$, $R_i = 0.2$ and $Q_i = 0.2$.

7.4.2 Control Performance

The two models trained in Section 7.3.2 are used in the MPC strategy to control the OER, λ_{o2} by manipulating the compressor motor voltage v_{cm} , subject to different step changes in the stack current. The control performance of the proposed MPC-FS is shown in Fig. 7-5. At every rises of step change, it can be observed in the response curve that a drop or rise in OER follows every step change in load current, it then quickly recovers by the tuning of the compressor motor voltage. For example, when the stack current drops at the 115th sample, the OER has an overshoot and then it drops before recovering to steady-states. Overall, it can be concluded that the MPC-FS achieves a satisfactory control performance with no steady state offset and fast recovery from the changes caused by the disturbances, so that the oxygen starvation can be successfully avoided.

The NARX-RBF model-based MPC exhibits a similar control performance to that shown in Fig. 7-5 and therefore, it is not displayed here. But the control performances of the two methods are compared by the MAEs, which are listed in Table 7-2. It seems that both control approaches have achieved competitive performances with the MPC-FS slightly outperforms the MPC-NARX, as suggested in Table 7-2. The reported results from both control approaches have clearly demonstrated that neural network based MPC is a potential control strategy for the PEMFC stack system. As mentioned in previous section, this achievement is imperative before assessing the computing loads in MPC which is presented in next section.

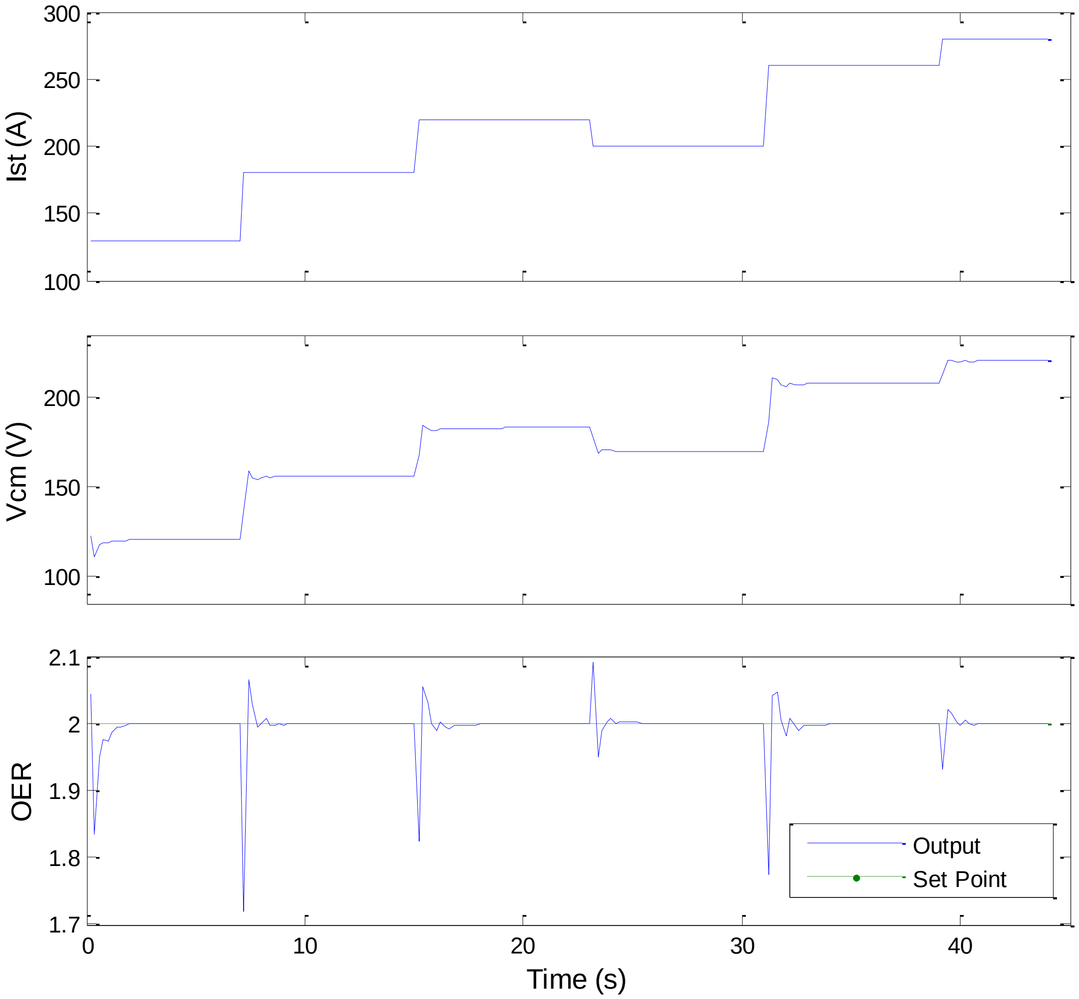

Fig. 7-5 Control performance of MPC-FS.

Control Strategy	MAEs
MPC-NARX	0.0079
MPC-FS	0.0065

Table 7-2 Control performances.

7.4.3 Computing Load

Due to the factorization in the output prediction of the f-step RBF model, the computation load in the on-line optimization in this model-based MPC is significantly reduced. This feature is validated in the MPC simulation of the PEMFC stack system. Therefore, after achieving a good control performance, the computing loads of the MPC is evaluated and compared to that of NARX-RBF model-based MPC. In this work, the simulation is performed with MATLAB R2009a on an Intel Core i3 laptop with Windows 7 operating system. The computing load is measured by recording the computation time at each sample period using the *tic-toc* command in Matlab. For a fair comparison, the optimization problem in all control strategies is solved using the Matlab function *fmincon SQP* as described in Section 5.11.

The recorded on-line computing times of both control approaches at each sample period are shown in Fig. 7-6. As can be observed, the computing times for both control approaches rise when the stack current has a step change. As the output tracks the constant set-point, the computing times reduces gradually. This is mainly because the computed control variable at the current sample instant is set as the initial value for the optimization problem at the next sample period.

From Fig. 7-6, it is obvious that the proposed f-step RBF model-based MPC requires less computing time compared with that of NARX-RBF model-based. Computing time is reduced more when the output exhibits transient responses. The highest computation times of the two model-based methods are 0.8184s and 0.4113s, respectively. Considering the computing time using Matlab is more than 10 times higher compared that using the industrial C code, the recorded computing time is acceptable for the sampling time of 0.2s. Furthermore, the total computation time recorded in Table 7-3 suggests that the MPC-FS with factorization is more efficient in terms of time than the MPC-NARX. This is because in contrast to the MPC-NARX, the factorized $\hat{y}_{past,k+1}^T, \dots, \hat{y}_{past,k+f}^T$ of MPC-FS in (6-23) are only needed to be computed once at each sample period, which reduces the computational requirement. In this experiment, the advantage of MPC-FS is presented and the method is proved to be a more efficient control strategy.

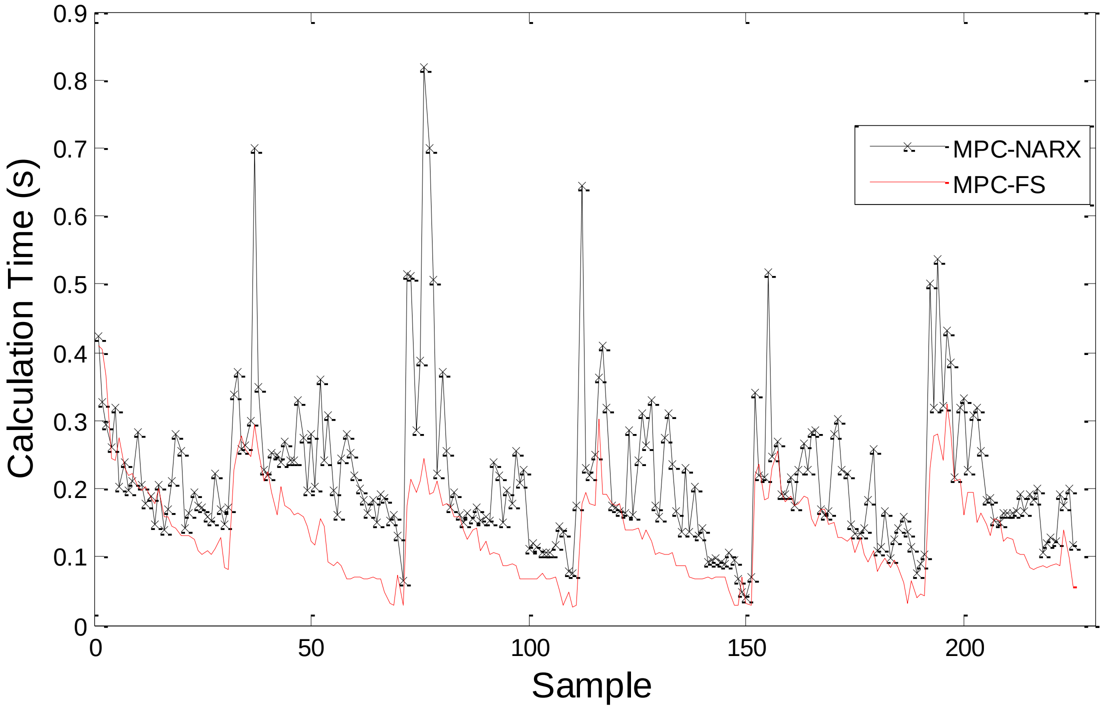

Fig. 7-6 Computing loads of the two model-based MPC.

Control Strategy	Total Computation Times (s)
MPC-NARX	49.3928
MPC-FS	31.1177

Table 7-3 Total computation times

7.5 Summary

In this Chapter, a newly proposed f-step RBF model is employed to model the PEMFC stack system and subsequently, it is used in the MPC to control the OER in the PEMFC stack system. Few key points have been concluded from this chapter:

Firstly, the developed f-step RBF model can make multi-step ahead prediction more accurately than the widely used NARX-RBF model, due to that the developed model uses available system output for prediction rather than the unknown future system output as in the NARX-RBF model.

Secondly, with the factorization algorithm, when the model used in the MPC, the multi-step ahead prediction can be computed more effectively by saving some repeated calculations, so that the time used for on-line optimization in the MPC is significantly reduced.

Thirdly, the developed model and the model-based MPC scheme are applied to control the oxygen excess ratio for a widely used industrial benchmark system: a simulated PEMFC stack system, the simulation results proves the effectiveness of the method in both long-term prediction and the execution time in the MPC scheme. The two supreme features of the developed method are also validated by comparing with a traditional NARX-RBF model and the based MPC.

Chapter 8

CONCLUSIONS AND FUTURE WORK

8.1 Conclusions

Issues on adaptive modelling techniques using RBF networks and the MPC based on RBF network models are investigated in this thesis. The contributions of the research are composed of a developed adaptation algorithm for RBF network, a proposed factorized f-step RBF model-based MPC and the application of the developed method to a PEMFC stack system.

For adaptive RBF networks, the problem of existing methods is the degradation in prediction performance after the operating point migration. This problem is solved by the developed adaptive modelling scheme for RBF networks based on ROLS training algorithm.

The proposed adaptive structure RBF network is equipped with on-line parameters optimization to achieve a fully adaptive model. The use of ROLS algorithm, which enables the training of structure and parameter simultaneously, simplifies the training procedure. In the adaptation procedure, an initial centre bank is firstly developed and three actions are proposed as follows. A new centre adding strategy has been developed to add new resourceful centres into the centre bank according to the information of new data. A centre pruning method is employed to eliminate insignificant centres to prevent an oversized centre bank. Lastly, a grouping algorithm is developed to select significant centres to form the final network model. These three integrated actions ensure that a parsimonious RBF network model is achieved. Using a non-linear simulation example with shifting operating regions, the comparison results demonstrate the effectiveness of the developed network.

The second aim is to develop a more efficient RBF model-based predictive control for complex processes, which often serves as an effectual control approach for non-linear complex processes. The NMPC requires a high computational ability as outlined in the literature review. A traditional independent NARX model restricts the factorization of a RBF network in the application of MPC, resulting in a non-explicit objective function.

A f-step prediction model is developed and implemented with a RBF network in this research. A RBF model with the f-step prediction is developed and proved to have the advantages in term of prediction accuracy and model compactness compared to that of the existing methods in modelling a CSTR plant. These advantages indicate that the developed network is a better internal model for MPC. More importantly, the f-step prediction model allows the RBF network to make future predictions without requiring the unknown process outputs. With this feature, the factorization of the model is derived to obtain an explicit MPC's objective function. The objective function consists of two parts, namely, the known past plant input/output and the unknown future plant input/output. Using the CSTR as an experimental plant, the evaluation results suggest that the developed f-step RBF model-based MPC needs much less time for execution of the on-line optimization and achieves a better control performance.

Another contribution is the application of MPC using the proposed f-step RBF model to an industrial benchmark Simulink model of a PEMFC stack system. The modelling results indicate that the proposed network has achieved a better overall performance with a smaller prediction error and a more compact structure compared to that of a NARX-RBF model. In comparison, a decrement in computing loads of the proposed control approach is observed while achieving competitive control performance. These achievements have verified the effectiveness of the developed network model.

An additional important point is that the control performances of the f-step RBF model-based MPC for both the CSTR plant (Chapter 6) and the PEMFC stack system

(Chapter 7) achieve satisfactory results. This further confirms the potential of this proposed approach. The evaluation using the CSTR plant shows that the proposed approach is able to cope with the changes in set-points and outperforms the other methods. The ability in disturbance rejection with fast response is demonstrated in controlling the PEMFC stack system. Besides, the difference between the computing times in both cases is observed. It is understood that the demands of changes in set-points require a higher computational load as longer prediction and control horizons are usually required as illustrated in controlling the CSTR plant.

On top of that, optimization techniques are investigated and a foundation of knowledge is constructed from the perspective of the author as one of the outlined motivations. The materials included in this thesis are specifically selected and arranged in a sequence that covers necessary basic knowledge and also offers the flexibility to incorporate advanced techniques related to MPC. This sequence helps ensure an effective flow of knowledge for readers. Additionally, key relationships between related techniques with the respect to LP, QP and NLP are emphasised to visualize a big picture in this discipline. This provides a clear direction that forms the background of the related optimization algorithms.

In conclusion, the following key points are drawn:

1. The proposed RBF network adaption algorithm presents a procedure that provides a smooth transition in the locations of centres according to the operating regions of a process as presented in Chapter 4. In addition, adaptive network parameter estimation is also developed, providing a fully adaptive modelling scheme for non-linear time-varying processes.

2. In the developed adaption algorithm, the centre adding strategy described in Chapter 4 is capable of producing new effective centres that cover the operating regions of a process, which is justified by its good recovery speed. This gives an alternative option in modelling processes with a wide operating region or processes with limited training data samples.

3. The f-step prediction model developed in Chapter 6 provides a great alternative to the conventional NARX-RBF model for long range prediction. The modelling of a CSTR plant and a PEMFC stack system outlines the process of verifying the developed model.

4. The developed f-step RBF network is proved to be a more potent choice for MPC due to its attractive feature of being compact and factorable, which helps reduce the computing load in the MPC. This is validated in the applications in Chapter 6 and 7 as the modelling and control outcomes in both chapters confirm that the f-step RBF model is suitable for complex processes.

8.2 Future Work

Industrial applications often consist of physical conditions and parameters that cannot be accounted by processes models. For this reason, further investigations can be carried out on the modelling and control abilities of both adaptive and factorized RBF network models. Furthermore, industrial process models from diverse areas are encouraged to extend the applications of both developed RBF network models. Recommended future work is summarized as follows.

1. It is necessary to have a suitable industry benchmark process model with migration operating regions. The effectiveness of the developed adaptive structure RBF network has been verified using a non-linear numerical example. However, the application to an industrial process model is worth doing as it would further verify the developed adaptation algorithm. Furthermore, it could be applied to MPC to examine its effectiveness.

2. The developed factorized f-step RBF model gives an explicit objective function for MPC which manages to reduce the computing loads. A further assessment on the objective function could be carried out to improve the understanding of the formulated optimization problem, which would help in the development of optimization algorithms. Besides, a continuous effort on exploring other optimization algorithms is required to expand the knowledge in this discipline.

3. Experimental works on the f-step RBF model-based MPC in the thesis are carried out using Matlab. It could be extended to a real-time simulation application using C language to fully test the proposed approach. The outcome would also be useful in evaluating the efficient of optimization algorithms for MPC. A real process, which can be used as an experimental test bed, with a good control purpose is needed to conduct this suggested experiment.

4. The developed structure adaptation algorithm can be integrated into the f-step RBF model. Based on this proposal, an implementation with MPC would be useful to control non-linear time-varying processes. In order to fully demonstrate the potential of this proposed approach, an industrial process simulation model with obvious migration of operating regions is encouraged.

5. The simulation model of the PEMFC stack system could be extended to a multivariable model. An additional input, the flow rate of hydrogen, would help to improve the OER when the demand of stack current increases. For instance, the increment in hydrogen flow rate would raise the generated power which could be used to compensate the demand of stack current. This proposed approach is possible as the hydrogen is assumed to store in a high pressured tank that allows rapid increment in its flow rate. However, the subsystem of hydrogen flow in the simulation model has a complex relationship with the oxygen supply as both of them are regulated by monitoring the manifold pressure. A further study on this relationship is required to extend the simulation model.

6. The consequence of depleted OER in the PEMFC stack system is it could burn the membrane. Aside from this, it is also imperative to control the amount of water to keep the membrane hydrated while preventing the flooding phenomena. Since the proposed control approach can be used in the PEMFC stack system, it should be tested to solve the water management problem. In order to implement this, a further investigation on this problem is needed to extend the simulation model. The proposed control scheme should not be limited to this only as the PEMFC stack system is a complex process that consists of many practical control problems within its subsystems.

REFERENCES

Abdullah, M. & Idres, M. (2014) Constrained model predictive control of proton exchange membrane fuel cell. *Journal of Mechanical Science and Technology,* 28(9), pp.3855-3862.

Allgower, F., Findeisen, R. & Nagy, Z.K. (2004) Nonlinear model predictive control: from theory to application. *J. Chin. Inst. Chem. Engrs,* 35(3), pp.299-315.

Bemporad, A., Borrelli, F. & Morari, M. (2002) Model predictive control based on linear programming— the explicit solution. *IEEE Transactions on Automatic Control,* 47(12), pp.1974-1985.

Benchouia, N.E., Derghal, A., Mahmah, B., Madi, B., Khochemane, L. & Aoul, E.H. (2015) An adaptive fuzzy logic controller (AFLC) for PEMFC fuel cell. *International Journal of Hydrogen Energy,* 40(39), pp.13806-13819.

Bhartiya, S. & Whiteley, J.R. (2001) Factorized approach to nonlinear MPC using a radial basis function model. *AIChE Journal,* 47(2), pp.358-368.

Blanco, T.B., Willems, P., De Moor, B. & Berlamont, J. (2008) Flooding prevention of the demer river using model predictive control. *IFAC Proceedings Volumes,* 41(2), pp.3629-3634.

Bland, R.G. (1977) New finite pivoting rules for the simplex method. *Mathematics of Operations Research,* 2(2), pp.103-107.

Bobrow, J.E. & Murray, W. (1993) An algorithm for rls identification parameters that vary quickly with time. *IEEE Transactions on Automatic Control,* 38(2), pp.351-354.

Boggs, P.T. & Tolle, J.W. (1995) Sequential quadratic programming. *Acta Numerica,* 4, pp.1-51.

Bordons, C., Arce, A. & Del Real, A. (2006) Constrained predictive control strategies for PEM fuel cells. *2006 American Control Conference*. pp.6.

Bortman, M. & Aladjem, M. (2009) A growing and pruning method for radial basis function networks. *IEEE Transactions on Neural Networks,* 20(6), pp.1039-1045.

Borwein, J.M. & Lewis, A.S. (2010) *Convex analysis and nonlinear optimization: theory and examples.* Springer Science & Business Media.

Boyd, S. & Vandenberghe, L. (2004) *Convex optimization.* Cambridge University Press.

Broomhead, D.S. & Lowe, D. (1988) Multivariable functional interpolation and adaptive networks. *Complex Systems,* 2, pp.321-355.

Bryan, K. & Shibberu, Y. (2005). Penalty functions and constrained optimization [Online]. lecture notes, Rose-Hulman Institute of Technology. Available: http:// www. rosehulman. edu/~ bryan/lottamath/penalty. pdf.

Byrne, C.L. (2013) Alternating minimization as sequential unconstrained minimization: a survey. *Journal of Optimization Theory and Applications,* 156(3), pp.554-566.

Chachuat, B. (2007). Nonlinear and dynamic optimization: from theory to practice [Online]. lecture notes, Automatic Control Laboratory. Available: http://lawww.epfl.ch/page4234.html.

Chang, T. & Seborg, D. (1983) A linear programming approach for multivariable feedback control with inequality constraints. *International Journal of Control,* 37(3), pp.583-597.

Chang, W.-Y. (2013) Estimation of the state of charge for a LFP battery using a hybrid method that combines a RBF neural network, an OLS algorithm and AGA. *International Journal of Electrical Power & Energy Systems*, 53, pp.603-611.

Chemori, A. & Marchand, N. (2008) Global discrete-time stabilization of the PVTOL aircraft based on fast predictive control. *IFAC Proceedings Volumes*, 41(2), pp.1747-1752.

Chen, S. (2009) Orthogonal-least-squares forward selection for parsimonious modelling from data. *Engineering*, 1(2), pp.55-74.

Chen, S., Billings, S. & Grant, P. (1992a) Recursive hybrid algorithm for non-linear system identification using radial basis function networks. *International Journal of Control*, 55(5), pp.1051-1070.

Chen, S., Billings, S. & Luo, W. (1989) Orthogonal least squares methods and their application to non-linear system identification. *International Journal of Control*, 50(5), pp.1873-1896.

Chen, S. & Billings, S.A. (1992) Neural Network for nonlinear dynamic system modeling and identification. *International Journal of Control,* 56(2), pp.319-346.

Chen, S., Cowan, C.F.N. & Grant, P.M. (1991) Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Transactions on Neural Netwoks*, 2(2), pp.302-309.

Chen, S., Grant, P.M. & Cowan, C.F.N. (1992b) Orthogonal least-square algorithm for training multioutput radial basis function networks. *Radar and Signal Processing, IEE Proceedings F,* 139(6), pp.378-384.

Chinneck, J.W. (2006). Practical optimization: a gentle introduction [Online]. lecture notes, Carleton University. Available: http://www.sce.carleton.ca/faculty/chinneck/po.html.

Chng, E., Yang, H.H. & Bos, S. (1996) Orthogonal least-squares learning algorithm with local adaptation process for the radial basis function networks. *IEEE Signal Processing Letters,* 3(8), pp.253-255.

Clarke, D.W., Mohtadi, C. & Tuffs, P. (1987) Generalized predictive control—Part I. the basic algorithm. *Automatica*, 23(2), pp.137-148.

Coetzee, L.C., Craig, I.K. & Kerrigan, E.C. (2010) Robust nonlinear model predictive control of a run-of-mine ore milling circuit. *IEEE Transactions on Control Systems Technology*, 18(1), pp.222-229.

Cottle, R.W. (2012) William Karush and the KKT theorem. *Doc. Math., Extra volume: Optimization stories*, pp.255-269.

Cui, Y., Shi, J. & Wang, Z. (2016) Lazy Quantum clustering induced radial basis function networks (LQC-RBFN) with effective centers selection and radii determination. *Neurocomputing*, 175, pp.797-807.

Cutler, C.R. & Ramaker, B.L. (1979) Dynamic matrix control--A computer control algorithm. *Joint Automatic Control Conference*. v.17 pp.72.

Da Costa Lopes, F., Watanabe, E.H. & Rolim, L.G.B. (2015) A control-oriented model of a PEM fuel cell stack based on NARX and NOE neural networks. *IEEE Transactions on Industrial Electronics*, 62(8), pp.5155-5163.

Dai, Y.-H. (2013) A perfect example for the BFGS method. *Mathematical Programming*, 138(1), pp.501-530.

Damour, C., Benne, M., Grondin-Perez, B., Chabriat, J.-P. & Pollet, B.G. (2015) A novel non-linear model-based control strategy to improve PEMFC water management – The flatness-based approach. *International Journal of Hydrogen Energy*, 40(5), pp.2371-2376.

Dantzig, G.B. & Thapa, M.N. (2006) *Linear programming 2: theory and extensions.* Springer Science & Business Media.

Dawson, C., Harpham, C., Wilby, R. & Chen, Y. (2002) Evaluation of artificial neural network techniques for flow forecasting in the River Yangtze, China. *Hydrology and Earth System Sciences Discussions*, 6(4), pp.619-626.

Dekeyser, R.M. (1988) Application of extended prediction self-adaptive control. *In:* Warwick, K. (ed.) *Implementation of Self-tuning Controllers.* London, UK: Peter Peregrinus Ltd.

Di Pillo, G. & Grippo, L. (1989) Exact penalty functions in constrained optimization. *SIAM Journal on Control and Optimization*, 27(6), pp.1333-1360.

Diaconescu, E. (2008) The use of NARX neural networks to predict chaotic time series. *WSEAS Transactions on Computer Research,* 3(3), pp.182-191.

Ding, Y., Lushi, E. & Li, Q. (2010) Investigation of quasi-newton methods for unconstrained optimization. *International Journal of Computer Application,* 29, pp.48-58.

Dreyfus, S.E. (1964) Some types of optimal control of stochastic systems. *Journal of the Society for Industrial and Applied Mathematics, Series A: Control,* 2(1), pp.120-134.

Drioli, C. & Rocchesso, D. (2003) Orthogonal least squares algorithm for the approximation of a map and its derivatives with a RBF network. *Signal Processing,* 83(2), pp.283-296.

Feroldi, D., Serra, M. & Riera, J. (2007) Performance improvement of a PEMFC system controlling the cathode outlet air flow. *Journal of Power Sources,* 169(1), pp.205-212.

Finan, R.A., Sapeluk, A.T. & Damper, R.I. (1996) Comparison of multilayer and radial basis function neural networks for text-dependent speaker recognition. *Neural Networks, 1996., IEEE International Conference on*. 3-6 Jun 1996 v.4 pp.1992-1997.

Forbes, M.G., Patwardhan, R.S., Hamadah, H. & Gopaluni, R.B. (2015) Model predictive control in industry: Challenges and opportunities. *IFAC-PapersOnLine,* 48(8), pp.531-538.

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

Gomm, J.B. & Yu, D.L. (2000) Selecting radial basis function network centers with recursive orthogonal least squares training. *IEEE Transactions on Neural Netwoks,* 11(2), pp.306-314.

Griva, I., Nash, S.G. & Sofer, A. (2009) *Linear and nonlinear optimization.* Siam.

Grondin-Perez, B., Roche, S., Lebreton, C., Benne, M., Damour, C. & Kadjo, J.-J.A. (2014) Mechanistic model versus artificial neural network model of a single-cell PEMFC. *Engineering,* 6(8), pp.418-426.

Grosdidier, P., Froisy, B. & Hamman, M. (1988) The IDCOM-M controller. *IFAC workshop on model based process control*. Oxford. pp.31-36.

Grötsch, M. (2010) *Nonlinear Analysis and Control of PEM Fuel Cells.* Ph.D thesis, Otto-von-Guericke-Universität Magdeburg.

Gu, L., Tok, D.K.S. & Yu, D.-L. (2016) Development of adaptive p-step RBF network model with recursive orthogonal least squares training. *Neural Computing and Applications*, pp.1-10.

Guenin, B., Könemann, J. & Tunçel, L. (2014) *A gentle introduction to optimization.* Cambridge University Press.

Guerreiro, B., Silvestre, C. & Cunha, R. (2008) Terrain avoidance model predictive control for autonomous rotorcraft. *IFAC Proceedings Volumes,* 41(2), pp.1076-1081.

Guo, A., Chen, W., Li, Q., Liu, Z. & Que, H. (2013) Air flow control based on optimal oxygen excess ratio in fuel cells for vehicles. *Journal of Modern Transportation,* 21(2), pp.79-85.

Hafidi, G., Tebbani, S., Dumur, D. & Vande Wouwer, A. (2008) Nonlinear model predictive control applied to e. coli cultures. *IFAC Proceedings Volumes,* 41(2), pp.14570-14575.

Han, H.-G., Chen, Q.-L. & Qiao, J.-F. (2011) An efficient self-organizing RBF neural network for water quality prediction. *Neural Networks*, 24(7), pp.717-725.

Han, H.-G., Wu, X.-L. & Qiao, J.-F. (2013) Real-Time Model Predictive Control Using a Self-Organizing Neural Network. *IEEE Transactions on Neural Netwoks and Learning Systems,* 24(9), pp.1425-1436.

Hao, Y., Xie, T., Paszczynski, S. & Wilamowski, B.M. (2011) Advantages of radial basis function networks for dynamic system design. *IEEE Transactions on Industrial Electronics,* 58(12), pp.5438-5450.

Haseltine, E.L. & Rawlings, J.B. (2005) Critical evaluation of extended Kalman filtering and moving-horizon estimation. *Industrial & Engineering Chemistry Research*, 44(8), pp.2451-2460.

Haverbeke, N., Van Herpe, T., Diehl, M., Van Den Berghe, G. & De Moor, B. (2008) Nonlinear model predictive control with moving horizon state and disturbance estimation - Application to the normalization of blood glucose in the critically ill. *IFAC Proceedings Volumes,* 41(2), pp.9069-9074.

Herrera, L.J., Pomares, H., Rojas, I., Guillén, A., Rubio, G. & Urquiza, J. (2011) Global and local modelling in RBF networks. *Neurocomputing,* 74(16), pp.2594-2602.

Holkar, K. & Waghmare, L. (2010) An overview of model predictive control. *International Journal of Control and Automation,* 3(4), pp.47-63.

Hong, X. & Billings, S.A. (1997) Givens rotation based fast backward elimination algorithm for RBF neural network pruning. *IEE Proceedings-Control Theory and Applications,* 144(5), pp.381-384.

Hou, K.-H., Lin, C.-H., Ger, M.-D., Shiah, S.-W. & Chou, H.-M. (2012) Analysis of the characterization of water produced from proton exchange membrane fuel cell (PEMFC) under different operating thermal conditions. *International Journal of Hydrogen Energy,* 37(4), pp.3890-3896.

Huang, G.-B., Saratchandran, P. & Sundarajan, N. (2005) A generalized growing and pruning RBF (GGAP-RBF) neural network for function approximation. *IEEE Transactions on Neural Netwoks*, 16(1), pp.57-67.

Huusom, J.K., Poulsen, N.K., Jørgensen, S.B. & Jørgensen, J.B. (2010) ARX-model based model predictive control with offset-free tracking. *Computer Aided Chemical Engineering,* 28, pp.601-606.

Huusom, J.K., Poulsen, N.K., Jørgensen, S.B. & Jørgensen, J.B. (2012) Tuning SISO offset-free model predictive control based on ARX models. *Journal of Process Control,* 22(10), pp.1997-2007.

Jayawardena, A., Fernando, D. & Zhou, M. (1997) Comparison of multilayer perceptron and radial basis function networks as tools for flood forecasting. *IAHS Publications-Series of Proceedings and Reports-Intern Assoc Hydrological Sciences*, 239, pp.173-181.

Jo, Y.Y., Cho, E.A., Kim, J.H., Lim, T.-H., Oh, I.-H., Jang, J.H. & Kim, H.-J. (2010) Effects of a hydrogen and air supply procedure on the performance degradation of PEMFCs. *International Journal of Hydrogen Energy,* 35(23), pp.13118-13124.

Johansen, T. (2011) Introduction to nonlinear model predictive control and moving horizon estimation. *In:* Huba, M., Skogestad, S., Fikar, M., Hovd, M. & Rohal'-Ilkiv, B. (eds.) *Selected topics on constrained and nonlinear control.* Slovakia.

Kalpesh, L. (2004) *Development of QP algorithms for real-time predictive control systems.* Masters thesis, University of Applied Science Darmstadt.

Kaminski, W. & Strumillo, P. (1997) Kernel orthonormalization in radial basis function neural networks. *IEEE Transactions on Neural Networks,* 8(5), pp.1177-1183.

Karayiannis, N.B. & Mi, G.W. (1997) Growing radial basis neural networks: merging supervised and unsupervised learning with network growth techniques. *IEEE Transactions on Neural Netwoks*, 8(6), pp.1492-1506.

Kon, J. & Yamashita, Y. (2010) Model predictive control based on ARX models. *ICCAS 2010*. 27-30 October pp.448-452.

Korenberg, M., Billings, S., Liu, Y. & Mcilroy, P. (1988) Orthogonal parameter estimation algorithm for non-linear stochastic systems. *International Journal of Control,* 48(1), pp.193-210.

Kumar, M. & Yadav, N. (2011) Multilayer perceptrons and radial basis function neural network methods for the solution of differential equations: A survey. *Computers & Mathematics with Applications,* 62(10), pp.3796-3811.

Leonard, J.A. & Kramer, M.A. (1991) Radial basis function networks for classifying process faults. *IEEE Control Systems,* 11(3), pp.31-38.

Leyffer, S. & Mahajan, A. (2010) Software for nonlinearly constrained optimization. *Wiley Encyclopedia of Operations Research and Management Science*.

Li, S., Lim, K.Y. & Fisher, D.G. (1989) A state space formulation for model predictive control. *AIChE Journal,* 35(2), pp.241-249.

Li, Y., Sundararajan, N. & Saratchandran, P. (2001) Neuro-controller design for nonlinear fighter aircraft maneuver using fully tuned RBF networks. *Automatica,* 37(8), pp.1293-1301.

Lightbody, G. & Irwin, G.W. (1997) Nonlinear control structures based on embedded neural system models. *IEEE Transactions on Neural Networks*, 8(3), pp.553-567.

Lin, C.L., Wang, J.F., Chen, C.Y., Chen, C.W. & Yen, C.W. (2009) Improving the generalization performance of RBF neural networks using a linear regression technique. *Expert Systems with Applications,* 36(10), pp.12049-12053.

Ljung, L. (1999) *System identification: theory for the user.* 2nd ed.: Prentice-Hall.

Lowe, D. (1989) Adaptive radial basis function nonlinearities, and the problem of generalisation. *Artificial Neural Networks, 1989., First IEE International Conference on (Conf. Publ. No. 313).* 16-18 Oct 1989 pp.171-175.

Lu, Y., Sundarajan, N. & Saratchandran, P. (1998) Performance evaluation of a sequential minimal radial basis function (RBF) neural network learning algorithm. *IEEE Transactions on Neural Networks,* 9(2), pp.308-318.

Lu, Y., Sundararajan, N. & Saratchandran, P. (1997) A sequential learning scheme for function approximation using minimal radial basis function neural networks. *Neural Computation,* 9, pp.461-478.

Luenberger, D.G. (1984) *Linear and nonlinear programming.* 2nd ed. Reading, Mass: Addison-Wesley.

Magni, L. & Scattolini, R. (2004) Stabilizing model predictive control of nonlinear continuous time systems. *Annual Reviews in Control,* 28(1), pp.1-11.

Mashor, M.Y. (2000) Hybrid training algorithm for RBF network. *International Journal of the Computer, the Internet and Management,* 8(2), pp.50-65.

Mayne, D.Q. (2014) Model predictive control: recent developments and future promise. *Automatica,* 50(12), pp.2967-2986.

Mayne, D.Q., Rawlings, J.B., Rao, C.V. & Scokaert, P.O.M. (2000) Constrained model predictive control: stability and optimality. *Automatica,* 36(6), pp.789-814.

Mehta, S., Ramani, H., Yelgatte, N.N. & Rahman, I. (2016) Recursive orthogonal least square based soft sensor for batch distillation. *Chemical Product and Process Modeling,* 11(3), pp.241-263.

Mills, A., Wills, A. & Ninness, B. (2009) Nonlinear model predictive control of an inverted pendulum. *2009 American Control Conference*. pp.2335-2340.

Mohd, N. & Aziz, N. (2015) Control of bioethanol fermentation process: NARX-based MPC (NARX-MPC) versus linear-based MPC (LMPC). *Chemical Engineering Transactions,* 45, pp.1297-1302.

Moody, J. & Darken, C.J. (1989) Fast learning in networks of locally-tuned processing units. *Neural Computation,* 1(2), pp.281-294.

Morari, M. & Lee, J.H. (1999) Model predictive control: past, present and future. *Computers & Chemical Engineering,* 23(4), pp.667-682.

Morningred, J.D., Paden, B.E., Seborg, D.E. & Mellichamp, D.A. (1990) An adaptive nonlinear predictive controller. *1990 American Control Conference*. 23-25 May pp.1614-1619.

Morshedi, A.M., Cutler, C.R. & Skrovanek, T.A. (1985) Optimal solution of dynamic matrix control with linear programing techniques (LDMC). *1985 American Control Conference*. 19-21 June pp.199-208.

Muddu, M., Narang, A. & Patwardhan, S.C. (2010) Reparametrized ARX models for predictive control of staged and packed bed distillation columns. *Control Engineering Practice,* 18(2), pp.114-130.

Narendra, K.S. & Parthasarathy, K. (1990) Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks,* 1(1), pp.4-27.

Nelles, O. & Isermann, R. (1995) Identification of nonlinear dynamic systems classical methods versus radial basis function networks. *American Control Conference, Proceedings of the 1995*. Seattle, WA. 21-23 June v.5 pp.3786-3790.

Panos, C., Kouramas, K.I., Georgiadis, M.C. & Pistikopoulos, E.N. (2011) Modelling and explicit model predictive control for PEM fuel cell systems. *Chemical Engineering Science,* 67, pp.15-25.

Platt, J. (1991) A resource-allocating network for function interpolation. *Neural Computation,* 3(2), pp.213-225.

Poggio, T. & Girosi, F. (1990) Networks for approximation and learning. *Proceedings of the IEEE,* 78(9), pp.1481-1497.

Prett, D.M. & Gillette, R. (1980) Optimization and constrained multivariable control of a catalytic cracking unit. *Joint Automatic Control Conference*. v.17 pp.73.

Propoi, A.I. (1963) Use of LP methods for synthesizing sampled-data automatic systems. *Automation and Remote Control*, 24, pp.837.

Pukrushpan, J.T., Stefanopoulou, A.G. & Huei, P. (2002) Modeling and control for PEM fuel cell stack system. *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)*. 8-10 May v.4 pp.3117-3122.

Pukrushpan, J.T., Stephanopoulou, A.G. & Peng, H. (2004) *Control of fuel cell power systems. principles, modelling, analysis and feedback design.* London: Springer.

Puranik, S.V., Keyhani, A. & Khorrami, F. (2010) Neural network modeling of proton exchange membrane fuel cell. *IEEE Transactions on Energy Conversion,* 25(2), pp.474-483.

Qiao, J.-F. & Han, H.-G. (2012) Identification and modeling of nonlinear dynamical systems using a novel self-organizing RBF-based approach. *Automatica,* 48(8), pp.1729-1734.

Qin, S.J. & Badgwell, T.A. (2003) A survey of industrial model predictive control technology. *Control Engineering Practice,* 11(7), pp.733-764.

Qin, Y., Peng, H., Zhou, F., Zeng, X. & Wu, J. (2015) Nonlinear modeling and control approach to magnetic levitation ball system using functional weight RBF network-based state-dependent ARX model. *Journal of the Franklin Institute,* 352(10), pp.4309-4338.

Rezazadeh, A., Sedighizadeh, M. & Karimi, M. (2010) Proton exchange membrane fuel cell control using a predictive control based on neural network. *International Journal of Computer and Electrical Engineering,* 2(1), pp.81-85.

Richalet, J., Estival, J.L. & Fiani, P. (1995) Industrial applications of predictive functional control to metallurgical industries. *Proceedings of the 4th IEEE Conference on Control Applications*. 28-29 September pp.934-942.

Richalet, J., Rault, A., Testud, J.L. & Papon, J. (1978) Model predictive heuristic control: applications to industrial processes. *Automatica,* 14, pp.413-428.

Rodríguez, M. & Pérez, D. (2005). First principles model based control. *In:* Puigjaner, L. & Espuña, A. (eds.) *Computer Aided Chemical Engineering.* Elsevier.

Rohendi, D., Majlan, E.H., Mohamad, A.B., Daud, W.R.W., Kadhum, A.a.H. & Shyuan, L.K. (2015) Effects of temperature and backpressure on the performance degradation of MEA in PEMFC. *International Journal of Hydrogen Energy,* 40(34), pp.10960-10968.

Rossiter, J.A. (2003) *Model-based predictive control: a practical approach.* CRC press.

S. Elanayar, V.T. & Shin, Y.C. (1994) Radial basis function neural network for approximation and estimation of nonlinear stochastic dynamic systems. *IEEE Transactions on Neural Networks,* 5(4), pp.594-603.

Samek, D. & Dostal, P. (2009) Artificial neural network with radial basis function in model predictive control of chemical reactor. *Mechanics,* 28(3), pp.91-95.

Seborg, D.E., Mellichamp, D.A., Edgar, T.F. & Doyle Iii, F.J. (2010) *Process dynamics and control.* John Wiley & Sons.

Sedighizadeh, M., Rezaei, M. & Najmi, V. (2011) A predictive control based on neural network for proton exchange membrane fuel cell. *International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering,* 5(2), pp.188-192.

Seshagiri, S. & Khalil, H.K. (2000) Output feedback control of nonlinear systems using RBF neural networks. *IEEE Transactions on Neural Networks,* 11(1), pp.69-79.

Sheng, S., Xianzhong, D., Chan, W.L. & Zhihuan, L. (2009) Erroneous measurement detection in substation automation system using OLS based RBF neural network. *International Journal of Electrical Power & Energy Systems,* 31(7–8), pp.351-355.

Shevade, S. (2012). Numerical Optimization [Online]. lecture notes, NPTEL. Available: http://nptel.ac.in/courses/106108056/.

Shi, Y., Yu, D.-L., Tian, Y. & Shi, Y. (2015) Air–fuel ratio prediction and NMPC for SI engines with modified Volterra model and RBF network. *Engineering Applications of Artificial Intelligence*, 45, pp.313-324.

Su, T.H. & Mcavoy, T.J. (1997). Artificial neural network for nonlinear process identification and control. *In:* Henson, M.A. & Seborg, D.E. (eds.) *Nonlinear process control.* Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

Sun, T.-Y., Liu, C.-C., Lin, C.-L., Hsieh, S.-T. & Huang, C.-S. (2009) A radial basis function neural network with adaptive structure via particle swarm optimization. *In:* Lazinica, A. (ed.) *Particle swarm optimization.* InTech.

Tajjudin, N., Ismail, N., Rahiman, M.H.F. & Taib, M.N. (2010) Model predictive control using ARX model for steam distillation essential oil extraction system. *Intelligent and Advanced Systems (ICIAS), 2010 International Conference on*. Kuala Lumpur, Malaysia. 15-17 June pp.1-5.

Tok, D.K.S., Yu, D.-L., Mathews, C., Zhao, D.-Y. & Zhu, Q.-M. (2015) Adaptive structure radial basis function network model for processes with operating region migration. *Neurocomputing,* 155, pp.186-193.

Vahidi, A., Stefanopoulou, A. & Huei, P. (2004) Model predictive control for starvation prevention in a hybrid fuel cell system. *American Control Conference, 2004. Proceedings of the 2004*. Boston, MA, USA. June 30-July 2 v.1 pp.834-839.

Vanderbei, R.J. (2015) *Linear programming.* Springer.

Wang, D., Zeng, X.-J. & Keane, J.A. (2012) A clustering algorithm for radial basis function neural network initialization. *Neurocomputing,* 77(1), pp.144-155.

Wang, D., Zhou, Y. & He, X. (2007) Radial basis function neural network-based model predictive control for freeway traffic systems. *International Journal of Intelligent Systems Technologies and Applications*, 2(4), pp.370-388.

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

Wang, S.W., Yu, D.L., Gomm, J.B., Page, G.F. & Douglas, S.S. (2006) Adaptive neural network model based predictive control for air–fuel ratio of SI engines. *Engineering Applications of Artificial Intelligence,* 19(2), pp.189-200.

Wilson, J., Charest, M. & Dubay, R. (2016) Non-linear model predictive control schemes with application on a 2 link vertical robot manipulator. *Robotics and Computer-Integrated Manufacturing,* 41, pp.23-30.

Wu, Q., Wang, X. & Shen, Q. (2016) Research on dynamic modeling and simulation of axial-flow pumping system based on RBF neural network. *Neurocomputing,* 186, pp.200-206.

Xie, T., Yu, H. & Wilamowski, B. (2011) Comparison between traditional neural networks and radial basis function networks. *2011 IEEE International Symposium on Industrial Electronics*. 27-30 June pp.1194-1199.

Ydstie, B.E., Kemna, A. & Liu, L. (1988) Multivariable extended-horizon adaptive control. *Computers & Chemical Engineering,* 12(7), pp.733-743.

Yu, D.-L., Yu, D.-W. & Gomm, J.B. (2006) Neural model adaptation and predictive control of a chemical process rig. *IEEE Transactions on Control Systems Technology,* 14(5), pp.828-840.

Yu, D.L. (2004) A localized forgetting method for Gaussian RBFN model adaptation. *Neural Processing Letters,* 20(2), pp.125-135.

Yu, D.L., Gomm, J.B. & Williams, D. (1997) A recursive orthogonal least squares algorithm for training RBF networks. *Neural Processing Letters,* 5(3), pp.167-176.

Yu, D.L. & Yu, D.W. (2007) A new structure adaptation algorithm for RBF networks and its application. *Neural Computing and Applications,* 16(1), pp.91-100.

Yu, D.L., Yu, D.W., Gomm, J. & Page, G.F. (2004) Adaptive RBF model for model-based control. *Intelligent Control and Automation, 2004. WCICA 2004. Fifth World Congress on*. 15-19 June v.1 pp.78-82.

Yuan, L., Zhao, H., Chen, H. & Ren, B. (2016) Nonlinear MPC-based slip control for electric vehicles with vehicle safety constraints. *Mechatronics,* 38, pp.1-15.

Yuan, Y.X. (1999). Problems on convergence of unconstrained optimization algorithms. *In:* Yuan, Y.X. (ed.) *Numerical linear algebra and optimization.* Beijing: Science Press.

Zadeh, L.A. & Whalen, B.H. (1962) On optimal control and linear programming. *IRE Transactions on Automatic Control,* 7(4), pp.45-46.

Zhang, L., Li, K., Bai, E.W. & Irwin, G.W. (2015) Two-stage orthogonal least squares methods for neural network construction. *IEEE Transactions on Neural Networks and Learning Systems,* 26(8), pp.1608-1621.

Zhang, Y.M. & Li, X.R. (1996) Hybrid training of RBF networks with application to nonlinear systems identification. *Decision and Control, 1996., Proceedings of the 35th IEEE Conference on*. 11-13 Dec. v.1 pp.937-942.

Zhao, N., Wen, X., Yang, J., Li, S. & Wang, Z. (2015) Modeling and prediction of viscosity of water-based nanofluids by radial basis function neural networks. *Powder Technology,* 281, pp.173-183.

Zhao, Z.-Q. & Huang, D.-S. (2007) A mended hybrid learning algorithm for radial basis function neural networks to improve generalization capability. *Applied Mathematical Modelling,* 31(7), pp.1271-1281.

Zhou, P., Li, D., Wu, H. & Chen, F. (2010) A novel OLS algorithm for training RBF neural networks with automatic model selection. *2010 International Conference on Computer Application and System Modeling (ICCASM 2010).* v.10 pp.40-44.

Zulkeflee, S.A., Aziz, N. & Sata, S.A. (2011) Nonlinear autoregressive with exogenous inputs based model predictive control for batch citronellyl laurate esterification reactor. *In:* Zheng, T. (ed.) *Advanced model predictive control.* InTech.

APPENDIX A – CSTR Parameters

product concentration	C_a	0.1 mol/l
reactor temperature	T	438.54 K
coolant flow rate	q_c	103.41 l/min
process flow rate	q	100 l/min
feed concentration	C_{a0}	1 mol/l
inlet coolant temperature	T_{c0}	350 K
CSTR volume	v	100 l
heat transfer coefficient	h_a	7 x 10^5 cal/min/K
reaction rate constant	k_o	7.2 x 10^{10} min^{-1}
activation energy term	E/R	1 x 10^4 K
heat of reaction	ΔH	-2 x 10^5 cal/mol
liquid densities	ρ, ρ_c	1 x 10^3 g/l
specific heats	C_p, C_{pc}	1 cal/g/k

$$k_1 = -\frac{\Delta H k_0}{\rho C_p} \qquad k_2 = \frac{\rho_c}{\rho C_p v} \qquad k_3 = \frac{h_a}{\rho_c C_{pc}}$$

APPENDIX B – List of Publications

Published Papers

1. Tok, D.K.S., Yu, D.-L., Mathews, C., et al. (2015) Adaptive structure radial basis function network model for processes with operating region migration. *Neurocomputing*, 155**,** pp.186-193.

2. Tok, D.K.S., Yu, D.-L. & Mathews, C. (2015) A model adaptation algorithm for radial basis function networks using recursive orthogonal least square algorithm. *Research Week 2015*, Liverpool John Moores University.

3. Gu, L., Tok, D.K.S. & Yu, D.-L. (2016) Development of adaptive p-step RBF network model with recursive orthogonal least squares training. *Neural Computing and Applications*, pp.1-10.

Submitted Papers

1. Tok, D.K.S. & Yu, D.-L. A new f-step radial basis function network model for model predictive control, Submitted to *Neurocomputing*, May 2016

2. Tok, D.K.S., Gu, L. & Yu, D.-L. Factorized f-step RBF neural network-based model predictive control for oxygen excess ratio of PEM fuel cells, Submitted to *International Journal of Hydrogen Energy*, September 2016