

ENHANCEMENTS FOR MODEL PREDICTIVE
CONTROL AND INFERENTIAL
MEASUREMENT

By

SHARAD BHARTIYA

Bachelor of Engineering
Regional Engineering College
Durgapur, India
1991


Master of Technology
Indian Institute of Technology
Madras, India
1993

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
July, 2000

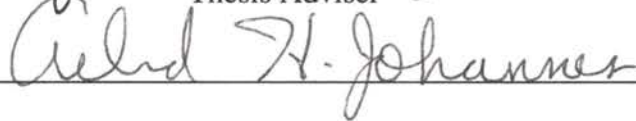
Thesis
2000
B575e

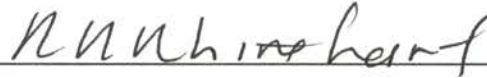
ENHANCEMENTS FOR MODEL PREDICTIVE
CONTROL AND INFERENTIAL
MEASUREMENT

Thesis Approved:



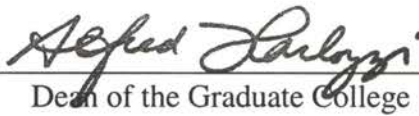
Thesis Adviser











Dean of the Graduate College

ACKNOWLEDGEMENTS

It is a pleasure to thank a number of special persons who contributed directly or indirectly towards my efforts aimed at graduate study. I begin by expressing my deep sense of appreciation and gratitude to my parents for making learning and education a worthy goal and my grandmother for being my first teacher.

Dr. Hagan, Dr. High, Dr. Johannes, Dr. Misawa, Dr. Rhinehart and Dr. Robinson have been very gracious in serving on my dissertation committee and I whole-heartedly thank them. I owe a special note of thanks to my adviser, Dr. Whiteley, for providing guidance and research opportunities. I also thank him and the Department of Chemical Engineering for providing me with financial support.

I take this opportunity to thank all of my teachers who inspired me to take a closer look at arguments presented in the classroom. Dr. J.R. Whiteley and Dr. R.R. Rhinehart trained me to think in terms of control issues relevant to process industry practice. Dr. Whiteley helped me to stay focussed on the bigger picture of utility of process control and encouraged me to work on control of the Eastman problem presented in Chapter 6. The hot/cold mixing problem used in Chapter 5 is due to Dr. Rhinehart.

Dr. Hagan and Dr. Yen introduced me to neural networks. This made possible the work presented in Chapter 4 and Chapter 5. I gratefully acknowledge the corporate Process Technology and Optimization group of Conoco, Inc. for providing data from a refinery. In particular, discussions with Paul Belcher and Paul Priba from that group were

instrumental in developing the correlation presented in Chapter 4. Dr. Hagan also shed light on the theories of stochastic systems and their identification. This is the theme of Chapter 3. I thank Dr. Misawa for concretizing abstract notions in nonlinear control by providing memorable responses to my queries. I would also like to express my gratitude to Prof. N.L. Ricker and his group for use of their MATLAB simulation of the Eastman process.

I thank my sister, brother, and friends for their encouragement, support and kindness extended to me over the years.

Finally, I dedicate this work in the memory of my friend, Arindam, in whose company the initial idea of pursuing higher studies was formulated.

TABLE OF CONTENTS

Chapter	Page
1. INTRODUCTION	1
1.1 Background.....	1
1.2 Objectives and Contributions	2
1.3 Organization of Thesis.....	4
2 LITERATURE REVIEW	6
2.1 MPC – Introduction.....	7
2.2 MPC – Origin.....	11
2.3 MPC and standard Linear Quadratic Regulator	12
2.4 Linear MPC – Developments in industry.....	13
2.5 Linear MPC – Theoretical Aspects	17
2.6 Adaptive Model Predictive Control.....	21
2.7 Nonlinear Model Predictive Control.....	25
2.8 Neural Network Models in Process Control.....	29
3 ADAPTIVE QUADRATIC DYNAMIC MATRIX CONTROL	34
3.1 Introduction.....	34
3.2 Recursive Identification and Model Parameterization	40
3.3 <i>A Priori</i> Information And System Identification.....	45
3.4 Adaptive Quadratic Dynamic Matrix Control	47
3.5 Simulation Examples	57
3.6 Conclusions	70
4 DEVELOPMENT OF INFERENTIAL MEASUREMENTS USING NEURAL NETWORKS	72
4.1 Introduction.....	73
4.2 Methodology	75
4.3 An Example from Petroleum Refinery.....	87
4.4 Identification and Collection of Candidate Independent Variables and Preprocessing of Data	91

4.5	Identification of a Suitable Subset for Regression.....	96
4.6	Regression Using Neural Networks	103
4.7	Conclusions	114
5	A FACTORIZED APPROACH TO NONLINEAR MPC USING A RADIAL BASIS FUNCTION PROCESS MODEL.....	115
5.1	Introduction.....	116
5.2	Nonlinear System Identification Using Neural Networks	117
5.3	Dynamic Modeling Using RBF network.....	122
5.4	MPC Using p-step Control Model	131
5.5	Simulation Examples	138
5.6	Conclusions.....	147
6	APPLICATION OF THE FACTORIZED RBF BASED MPC TO THE EASTMAN CHALLENGE PROBLEM	149
6.1	Introduction.....	149
6.2	RBF based nonlinear MPC Algorithm	154
6.3	Overview of Eastman Process.....	159
6.4	Application of RBF-Based NMPC to Eastman Problem	162
6.5	NMPC Controller Performance.....	171
6.6	Conclusions.....	186
7	CONCLUSIONS AND RECOMMENDATIONS	188
7.1	Conclusions.....	188
7.2	Recommendations.....	189
	BIBLIOGRAPHY	194
	APPENDIX A -- QUADRATIC DYNAMIC MATRIX CONTROL.....	205
	APPENDIX B -- RECURSIVE LEAST SQUARES ALGORITHM.....	212
	APPENDIX C -- QUADRATIC PROGRAMMING USING ACTIVE SET METHOD	217

LIST OF TABLES

Table	Page
4.1: Variables determined to be significant influences on gap/overlap between kerosene and HOD as identified by scatter plots, simple and partial correlation coefficients.....	99
4.2: Variables determined to be significant influences on ASTM 95% endpoint of kerosene as identified by scatter plots, simple, and partial correlation coefficients.....	102
4.3: Mallow Statistic for certain Subsets of Variables of Table 4.1.....	103
4.4: Mallow Statistic for certain Subsets of Variables of Table 4.2.....	104
4.5: Performance measures for neural network NN (rearranged scheme of work) when different number of neurons are used in the hidden layer. Also the results of the revised scheme of work are shown.....	113
5.1: RBF model summary for hot/cold mixing example.....	127
5.2: Comparison of computation time needed in control of hot/cold mixing example....	142
6.1: Input-output pairing determined by McAvoy and Ye, (McAvoy and Ye, 1994).....	163
6.2: Controlled and manipulated variables for control by the RBF based MPC controller.....	165
6.3: Operating region represented in network training.....	167
6.4: Elements of RBF input pattern vector x_k	167
6.5: Training and test set error statistics.....	168
6.6: Weights used in the RBF based MPC simulations.....	170
6.7: Comparison of computation time needed for implementation of a setpoint change of -60 kPa in reactor pressure (Figure 6.6).	184

LIST OF FIGURES

Figure	Page
2.1: Schematic description of the linear MPC algorithm	8
2.2: Role of model predictive control from process operation perspective.	10
3.1: Block diagram of adaptive model predictive control system.	39
3.2: Use of estimated parameters is first made when they converge. The step response coefficients are calculated after this point whenever new converged estimates become available. Parameter adaptation is stopped when the data are no longer exciting.	54
3.3: Adaptive QDMC algorithm.....	56
3.4: Example 1: Comparison of AQDMC and QDMC algorithms used to control a SISO process. Both the process and the model structure used by AQDMC are as shown in equation (3.5). The time delay of the process is known.	59
3.5: Example 2: Performance of the AQDMC algorithm in presence of white noise.	61
3.6: Example 3: Comparison of AQDMC and QDMC algorithms used in the control of a 2 x 2 process. At $k = 0$, the setpoint for output 1 is set to 0.02. At $k = 101$, the setpoint is stepped back to 0.0.	64
3.7: Example 3: Model parameter estimation by the adaptive QDMC algorithm: a) Estimation trajectory of some of the model parameters; b) Step response of the estimated model at $k = 198$	65
3.8: Example 4: Steady-state behavior of CSTR for $\beta = 3.0$, $\gamma = 40.0$, $B = 22$ and $Da = 0.082$	67
3.9: Example 4: Comparison of adaptive DMC and non-adaptive DMC performances for control of nonlinear CSTR. The nominal model was developed for low conversions (1 to 4% conversion). The DMC system response is unbounded for operation in high conversion regions while the adaptive DMC controller modifies controller parameters to reflect larger gain in the high conversion region.....	69

4.1: Schematic of the matrix of collected data. The first column refers to the inferred variable. The subsequent columns contain measurements of candidate independent variables. Augmented variables refer to variables such as enthalpy, product yield, etc. which cannot be measured but calculated using direct measurements.....	77
4.2: Architecture of multilayer perceptron neural network used in construction of inferential model. The inputs to the network are determined in sub-problem (b). Sub-problem (c) determines the optimal number of nodes in hidden layer.....	85
4.3: A schematic of a typical atmospheric distillation tower. Shown are two feed streams that enter the tower. Side streams are often fed to side-strippers to obtain intermediate products between the overhead product and bottoms.	88
4.4: Illustration of regression scheme. Neural network NN1 predicts gap/overlap. Network NN2 uses the predicted gap/overlap as one of its inputs among others to estimate the unmeasured variable, ASTM 95% endpoint of kerosene.....	94
4.5: Scatter plot depicting relationship between kerosene/HOD gap/overlap and pressure differential across atmospheric tower (PD1 in Figure 4.1).	97
4.6: Scatter plot depicting relationship between ASTM 95% endpoint of kerosene and kerosene/HOD gap/overlap.....	100
4.7: Performance of network NN1 when trained using trimmed data. The trimming of outliers was performed to refine regression (section 4.2.3.2). Five neurons are used in hidden layer.	106
4.8: Performance of network NN2. The input gap/overlap is predicted by NN1 when it is trained using the trimmed data set (Figure 4.7). Four neurons are used in hidden layer.....	107
4.9: Performance of NN2 when trained by data set trimmed to refine regression. The input values of gap/overlap used to evaluate performance are the SimDist measurements and do not represent prediction by NN1 as in Figure 4.8.	108
4.10: The top schematic shows the configuration of the inferential correlation in the original scheme. The inputs to the network were identified using procedures described in sub-problem (b). The bottom schematic shows the revised scheme of work with identified inputs. The inputs to NN1 and NN2 form the inputs to network NN (except the intermediate variable, gap/overlap).....	110
4.11: Performance of network NN when five neurons are used in hidden layer. The inputs and output are shown schematically in Figure 4.10. Performance measures are shown in Table 4.5.	112

5.1: Timelines showing inputs to the p -step control model. For this example, $p = 4$, $N_y = 3$, and $N_u = 2$. Predicted outputs are generated from known information only, previous model predictions for y are not used in model input.....	123
5.2: Schematic diagram of the hot/cold mixing process.....	126
5.3: Comparison of cascaded 1-step predictions with the p -step model ($p = 8$).....	128
5.4: Comparison of steady state process output with RBF model predictions. Summary of RBF network is provided in Table 5.1. The signal to hot leg valve was maintained at a fixed value of 25%. The cold leg valve input was varied in steps of 5%. Each RBF prediction was obtained in one computational step.	130
5.5: Performance of RBF model for approximation of dynamic response. The two cases shown depict response of temperature to steps in cold leg valve from 50% to 75% and from 50% to 25%. The hot leg valve was maintained at 25%.	132
5.6: (a) and (b) illustrate control of the 2x2 mixing process in presence of measured disturbances in hot leg (at $k = 50, 200$, and 375) and cold leg (at $k = 125, 200$, and 375) temperatures by the RBF based NMPC and linear QDMC. Setpoint changes were made at $k=175, 375, 575$ and 775 for temperature and at $k=175, 375$ and 775 for flowrate. (c) and (d) show control action implemented by the NMPC and QDMC controllers. (e) and (f) process model mismatch for temperature and flowrate	140
5.7: (a) and (b) illustrate the control of the 2x2 mixing process in presence of unmeasured disturbances in hot and cold leg temperatures (steps, drifts and spikes). Setpoint changes were made at $k=0, 65$ and 365 for temperature and at $k=0, 200$, and 365 for flowrate. (c) control action implemented by the NMPC controller. (d) process model mismatch for temperature and flowrate.....	144
5.8: (a) shows the steady state characteristics of the CSTR process. (b) illustrates the control of the CSTR process using a linear QDMC controller and the RBF based NMPC algorithm. Setpoint changes were made at $k=200, 400$, and 600 . The linear QDMC control system becomes unstable in high gain region ($k>646$). The RBF-NMPC controller provides tight control in both, the low and high gain regions. (c) control action. (d) process model mismatch.....	146
6.1: Timelines showing inputs to the p -step control model. For this example, $p = 4$, $N_y = 3$, and $N_u = 2$. Predicted outputs are generated from known information only, previous model predictions for y are not used in model input.....	157
6.2: Schematic of the Eastman challenge process.....	161
6.3: Comparison of RBF model predictions with plant measurements for step changes in A & C feed rate. All other variables are maintained at their base values. After	

5 hours, the A & C feed rate is brought to 6 m ³ /h, which lies outside the lower limit of training data of 7.5m ³ /h.	169
6.4: Product flowrate setpoint change (-15%).....	172
6.5: Product G/H ratio setpoint change from 50/50 to 40/60.....	173
6.6: Reactor pressure setpoint change (-60 kPa).	174
6.7: Purge B composition setpoint change (+2 mol%).....	175
6.8: Process-model mismatch during implementation of +2% setpoint change in composition of component B in purge stream.	177
6.9: Disturbance IDV(1) (step change in A/C feed ratio in A & C stream).....	178
6.10: Process-model mismatch during occurrence of disturbance IDV(1). The unmeasured disturbance is not modeled by the RBF network, leading to poor predictions. However, due to the step nature of the disturbance, the process-model mismatches settle to near-steady values.	179
6.11: The step disturbance IDV(1) is detected by the drop in the controlled variable, A/C ratio in reactor feed, from its setpoint. The variable is then brought back to its setpoint by increasing A feed rate.	180
6.12: Disturbance IDV(8) (random variation in A, B, C compositions in A & C stream).....	182
6.13: Process-model mismatch during occurrence of disturbance IDV(8). The unmeasured disturbance is not modeled by the RBF network, leading to poor predictions. The random nature of the disturbance leads to random variations in process-model mismatch.	183

1 INTRODUCTION

1.1 Background

Market competition and stringent safety and environmental concerns have led to development of strategies to improve the efficiency of process operations and reduce operating costs. A significant portion of project costs includes developing, installing and maintaining advanced process control systems. Ramaker et al. (1997) list the following economic motivations that exist for contribution by process control:

- (1) use existing process equipment fully,
- (2) deliver the same product consistently,
- (3) minimize product variability,
- (4) meet safety or regulatory requirements,
- (5) increase the operator's span of control,
- (6) reduce the cost of implementing and supporting control and information systems,
- (7) improve the operating range and reliability of control and information systems.

Recent developments in process control technology have been directed at satisfying some of these stipulations. These advances are usually guided by the philosophy that an accurate description of future process response increases the potential for producing desired process behavior. In this work, we utilize causal, empirical

models, such as time series and neural networks to model the process behavior and explore their use in model predictive control algorithms and inferential modeling applications.

1.2 Objectives and Contributions

The work documented in this thesis can be classified into 3 categories:

- Adaptive linear model predictive control,
- Inferential modeling using neural networks, and
- Radial basis function model based nonlinear model predictive control

The objectives pursued and the contributions made by the present work within each category are described below.

(a) Adaptive Linear Model Predictive Control:

Model Predictive Control (MPC) algorithms share the common characteristic of using an explicit model of the process to predict future behavior over a specified horizon. The dependence of MPC techniques on model fidelity presents an ideal opportunity for adaptation of model parameters. The potential to estimate accurate models for the current operating conditions forms the prime motivation for an adaptive MPC scheme.

In this work, an adaptive linear MPC algorithm is developed that utilizes closed-loop process data to construct control models online. These models are subsequently used by the controller. Model parameters are adjusted by a recursive least squares algorithm. Benefits of using adaptation are shown using simulation examples. We also

demonstrate closed loop identifiability of single input, single output transfer function models used in conjunction with an industrially important form of linear MPC known as Dynamic Matrix Control (DMC).

(b) *Inferential Modeling Using Neural Network:*

In numerous processes, measurement of key variables is not available or too slow to be included in online control and optimization calculations. There is considerable economic incentive in developing inferential sensors which provide an estimate of such hard-to-measure variables. The nonlinear nature of chemical processes makes neural networks a logical choice for modeling and prediction of these variables.

In this area, we present a framework for development of inferential measurements using neural networks. The method involves a three-step procedure. The first step consists of data collection and preprocessing. In the second step, the process variables are subjected to simple statistical analyses to identify a subset of measurements to be used in the inferential scheme. The third step involves generation of the inferential scheme by regression. For this purpose, the multi-layer perceptron network is employed. Finally, the methodology is demonstrated using real data from a large refinery to infer an ASTM property of a petroleum product.

(c) *Radial Basis Function Model based Nonlinear Model Predictive Control:*

Application of commercial linear model predictive control technology to highly nonlinear processes provides only partially successful results. This has led to an active

interest in the development and application of nonlinear model predictive control (NMPC). NMPC adheres to the general MPC philosophy but uses a nonlinear model to provide a better approximation of the underlying nonlinear system. The resulting implementation requires development of nonlinear models and an expensive online solution of a nonlinear program. These problems are severe enough that NMPC remains an unrealized concept in industry.

Here, the use of Radial Basis Function (RBF) networks as nonlinear process models in NMPC is explored. A novel RBF based NMPC algorithm is presented that is computationally efficient and provides enhanced control of nonlinear processes. The algorithm was tested by simulation for control of a mixing process and an exothermic CSTR. To evaluate the applicability of the algorithm for large processes, we successfully applied it for control of the Eastman challenge problem presented by Downs and Vogel (1993).

1.3 Organization of Thesis

This thesis was prepared using the manuscript format. Chapters 3-6 represent verbatim copies of manuscripts that have been submitted for publication in peer-reviewed journals.

Chapter 2 contains a literature survey on the various topics discussed above with an emphasis on model predictive control. Chapter 3 documents the work on adaptive linear MPC. Development of inferential models using neural networks is presented in

Chapter 4. Chapter 5 describes the RBF based NMPC algorithm. Application of the algorithm to the Eastman challenge process is presented in Chapter 6. Since Chapters 3, 4, 5, and 6 are in manuscript form, they are standalone in nature. Finally, conclusions based on the resulting work and avenues to improve upon the current work are presented in Chapter 7.

2 LITERATURE REVIEW

Advances in the computer industry have facilitated implementation of process control technology. This allowed replacement of conventional analog controllers by more flexible control algorithms. Among the various advanced control technologies, model predictive control (MPC) has been widely accepted in the process industries. In their vision of advanced information and control circa 2020, Ramaker et al. (1997) foresee computer control technology to use online economic information to dynamically maximize economic benefits, by adjusting the existing process equipment along with continuous process analysis. As discussed in Chapter 1, the work documented in this thesis explores and proposes enhancements to process control in the fields of model predictive control and inferential modeling and spans the following topics,

- linear and nonlinear MPC,
- adaptive linear MPC and system identification,
- multilayer perceptron and radial basis function neural networks, and
- inferential measurements

In this chapter, a review of developments in MPC and the use of neural network models for inferential measurement and MPC is provided.

Section 2.1 introduces the MPC algorithm. The subsequent four sections present an account of the various MPC algorithms and note some theoretical results available in the literature. Section 2.6 surveys the literature on adaptive MPC. Literature available on nonlinear MPC is discussed in section 2.7. Section 2.8 focuses on applications of

neural networks for nonlinear MPC and inferential measurement. Sections 2.6 and 2.8 also put the current work in perspective of that reported in literature.

2.1 MPC – Introduction

All MPC algorithms share a common philosophy, that is, use of an explicit model of the process to predict future behavior over a time interval called the prediction horizon. Process input and output constraints are directly incorporated in the algorithm. This allows for anticipation of constraint violation and hence an appropriate computation of input moves. The manipulated variable profile is computed via online optimization of an open-loop objective function subject to constraints. The first move of the resultant profile, corresponding to the current sample instant is implemented. The entire procedure is repeated at each sampling period (for example, see Muske and Rawlings, 1993). A schematic description of the traditional MPC algorithm is provided in Figure 2.1. The future duration of the forecast of process behavior is often referred to as the prediction horizon while the length of the manipulated variable profile is called control horizon.

The use of MPC in the process industries first began in the 1970s under the names of "model predictive heuristic control" or "model algorithmic control" (Richalet, et al., 1978; Mehra, et al., 1982) and "dynamic matrix control" (Cutler and Ramaker, 1979; Prett and Gillette, 1979). Since then MPC has been widely adopted as a high-performance, multivariable constrained control technique with commercial products supplied by a number of vendors. In addition to developing more flexible control

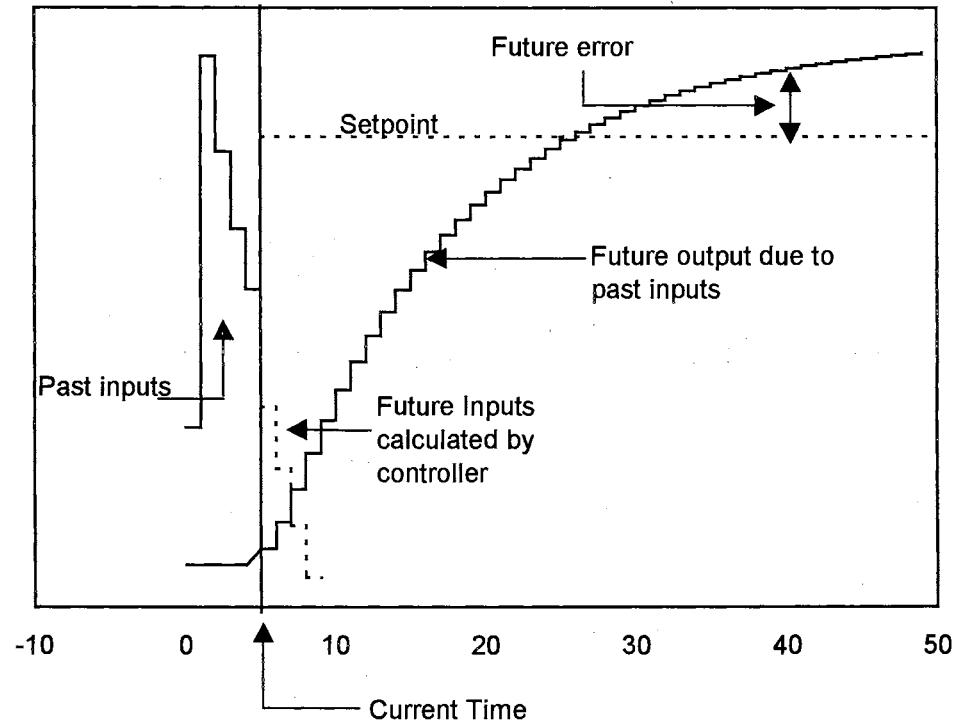


Figure 2.1: Schematic description of the linear MPC algorithm.

technology, new process identification technology was developed to allow quick estimation of empirical dynamic models from test data, substantially reducing the cost of model development. Qin and Badgwell (1997) refer to the combined effort of industrial process modeling and control as model predictive control technology. In a survey by Qin and Badgwell (1997) in 1995, over 2,200 applications of MPC were reported with the majority in the refining industry.

In modern processing plants the MPC controller is part of a multi-level hierarchy of control functions. This is illustrated in Figure 2.2. At the top of the structure, a plant-wide optimizer determines optimal steady-state settings for each unit in the plant. These may be sent to local optimizers at each unit, which run more frequently or consider a more detailed unit model than is possible at the plant-wide level. The unit optimizer computes an optimal economic steady state and passes the setpoints to the dynamic constraint control system for implementation. The dynamic constraint control must move the plant from one constrained steady state to another while minimizing constraint violations along the way. In the conventional structure this is accomplished by using a combination of PID algorithms, lead-lag blocks and high/low select logic. It is often difficult to translate the control requirements at this level into an appropriate conventional control structure. In the MPC methodology, this combination of blocks is replaced by a single MPC controller. The MPC controller frequently functions in a supervisory mode by specifying setpoints to lower level controllers.

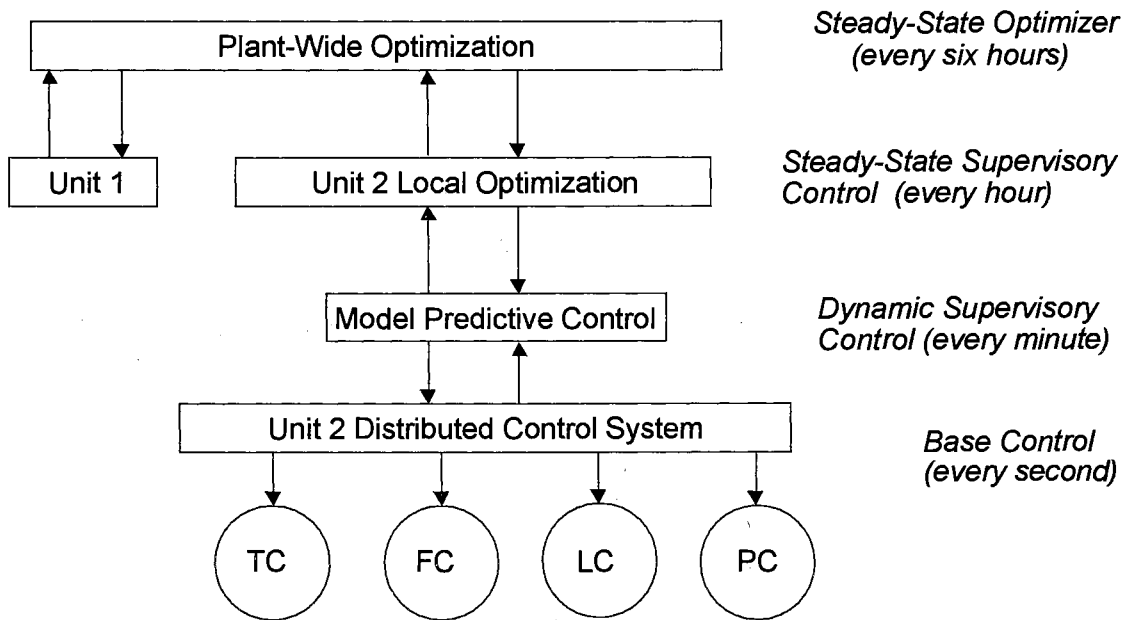


Figure 2.2: Role of model predictive control from process operation perspective (adapted from Qin and Badgwell (1997)).

2.2 MPC – Origin

Model predictive control has appeared in different branches of control literature over the past thirty-five years. Eaton and Rawlings (1992) suggest that the concept of using an open-loop optimal control computation to synthesize a feedback controller is so intuitive that it probably occurred to researchers prior to the availability of hardware and software technology to realize it. Garcia et al. (1989) cite Propoi (1963) as the first to introduce the idea of a finite moving horizon in 1963. A description of the essence of MPC is provided by Lee and Markus (1967) in their textbook on optimal control. They pointed out the difficulty in real-time implementation of the algorithm due to inadequate hardware and software (as of 1967).

In the electrical engineering literature, MPC is usually called receding horizon control. In 1970, Klienman (1970) used the finite horizon concept to find a state feedback gain that stabilizes a time invariant system. Thomas (1975) formulated a quadratic objective function penalizing only the input with the constraint that the state at the end of the horizon must be brought to zero. He showed the resulting state feedback law to be stabilizing for linear time-invariant systems. Later, Kwon and Pearson (1977) generalized the results to linear time-varying systems. As pointed by Eaton and Rawlings (1992), the MPC framework is also employed in aerospace engineering applications. They cite publications by Bruschi (1974) and Johnson (1975) where finite horizon problems are solved to obtain optimal aircraft trajectories.

2.3 MPC and standard Linear Quadratic Regulator

The development of modern MPC can be traced back to the works of Kalman (Kalman, 1960a; Kalman, 1960b) on the linear quadratic regulator. Consider a process described by a discrete-time, linear state-space model:

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k \\ \mathbf{y}_k &= \mathbf{C}\mathbf{x}_k\end{aligned}\tag{2.1}$$

The vector \mathbf{u} represents process inputs, vector \mathbf{y} describes process output measurements and \mathbf{x} represents the process states. Kalman sought to find the control sequence \mathbf{u}_k which minimizes the quadratic cost function for the infinite horizon regulator problem:

$$J = \sum_{j=1}^{\infty} \left(\|\mathbf{x}_{k+j}\|_{\mathbf{Q}}^2 + \|\mathbf{u}_{k+j}\|_{\mathbf{R}}^2 \right)\tag{2.2}$$

where, the weighted norms are defined as,

$$\|\mathbf{x}\|_{\mathbf{Q}}^2 = \mathbf{x}^T \mathbf{Q} \mathbf{x}\tag{2.3}$$

$$\|\mathbf{u}\|_{\mathbf{R}}^2 = \mathbf{u}^T \mathbf{R} \mathbf{u}\tag{2.4}$$

The weight matrices \mathbf{Q} and \mathbf{R} allow for scaling differences and tuning trade-offs. Variables \mathbf{x} and \mathbf{u} in the objective function in equation (2.2) represent deviations from the desired steady-state. The solution to the LQR problem was shown to be a constant gain controller:

$$\mathbf{u}_k = -\mathbf{K}\mathbf{x}_k\tag{2.5}$$

where, the gain matrix, \mathbf{K} , was computed from the solution of a discrete algebraic Riccati equation. The LQR solution was shown to be stabilizing for the process (equation (2.1)) with (\mathbf{A}, \mathbf{B}) stabilizable and (\mathbf{A}, \mathbf{C}) detectable with weight matrices \mathbf{Q} and \mathbf{R} as positive semi-definite and positive definite, respectively. A dual theory was developed to estimate process states from noisy input and output measurements, known as the Kalman

filter. The Kalman filter in conjunction with LQR yields the linear quadratic Gaussian (LQG) controller.

Qin and Badgwell (1997) point out that although LQG theory provided an elegant and powerful solution to control of unconstrained linear processes, "it had little impact on the control technology development in the process industry." They summarize some of the reasons for the failure as:

- 1) lack of the ability to address constraints
- 2) process nonlinearities
- 3) model uncertainty
- 4) unique performance criteria
- 5) cultural reasons (people, education, etc.)

The importance of constraints has been demonstrated by Prett and Gillette (1979). They show that the economic operating point often lies at the intersection of constraints. Thus, it is desirable to operate the closed loop system near constraints without violating them. Moreover, chemical processes are inherently nonlinear and their dynamics change with operating conditions. For all of these reasons, the LQG technique had minimal impact on the process industries.

2.4 Linear MPC – Developments in industry

Model predictive control philosophy as described in Section 2.1 does not prescribe the model type to be used in the control scheme. The initial industrial implementation of MPC used linear models to predict future process dynamics. This

class of control algorithms is referred to a linear model predictive control, i.e. MPC with linear models, and primarily differs in the choice of the control model and incorporation of constraints. The following is a brief description of the important control algorithms developed in the process industry.

(a) Model predictive heuristic control (MPHC)

The first description of MPC application was presented by Richalet et al. (1978). They referred to their algorithm as model predictive heuristic control. The solution software was named IDCOM, an acronym for identification and command.

Richalet et al. chose a discrete-time finite impulse response (FIR) model to describe the relationship between the process inputs, u , (manipulated and disturbance variables) and the process output, y , (controlled variable). For a single input, single output case, the FIR model takes the form,

$$y_{k+j} = \sum_{i=1}^N h_i u_{k+j-i} \quad (2.6)$$

The weights h_i , are called the impulse response coefficients. The sum is truncated after N sample periods when past inputs no longer influence the outputs. This representation is only possible for stable systems. The FIR model was identified from plant test data based on an algorithm that minimized the distance between process outputs and model response in the coefficient space. The reference trajectory was defined as a first order path from the current output value to the desired setpoint. The time constant of the reference trajectory controlled the speed of the desired closed loop response.

The control problem was interpreted as the dual of the identification problem. It consisted of estimating process inputs, which would minimize the distance between the predicted future output trajectory and the reference trajectory. Thus, the control and identification problems were solved using the same algorithm. Richalet et al. described applications of MPHC algorithm to a fluid catalytic cracking unit, a power generator, a polyvinyl chloride plant and a main fractionator unit. Mehra et al. (1982) provided further applications including a superheater, a steam generator, a wind tunnel, a utility boiler connected to a distillation column and a glass furnace.

(b) Dynamic Matrix Control (DMC)

Engineers at Shell Oil independently developed their own MPC technology in the 1970s. Cutler and Ramaker (1979) presented an unconstrained multivariable control algorithm which they called Dynamic Matrix Control (DMC). The DMC algorithm used a linear discrete-time, step response model to relate changes in process output to a weighted sum of past input changes. For a SISO process, the step response model takes the form

$$y_{k+j} = \sum_{i=1}^{N-1} s_i \Delta u_{k+j-i} + s_N u_{k+j-N} \quad (2.7)$$

The weights s_i , are called the step response coefficients. As in MPHC, the sum is truncated after N sample periods when past inputs no longer influence the outputs. The future outputs and future input moves were related to each other by a Dynamic Matrix. Using this representation, future input moves could be computed analytically as a solution of the least squares problem. In practice, the matrix inverse can be calculated offline. Cutler and Ramaker demonstrated the superiority of the DMC algorithm over

conventional PID lead/lag compensator by an application to furnace temperature control. Prett and Gillette (1979) provided further applications of DMC technology to fluid catalytic cracking unit reactor/ regenerator. They also described additional ad-hoc modifications to the unconstrained DMC algorithm to prevent violation of absolute input constraints. When a predicted future input came sufficiently close to a constraint, an extra equation was added to the process model. This would drive the input back to the feasible region. Cutler and Hawkins (1987) report a complex industrial DMC application to a hydrocarbon reactor involving seven input variables (five manipulated and two disturbance variables) and four output variables.

(c) Quadratic Dynamic Matrix Control (QDMC)

The original IDCOM and DMC algorithms provided adequate control of unconstrained multivariable processes. However, constraints were handled in an indirect manner. Engineers at Shell Oil addressed this weakness by posing the DMC algorithm as a quadratic program, in which input and output constraints appear explicitly. Garcia and Morshedi (1986) published a comprehensive description of this method and termed it as Quadratic Dynamic Matrix Control.

The QDMC algorithm strictly enforces input and output constraints at each point of the prediction horizon. Constraints enforced strictly are called hard constraints. In practice, Garcia and Morshedi reported that hard output constraints are typically required to be satisfied only over a portion of the horizon which they referred to as the constraint window. They also observed that if non-minimum phase dynamics are present,

performance is improved by pushing the constraint window further to the future. An alternate option was suggested for handling output constraints in presence of non-minimum phase dynamics. When output constraint violations are predicted to occur, the controller should attempt to minimize the violation in a least squares sense. This approach is known as the soft constraint concept.

Garcia and Morshedi (1986) presented results from a pyrolysis furnace application. The QDMC controller adjusted fuel gas pressure in three burners in order to control steam temperature at three locations in the furnace. They also reported good results in many Shell problems, one of them as large as 12 x 12. A number of applications using DMC/QDMC are available in open literature, for example, see (Kelly, et al., 1988; Van Hoof, et al., 1989; Bozin and Austin, 1995; Meziou, et al., 1996).

2.5 Linear MPC – Theoretical Aspects

Later refinements of industrial MPC technology came in terms of constraint handling and recovery from infeasibility. A comprehensive review of industrial MPC technology is provided by Qin and Badgwell (1997). Industrial implementation of MPC requires robust algorithms with acceptable performance that can be implemented online. Hence, a large number of heuristic approaches were adopted with little theoretical justification (Muske and Rawlings, 1993). Control researchers, therefore, have attempted to evaluate MPC algorithms from a theoretical perspective. In the following paragraphs a brief account of a few important results is given.

Nominal Stability: Garcia and Morari (1982) discussed the fundamental similarities between DMC and IDCOM for the SISO case and noted their relationship to other forms of optimal control. They developed a unifying control structure for such algorithms and termed it as Internal Model Control (IMC). A key result from their stability analysis concluded nominal stability of the feedback system if stability of the plant and controller is guaranteed. They also investigated the effect of controller tuning on stability. Based on their results, the authors provided tuning procedures for IMC that provide robust model predictive control for linear, time invariant processes. In a later publication, Garcia and Morari (1985) extended the IMC method to multivariable systems. Although they developed stability theorems for certain types of unconstrained problems, no provision for constrained optimization was included. Ricker (1985) presented constrained IMC solved by quadratic programming technique. No formal stability analysis was presented. However, closed loop stability is discussed via heuristic rules of feedback filtering and input blocking.

Zafriou (1990) noted that the presence of hard constraints in the online optimization problem produces a nonlinear controller even when the plant and model dynamics are assumed linear. He provided a contraction mapping framework to study the properties of the control algorithm. This framework accounts for the minimization of the objective function subject to certain hard constraints. Subsequently, he provided tuning guidelines so that constrained nominal stability is achieved. Zafriou and Marchal (1991) showed that inclusion of hard output constraints in the online optimization problem solved by QDMC may result in very aggressive control action. Based on contraction

mapping, they presented a necessary condition, which was shown to be a good indicator of stability.

Rawlings and Muske (1993) presented a constrained receding horizon controller that is stabilizing for both stable and unstable plants and for all choices of tuning parameters. A state-space formulation is used to account for both stable and unstable plant representations. Output feedback is performed using linear quadratic filtering theory. The salient feature of their regulator is use of an infinite horizon open-loop quadratic objective,

$$J = \sum_{j=1}^{\infty} y_{k+j}^T Q y_{k+j} + u_{k+j}^T R u_{k+j} + \Delta u_{k+j}^T S \Delta u_{k+j} \quad (2.8)$$

subject to linear constraints on the process inputs and outputs. Only a finite number of decision variables, N , (the control horizon) are retained by the assumption

$$u_{N+i} = 0, \quad i = 1, 2, \dots \quad (2.9)$$

Finally, they presented separate rigorous proofs for nominal, constrained, closed-loop stability of open-loop stable and unstable plants. An important feature of this method is identifying bounds for the output constraint window so that the resulting quadratic problem is feasible. For open-loop unstable processes, an equality constraint is appended that requires the unstable modes be brought to zero at the end of the control horizon. A number of examples presented by Muske and Rawlings (1993) demonstrate the features of this method.

Robust Stability: Limited work on closed-loop stability in presence of modeling errors exists in the literature. A survey in 1995 by Qin and Badgwell (1997) found that robust stability is a serious concern in the industry and is addressed by extensive closed loop simulation. Vunthamdam et al. (1995) reformulated the QDMC algorithm with an end condition, which they called EQDMC, for the multivariable problem. They presented a sufficient robust stability condition for SISO systems with hard input and soft output constraints. The robust stability condition dictates values of the move-suppression factors of the online objective function that increase as the modeling uncertainty increases. They parameterized model uncertainty in the time domain through maximum and minimum impulse response coefficients and developed a constraint involving the move suppression factor that, if satisfied, guarantees robust stability.

Lee and Cooley (1995) describe the well known min-max approach for solving the robust stability problem. The maximum cost function for all possible plants in the uncertainty description is minimized. Badgwell (1997) suggests that this method may result in a conservative solution since the worst-case scenario is optimized. He describes a robust MPC algorithm for open-loop stable, linear plant subject to hard input and soft output constraints. Model uncertainty is parameterized by a list of possible plants. Robust stability is achieved by adding constraints that prevent the sequence of optimal control costs from increasing for the true plant. The algorithm is a direct generalization of the nominal stabilizing regulator of Rawlings and Muske (1993). However, the resulting optimization problem becomes a convex nonlinear program.

Sznaier and Damborg (1990) discuss stability of a restricted class of constrained linear systems. In their problem statement, the states and inputs are constrained to lie in a bounded convex polyhedron. They show that under certain conditions, the resulting closed-loop system is asymptotically stable in the region of interest. Scokaert and Rawlings (1998) extended the work by Sznaier and Damborg by eliminating restrictions on the boundedness of constraint region.

Disturbance handling via state estimation: Most industrial MPC algorithms use a heuristic of lumping uncertainty in model parameters and structure, measurement errors, unmeasured disturbances and other sources of plant-model mismatch into a constant bias term, viewed as a step disturbance acting on the output. Lundstrom et al. (1995) show by an example that such an assumption leads to poor response in presence of ramp-like disturbances. To obviate the need for the assumption of step disturbance at the output, observer based MPC algorithms have been proposed in literature (Ricker, 1990; Lee, et al., 1994). In the above approaches, the observer (e.g. Kalman filter) is constructed using the nominal model of the plant and no attempt is made to update the model parameters.

2.6 Adaptive Model Predictive Control

There have been a large number of applications of adaptive feedback control over the past 30 years (Astrom and Wittenmark, 1995). Adaptive methods are readily available for automatic tuning of PID controllers. Traditionally, manual tuning of complex controllers has taken the route of modeling or identification followed by controller design. This is often a time-consuming and costly procedure. In adaptive

applications, on the other hand, the adaptation loop is simply switched on. The adaptive controller uses the current input/output data to identify a process model and/or controller parameters. The adaptive loop is run until the performance is satisfactory; then it is disconnected, and the system is left running with fixed controller parameters.

Despite the widespread use and appeal of adaptation in PID controllers, adaptive MPC has received relatively little attention. Clarke et al. (1987) proposed the generalized predictive control (GPC) algorithm in 1987. It uses a controlled auto regressive integrating moving average model to describe the process. GPC is similar to MPC in that the process is controlled using long-range model predictions. Variants of GPC have been presented by many authors, for example, by Lelic and Zarrop (1987), De Keyser et al. (1988), Clarke and Mohtadi (1989) and Clarke (1988), which depend on assumed model structures and choice of cost function. GPC has also been extended to multiple-input multiple-output systems (Kinnaert, 1989; Dion, et al., 1991). The early literature on GPC did not include a stability constraint. Later Clarke et al. (1991) modified the GPC algorithm by appending constraints for model stabilization. However, as noted by Lee and Cooley (1996) and Garcia et. al. (1989), these results rely on linear adaptive control theory precluding consideration of the industrially important issue of control and state constraints.

The use of continuous autoregressive integrated moving average models makes GPC suitable for parameter estimation by recursive least squares. DMC/QDMC algorithms, on the other hand, use a truncated step response model and no attempt is

made to evaluate model parameters online. The presence of a large number of step response coefficients in the model makes recursive identification at each sampling period an impractical task. To overcome this difficulty, Maiti and Saraf (1995) proposed calculation of only a few step response coefficients from the process input-output data. They fit a first order plus time delay (FOPTD) model by minimizing the sum of squared deviations between the calculated step response coefficients and those predicted by the FOPTD model. The FOPTD model was then used to extrapolate the remaining coefficients needed to fill the dynamic matrix. The implementation of their adaptive DMC controller on a single-input, single-output (SISO) distillation column yielded superior results when compared with non-adaptive DMC.

Maiti and Saraf predicted the step response coefficients, a_i , using the FOPTD model as follows:

$$a_i = K_p \left(1 - \exp \left(\frac{T_d - t_i}{\tau} \right) \right) \quad (2.10)$$

It is apparent that equation (2.10) is nonlinear with respect to the model parameters, viz. K_p , the process gain; τ , the process time constant; and T_d , the process delay. Consequently these cannot be estimated using standard recursive least squares.

In Chapter 3, an adaptive strategy to identify the process by a low order parametric model is presented. Unlike the approach by Maiti and Saraf, we base the identification criterion on minimization of the sum of squared deviations between the predicted model output and the process measurements. Thus, while the DMC/QDMC controller uses the step response model, we utilize an autoregressive model with external

inputs (ARX) for identification purposes. The step response coefficients needed by the DMC/QDMC algorithm are obtained from the ARX model in a straightforward computation. The use of an ARX model enables use of standard recursive least squares for process identification.

In all the above adaptive approaches, models are estimated online and control action calculated based on the assumption that the estimated model gives an exact representation of system dynamics. This is often referred to as the certainty equivalence principle (Astrom and Wittenmark, 1995). Ydstie (1996) identifies two issues that must be addressed in certainty equivalence control: (1) the estimated model must be well-behaved in the sense that the controller stabilizes the model. This is often referred to as the admissibility problem; (2) if the parameter estimator ignores model uncertainty and unmodeled dynamics, the parameter values may grow unbounded. This is referred to as the parameter drift problem. Ydstie also discusses the gap between theory and practice in adaptive control and reports on the status of work to bridge the gap.

Despite the strong market incentive for a self-tuning MPC controller, only one industrial application has been reported (Dollar, 1993). Qin and Badgwell (1997) suggest that, "barring a theoretical breakthrough, the situation is not likely to change in the near future." However, limited adaptation for deviations from base case models is foreseen as a practical solution (Froisy, 1994).

2.7 Nonlinear Model Predictive Control

The current generation of commercially available MPC technology is based on linear dynamic models, and therefore is referenced by the generic term "linear model predictive control." Although often unjustified, the assumption of process linearity greatly simplifies model development and controller design. However, many processes are sufficiently nonlinear to preclude the successful application of LMPC technology. Such processes include highly nonlinear processes that operate near a fixed operating point (e.g., high purity distillation columns) and moderately nonlinear processes with large operating regimes (e.g., multi-grade polymer reactors) (Henson, 1998).

Henson notes that, "while NMPC offers the potential for improved process operation, it offers theoretical and practical problems which are considerably more challenging than those associated with LMPC." The prime difficulties arise from nonlinear process modeling and the subsequent computational issues associated with online solution of nonlinear programs. Bequette (1991) notes that all NMPC algorithms are formulated using nonlinear programming techniques. Further, Mayne (1996) argues that model constraints corresponding to satisfaction of model equations over the prediction horizon, generally, result in a nonconvex optimization. Various solution methods of solving the online finite horizon nonlinear control problem are available (Santos, et al., 1995; Mayne, 1995). Staus et. al (1996) study a class of nonlinear problems for which the global optimum can be computed online. A similar study is reported by Srinivas and Arkun (1995).

A common approach to the nonlinear MPC problem has been to use successive linearization of nonlinear models. Garcia (1984) proposed a nonlinear QDMC algorithm, a simple extension of DMC/QDMC based on online successive linearization of a mechanistic nonlinear model. Nonlinear MPC using closed-loop state estimation by an extended Kalman filter has been proposed by Lee and Ricker (1994). Gattu and Zafiriou (1995) augmented the system states with stochastic states to account for modeling errors and disturbances. Banerjee et al. (1997) describe a method of state estimation for nonlinear systems that are subject to multiple operation regimes and make transitions between them. The nonlinear process is approximated by a linear parameter varying system which consists of local linear models. Krishnan and Kosanovich (1998) also present a multiple model based MPC scheme. The linear time invariant models are computed offline along a pre-defined reference trajectory of a batch process. Each of the above nonlinear MPC techniques use the standard quadratic programming optimization method to obtain control inputs. Also, they assume availability of accurate nonlinear model (or multiple linear models).

The industrial success of LMPC is largely due to the availability of commercial software packages, which can be used to develop linear dynamic models directly from the process data. These linear empirical models are used by LMPC controller to predict and optimize process performance. On the other hand, the complexity of nonlinear systems precludes straightforward extension of linear theory to nonlinear system identification techniques (Pearson and Ogunnaike, 1997). Cook (1986) indicates that because of the large number of different types of nonlinearities can occur in practice, extending a basic

control scheme to account for all possibilities is unrealistic. One way of tackling the general nonlinear problem is to employ a framework, within which a large number of nonlinear processes can be adequately approximated. Volterra and Hammerstein models (Agarwal and Seborg, 1987) and neural networks (Hussain, 1999) have been studied as nonlinear modeling tools. As an alternative, the NMPC controller may be based on a fundamental model which is derived from conservation laws and constitutive equations. These two classes of nonlinear models are discussed below:

Fundamental Models: Fundamental dynamic models are derived by application of transient mass, energy and momentum balances in conjunction with constitutive equations. The continuous time differential equations are discretized by some method (e.g., orthogonal collocation on finite elements (Meadows and Rawlings, 1997)) to allow incorporation in the NMPC scheme.

Fundamental models enjoy certain advantages over nonlinear empirical models. As long as the underlying assumptions remain valid, fundamental models can be expected to extrapolate to operating regions which are not represented in the data set. Further, model parameters can be estimated from laboratory experiments and routine operating data instead of time-consuming plant tests.

Henson (1998) notes that most of the NMPC studies based on fundamental models reported in the open literature consist of a single unit operation and a relatively simple nonlinear dynamic model. He suggests that this is "attributable to the inherent

difficulties involved in deriving fundamental dynamic models for large scale processes." One solution is to apply model reduction techniques which result in a simplified model with similar input/output behavior as of the rigorous fundamental model. Use of such an approach has been applied to chemical reactors (Duchene and Rouchon, 1996) and distillation columns (Levine and Rouchon, 1991). Alternatively, one may derive simplified fundamental models that partially describe process characteristics. Applications of the simplified fundamental model approach have been reported by Benallou et al. (1986) and Hwang (1991).

Empirical Models: In many applications, lack of process knowledge precludes the formulation of a fundamental model. This necessitates the development of empirical models from dynamic plant data. However, unlike the well-developed theory of linear system identification, nonlinear system identification is a less well understood area. A prime difficulty associated with nonlinear empirical modeling is selection of a suitable model structure. Pearson and Ogunnaike (1997) summarize the following types of discrete-time nonlinear models utilized for NMPC: (1) Hammerstein and Weiner models, which consist of a serial combination of a static nonlinearity with a linear dynamic model, (2) Volterra models, which are expansions of nonlinear functions, (3) autoregressive moving average models with external inputs (ARMAX), and (5) artificial neural network models. Henson (1998) describes nonlinear system identification as a five step procedure:

- (1) model structure selection,
- (2) test input sequence design,

- (3) noise model,
- (4) estimation of model parameters and
- (5) model validation.

Empirical models offer several advantages over fundamental models. Detailed process knowledge is not necessary for empirical model development. This consideration is important for complex processes. Secondly, complexity of empirical models can be restricted thereby reducing computations during the online nonlinear optimization. NMPC based on empirical models such as Hammerstein and Wiener models (Chu and Seborg, 1994), Volterra models (Maner, et al., 1996), ARMAX models (Srinivas and Arkun, 1995) and neural network models (Su and McAvoy, 1997) has been reported by several investigators. Among these, artificial neural networks are the most popular framework for empirical model development.

2.8 Neural Network Models In Process Control

In the past few years, renewed interest has been paid to neural network based models because of their simple structure and fast and effective computational performance (Su and McAvoy, 1997). Their flexibility makes them suitable for a wide class of applications such as system identification and control, inferential modeling, and fault diagnosis. The most attractive property of neural networks (NN) is their ability to represent any arbitrary nonlinear functional mapping between input and output data (Hornik, et al., 1989). This is achieved through a training process that takes place by repeatedly presenting the input data and the corresponding target output to the network.

After a sufficient number of training iterations, the network creates an internal approximate process model by learning to recognize the map relating the outputs to the corresponding inputs. It is important to note that this internal model is not based on any specification of the actual process mechanism; the NN itself generates this approximate model.

In reality, the control engineer has to have a reasonable amount of process knowledge (MacGregor, et al., 1991). Indeed, the critical point in developing a robust NN model is selecting the most representative process inputs, and this can only be achieved through an understanding of the underlying process physics. The ability of neural networks to handle complex nonlinear processes opens a wide range of opportunities in advanced nonlinear process control. Among the large number of feedforward NN algorithms, multilayer perceptron (MLP) (Rumelhardt, et al., 1986) and radial basis function (RBF) networks (Moody and Darken, 1989) have been widely used as nonlinear models for MPC, inferential measurements and process monitoring. In the next two sub-sections, literature on application of NNs in inferential modeling and nonlinear MPC is reviewed.

(a) **Neural Networks in Inferential Modeling:** A number of applications of neural networks as inferential models (Kramer, 1992; Yang, et al., 1995) have been reported. Industrial use of NN based inferential models has also been reported (Samdani, 1990; Piovosio and Owens, 1991; Schnelle and Fletcher, 1990). Kresta et al. (1996) presented model development using partial least squares (PLS). The efficacy of the method was demonstrated by inference of the heavy key composition in the distillate.

The independent latent variables were constructed using various temperature and flowrate measurements. Qin et al. (1997) constructed soft sensors using a principle component analysis (PCA) approach for continuous monitoring of emissions. However, in most instances, the network inputs are assumed to be known *a priori* and the model developed by training the neural network using exemplars. As noted by McGregor et al. (1991), a poor choice of model inputs can result in a poor NN model.

In Chapter 4, we discuss a unified framework to construct neural network based inferential models. The methodology includes selection of model variables from a large candidate set.

(b) Neural Networks in NMPC: Applications using recurrent NNs have also been reported by Karjala and Himmelblau, (1994). Psychogis and Ungar (1991) used an MLP model of a continuous stirred tank reactor to control product concentration using the MPC scheme. Using feedback to account for modeling errors, they obtained offset-free tracking of setpoints. Willis et al. (1991) used an MLP to control product concentration in a CSTR. Turner et al. (1995) used NNs for distillation column control. Gokhale et al. (1995) used a steady-state MLP model to replace the tray-to-tray model used in a predictive model based controller to control the product compositions in a propylene-propane splitter. Emmanouilides and Petrou (1997) utilized an MLP model in a model predictive scheme to control the substrate concentration and pH of a complex nonlinear anaerobic digestion system. The model was estimated online and provided setpoint tracking in presence of process characteristic changes.

Hernandez and Arkun (1990) applied MLP networks to estimate the disturbance due to nonlinearities in conjunction with dynamic matrix control. This estimate was added to the linear model prediction during feedback. Case studies demonstrated the superiority of this algorithm relative to conventional linear DMC.

In all of the above neural network based NMPC approaches, the neural network model is used to predict the future process behavior and this information is used by the online optimizer to generate the next control input. Thus, the model and the optimizer are separate entities of the controller. In Chapter 5, the optimization problem of the controller is directly parameterized in terms of an RBF network model. This novel approach exploits factorability of Gaussian nodes to separate the decision variables of the nonlinear program from all known quantities. Such a strategy allows analytical expressions for the gradient and Hessian of the objective function. Consequently, the computational efficiency of the controller is enhanced by reducing the computational burden during each iterative step of the nonlinear program and also the number of function calls during optimization.

Most of the neural network based NMPC applications reported in the literature use small processes to demonstrate effectiveness of control. However, to be useful, process control algorithms must successfully operate in the modern process industry environment. To this end several challenging problems were published in *Computers & Chemical Engineering*, Vol. 17, 1993, to enable the control community to test their

control algorithms on industrially significant problems. The Eastman problem (EP) entitled, "A Plantwide Industrial Process Control Problem" by Downs and Vogel (1993) is one such case. In Chapter 6, we test the factorized RBF based NMPC developed in Chapter 5 for control of the Eastman challenge process.

3 ADAPTIVE QUADRATIC DYNAMIC MATRIX CONTROL

Chapter Overview

This chapter presents an adaptive application of quadratic dynamic matrix control (QDMC) using recursive least squares method. Model adaptation is useful when the existing model is inaccurate and an accurate model is desired. In the proposed work, the process is identified under closed-loop conditions using an autoregressive model with external inputs. A detailed description is provided for the case where the process can be approximated as a first-order-plus-time-delay model. The issue of unknown time-delay is addressed by making use of a (1,1) Pade approximation. Parameterization of such models is discussed for single input single output and multiple input multiple output systems. Simulation studies using generic first-order-plus-time delay processes and a nonlinear CSTR demonstrate performance of the adaptive QDMC scheme.

3.1 Introduction

Model predictive control (MPC) has been widely accepted in the process industries (Qin and Badgwell, 1997) for over two decades. During this period, a large number of MPC algorithms have been proposed in the literature. Generalized predictive control (Clarke, et al., 1987) (GPC), model algorithmic control (Richalet, et al., 1978), dynamic matrix control (Cutler and Ramaker, 1979) (DMC), and quadratic dynamic matrix control (Garcia and Morshedi, 1986) (QDMC) are some of the most popular approaches. A number of successful implementations of the DMC/QDMC (Cutler and Hawkins, 1987;

Kelly, et al., 1988; Van Hoof, et al., 1989; Bozin and Austin, 1995; Meziou, et al., 1996) and GPC (Clarke, 1988; Dion, et al., 1991) algorithms have been reported in the literature. All MPC algorithms share a common underlying philosophy, that is, use of an explicit model to predict the process behavior over a future horizon, and implementation of control action that steers the process towards predetermined objectives in an optimal sense. The dependence of MPC techniques on the "goodness" of a model makes it an ideal candidate for adaptation.

Many chemical processes are inherently nonlinear in their input-output relationships. The models used in linear MPC describe a process well only in the vicinity of some fixed operating point. As process conditions deviate from the nominal operating point, model mismatch increases with a corresponding degradation in control performance. The problem is particularly severe in the process industries where the areas of a plant with the greatest economic incentives to apply MPC typically exhibit nonlinearity (e.g., a reactor system) and are time varying in nature (e.g., equipment fouling, catalyst deactivation, etc.). Plant operators frequently disable an MPC system when model mismatch compromises overall control performance. Recommissioning cannot be performed until the MPC models are updated or the operating conditions return to original design point. With the increasing emphasis on agile or flexible manufacturing, the latter may no longer represent a viable option. To address this issue, a number of enhancements of the MPC technique have been proposed. Some of these are briefly reviewed below.

Most industrial MPC algorithms use a heuristic of combining errors in model parameters and structure, measurement errors, unmeasured disturbances and other sources of plant-model mismatch into a constant bias term, viewed as a step disturbance acting on the output. Lundstrom et al. (1995) showed by an example that such an assumption leads to poor response in presence of ramp-like disturbances. To obviate the need for the assumption of step disturbance at the output, observer based MPC algorithms have been proposed in literature (Ricker, 1990; Lee, et al., 1994). In the above approaches, the observer (e.g. Kalman filter) is constructed using the nominal model of the plant and no attempt is made to update the model parameters.

Control of nonlinear processes has been addressed by direct development of nonlinear MPC capability. Garcia (1984) extended DMC/QDMC to nonlinear MPC by performing online successive linearization of a mechanistic nonlinear model. Nonlinear MPC using closed-loop state estimation by an extended Kalman filter has been proposed by Lee and Ricker (1994). Gattu and Zafiriou (1995) augmented the system with stochastic states to account for modeling errors and disturbances.

Control of processes over a wider operating region by using multiple models has been described by Banerjee et al. (1997). Their method uses state estimation for nonlinear systems that are subject to multiple operation regimes. The nonlinear process is approximated by a linear parameter varying system consisting of local linear models. Krishnan and Kosanovich (1998) also presented a multiple model based MPC scheme. The linear time invariant models are computed offline along a pre-defined reference

trajectory for a batch process. Each of the above nonlinear MPC techniques use the standard quadratic programming optimization method to obtain control inputs. Also, they assume availability of an accurate nonlinear model (or multiple linear models). No attempt is made to address uncertainty in the model parameters.

All of the previously referenced approaches treat errors in model parameters along with other sources of process-model mismatch as a disturbance on the output and no attempt is made to estimate the model parameters directly. Model identification is considered as a separate activity from control. In this paper, we present a strategy that combines model parameter estimation and linear QDMC via standard methods in indirect adaptive control, while retaining the constant bias heuristic. Model uncertainty is addressed by estimating DMC/QDMC model parameters online. Such an approach may be useful in situations, where some or all model parameters are uncertain and a better estimate is desired.

While the DMC/QDMC controller uses a convolution step response model, we utilize an autoregressive model with external inputs (ARX) for identification purposes. The step response coefficients needed by the DMC/QDMC algorithm are obtained from the ARX model by a simple computation. ARX models enable use of standard recursive least squares (RLS) for process identification. The proposed technique can be used to address model uncertainty in existing industrial MPC implementations in a straightforward way. The method may also be used in conjunction with state estimation techniques with linear models, but has not been pursued in the current work.

Use of parameter adaptation within the MPC framework is not novel. GPC was specifically developed for self-tuning/adaptive control applications and thus offers the potential for online model updates. The use of continuous autoregressive integrated moving average models makes GPC suitable for parameter estimation by recursive least squares. The industrially significant DMC/QDMC algorithms, on the other hand, use a truncated step response model and no attempt is made to evaluate model parameters online. The presence of a large number of step response coefficients in the DMC/QDMC model makes recursive identification at each sampling period an impractical task. To overcome this difficulty, Maiti and Saraf (1995) proposed calculation of only a few step response coefficients from the process input-output data. They fit a first order plus time delay (FOPTD) model by minimizing the sum of squared deviations between the calculated step response coefficients and those predicted by the FOPTD model. The FOPTD model was then used to extrapolate the remaining coefficients needed to fill the dynamic matrix. The implementation of their adaptive DMC controller on a single-input, single-output (SISO) distillation column yielded superior results when compared with non-adaptive DMC.

The structure of an adaptive DMC/QDMC system is shown in Figure 3.1. The adaptive system has two distinct loops (Astrom and Wittenmark, 1995):

- 1) a standard feedback loop containing the process and the DMC/QDMC controller,
- 2) a model parameter estimation loop using the recursive estimator.

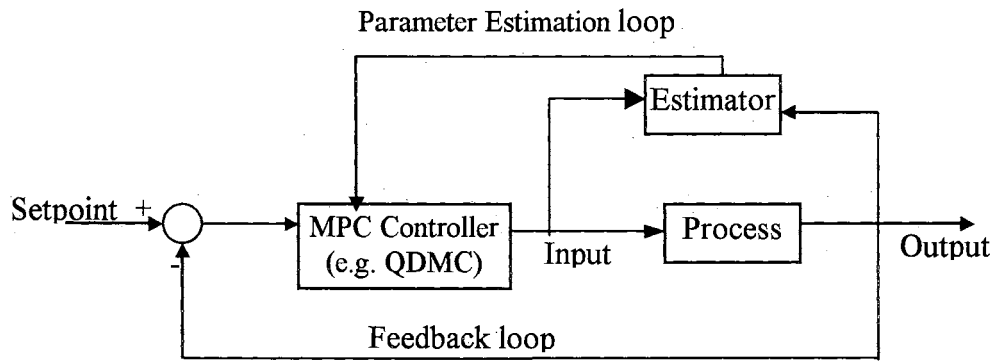


Figure 3.1: Block diagram of adaptive model predictive control system.

In sections 3.2 and 3.3, key issues of indirect adaptive control and closed-loop identification are reviewed. These include parameterization for SISO and MIMO models and use of *a priori* information. In section 3.4, we use these ideas to synthesize an adaptive QDMC algorithm. A simplified analysis is presented to show that a SISO process controlled by DMC is closed-loop identifiable. Finally, section 3.5 presents simulation examples demonstrating the benefits of adaptive QDMC/DMC relative to the non-adaptive QDMC/DMC for FOPTD models and a nonlinear CSTR.

3.2 Recursive Identification and Model Parameterization

Ideally, process models used for control should be updated under closed-loop conditions. Furthermore, the model update procedure should be completed in less than one control interval so that it does not lag behind the input/output information flow. Recursive schemes are desirable in such situations for computational efficiency. In recursive schemes, the results of previous calculations are used to obtain a current estimate of the desired parameters. The recursive least squares (RLS) method is one such algorithm. Implementation of the RLS algorithm simplifies significantly when the model has the property of being linear in the parameters. A detailed treatment of least square estimators can be found in standard texts on estimation theory (Ljung, 1987; Mendel, 1995).

One of the key elements of recursive process identification is the selection of the model structure. If sufficient open-loop data exist, it is possible to determine the order of a linear process using statistical methods (Box and Jenkins, 1994). However, many open-loop stable chemical processes are well-described by low order models with time delay (Astrom and Wittenmark, 1995; Ogunnaike and Ray, 1994). The use of a first order plus time delay (FOPTD) model has been reported in a number of applications (Astrom and Wittenmark, 1995; Clarke, 1988; Ogunnaike and Ray, 1994; Wood and Berry, 1973). The remainder of this section discusses parameterization of FOPTD models.

3.2.1 Single Input Single Output Systems (SISO)

Let the process be described by the following model:

$$y(i) + a_1 y(i-1) + \dots + a_n y(i-n) = b_1 u(i-d-1) + \dots + b_m u(i-d-m) \quad (3.1)$$

where $y(i)$ and $u(i)$ are the process output and input, respectively, at the i^{th} instant, d is the process delay, and a_j and b_j represent the parameters of the model. In vector notation,

$$y(i) = \phi^T \theta \quad (3.2)$$

with the regressor vector,

$$\phi = [-y(i-1) \quad \dots \quad -y(i-n) \quad u(i-d-1) \quad \dots \quad u(i-d-m)]^T \quad (3.3)$$

and the parameter vector,

$$\theta = [a_1 \quad \dots \quad a_n \quad b_1 \quad \dots \quad b_m]^T \quad (3.4)$$

For a SISO process, the FOPTD model can be represented in the z-domain by the following transfer function:

$$h(z, \theta) = \frac{b_1 z^{-1}}{1 + a_1 z^{-1}} z^{-d} \quad (3.5)$$

where a_1 , b_1 , and d are the model parameters and θ represents the model parameter vector. It is noted that the above equation does not represent a single model, but a set of models (Ljung, 1987). The task of the identification algorithm is to determine the optimum value of the model parameter vector. The model represented by equation (3.5) is rewritten in difference form as:

$$y(k) + a_1 y(k-1) = b_1 u(k-d-1) \quad (3.6)$$

If the delay, d , associated with the process is known, then equation (3.6) is linear in the unknown parameters a_1 and b_1 . Often, the delay is determined by physical transport lag and varies with the magnitude of the manipulated variable (e.g., hydraulic delay in a pipe varies with the flowrate). Consequently, it may also be necessary to estimate d in the FOPTD model. In this case, equation (3.6) becomes nonlinear and the standard RLS method cannot be used for parameter estimation. To overcome this problem, the delay term can be replaced by a pole-zero pair using the first order Pade approximation. Using this approach, the reparameterized model in z -domain becomes,

$$h(z, \theta) = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \quad (3.7)$$

where $[a_1 \ a_2 \ b_1 \ b_2]^T$ is the vector of unknown parameters. The input-output model described by equation (3.7) may be written in difference form without a delay term as:

$$y(k) + a_1 y(k-1) + a_2 y(k-2) = b_1 u(k-1) + b_2 u(k-2) \quad (3.8)$$

or

$$y(k) = \phi^T \theta \quad (3.9)$$

where the regressor vector,

$$\phi = [-y(k-1) -y(k-2) u(k-1) u(k-2)]^T \quad (3.10)$$

and the parameter vector

$$\theta = [a_1 \ a_2 \ b_1 \ b_2]^T \quad (3.11)$$

More parameters are required to describe an FOPTD process using equation (3.8) compared to equation (3.6). However, equation (3.8) provides the ability to model a process with variable time delay and allows use of recursive least squares for closed-loop system identification.

3.2.2 Multi-Input Multi-Output Systems (MIMO)

A similar approach can be used to parameterize MIMO systems. Consider a 2x2 system:

$$\begin{bmatrix} y_1(z) \\ y_2(z) \end{bmatrix} = \begin{bmatrix} h^{11}(z) & h^{12}(z) \\ h^{21}(z) & h^{22}(z) \end{bmatrix} \begin{bmatrix} u_1(z) \\ u_2(z) \end{bmatrix} \quad (3.12)$$

or in vector notation,

$$\mathbf{y}(z) = \mathbf{H}(z, \theta) \mathbf{u}(z) \quad (3.13)$$

The ij^{th} element of $\mathbf{H}(z, \theta)$ describes the relationship between the j^{th} input and the i^{th} output of the process. If the time delay in the input-output relations is not known or variable, the transfer function matrix $\mathbf{H}(z, \theta)$ is constructed from the following elements,

$$h^{ij}(z) = \frac{b_1^{ij} z^{-1} + b_2^{ij} z^{-2}}{1 + a_1^{ij} z^{-1} + a_2^{ij} z^{-2}} \quad (3.14)$$

The superscript ij indicates the location of the parameters in the matrix $\mathbf{H}(z, \boldsymbol{\theta})$. The transfer function matrix can be expressed as the product of a numerator and denominator polynomial matrix using a left matrix fraction description:

$$\mathbf{H}(z, \boldsymbol{\theta}) = \mathbf{P}^{-1}(z, \boldsymbol{\theta})\mathbf{N}(z, \boldsymbol{\theta}) \quad (3.15)$$

The factors, $\mathbf{P}^{-1}(z, \boldsymbol{\theta})$ and $\mathbf{N}(z, \boldsymbol{\theta})$, are not unique. In one of the representations of the model described by equations (3.12) and (3.14), the denominator polynomial matrix, $\mathbf{P}(z, \boldsymbol{\theta})$, is

$$\begin{bmatrix} (1 + a_1^{11}z^{-1} + a_2^{11}z^{-2})(1 + a_1^{12}z^{-1} + a_2^{12}z^{-2}) & 0 \\ 0 & (1 + a_1^{21}z^{-1} + a_2^{21}z^{-2})(1 + a_1^{22}z^{-1} + a_2^{22}z^{-2}) \end{bmatrix} \quad (3.16)$$

and the numerator polynomial matrix, $\mathbf{N}(z, \boldsymbol{\theta})$, is represented by

$$\begin{bmatrix} (b_1^{11}z^{-1} + b_2^{11}z^{-2})(1 + a_1^{12}z^{-1} + a_2^{12}z^{-2}) & (b_1^{12}z^{-1} + b_2^{12}z^{-2})(1 + a_1^{11}z^{-1} + a_2^{11}z^{-2}) \\ (b_1^{21}z^{-1} + b_2^{21}z^{-2})(1 + a_1^{22}z^{-1} + a_2^{22}z^{-2}) & (b_1^{22}z^{-1} + b_2^{22}z^{-2})(1 + a_1^{21}z^{-1} + a_2^{21}z^{-2}) \end{bmatrix} \quad (3.17)$$

It is desirable to reduce the degree of the determinant of the denominator polynomial matrix to ensure that a lesser number of parameters are required to describe the system. The model described by equation (3.15) will be maximally reduced if the matrices $\mathbf{P}(z, \boldsymbol{\theta})$ and $\mathbf{N}(z, \boldsymbol{\theta})$ are left coprime (Brogan, 1991). However, in the absence of numerical values of the parameter vector, $\boldsymbol{\theta}$, no further identification of common factors is possible. Thus, the structure of the polynomial matrices in equations (3.16) and (3.17) is used to formulate the parametric model for recursive identification. Let α_k^{ij} represent

the coefficient associated with the k^{th} power of z^{-1} in the ij^{th} monic polynomial of the denominator matrix, $\mathbf{P}(z, \boldsymbol{\theta})$ and β_i^{lj} the coefficients associated with the l^{th} power of z^{-1} in the ij^{th} polynomial of the numerator matrix $\mathbf{N}(z, \boldsymbol{\theta})$. The model represented by equations (3.12) and (3.14) is rewritten in difference form as follows:

$$y_1(k) = -\sum_{i=1}^4 \alpha_i^{11} y_1(k-i) + \sum_{j=1}^2 \sum_{i=1}^4 \beta_i^{1j} u_j(k-i) \quad (3.18)$$

$$y_2(k) = -\sum_{i=1}^4 \alpha_i^{22} y_2(k-i) + \sum_{j=1}^2 \sum_{i=1}^4 \beta_i^{2j} u_j(k-i) \quad (3.19)$$

Since the equations are linear combinations of the past measurements and inputs, they can be rewritten in the form of equation (3.9). The regressor vector, $\boldsymbol{\phi}$, contains the past input-output data and the parameter vector, $\boldsymbol{\theta}$, consists of the unknown model parameters.

3.3 *A Priori* Information And System Identification

System identification is computationally simplified by incorporating all available *a priori* knowledge of the process. For instance, it may be known that in a 2x2 process, the output y_1 and input u_2 are uncoupled. In this case, the transfer function element $h^{12}(z)$ will be identically zero. The (1,1) element of the denominator polynomial matrix, $\mathbf{P}(z, \boldsymbol{\theta})$ would simplify to $(1 + \alpha_1^{11} z^{-1} + \alpha_2^{11} z^{-2})$ while the (1,1) and (1,2) elements of $\mathbf{N}(z, \boldsymbol{\theta})$ would become $(b_1^{11} z^{-1} + b_2^{11} z^{-2})$ and 0 respectively. Compared to the generic model described by equations (3.16) and (3.17), the use of *a priori* information obviously makes a significant reduction in the number of model parameters that must be estimated.

As an additional demonstration of use of *a priori* information, consider the two input, one output system,

$$y(z) = \frac{0.2z^{-4}}{1-0.8z^{-1}}u_1(z) + h_{12}(z, \alpha_1, \alpha_2, b_1, b_2)u_2(z) \quad (3.20)$$

where the relationship between y and u_1 is known but h_{12} , is unknown. We assume h_{12} is of the form of equation (3.14). For identification purpose, the model may be reparameterized as:

$$y(k) - 0.8y(k-1) - 0.2u_1(k-4) = \begin{bmatrix} -y(k-1) + 0.8y(k-2) + 0.2u_1(k-5) \\ -y(k-2) + 0.8y(k-3) + 0.2u_1(k-6) \\ u_2(k-1) - 0.8u_2(k-2) \\ u_2(k-2) - 0.8u_2(k-3) \end{bmatrix}^T \begin{bmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \end{bmatrix} \quad (3.21)$$

Thus, only the unknown part of the process is estimated. The known information about the relationship between y and u_1 , and the structure of h_{12} constitutes the *a priori* knowledge about the process and was employed to reduce the number of parameters in the identification model.

Parameter reduction steps such as this are crucial since closed-loop identification is sensitive to the number of parameters to be estimated. The following example adapted from Gustavsson, Ljung and Soderstrom (1977) illustrates the problem. Consider a FOPTD process modeled by equation (3.6). Let us assume that the time delay is known. Also, let the process be regulated by a proportional feedback controller using the control law: $u(k) = gy(k)$ where g is the fixed control gain. Then, all parameter estimates, $a = a^0 + \gamma g$ and $b = b^0 + \gamma$, for arbitrary γ , (where a^0 and b^0 are the true process values)

give identical values of the least square criterion. Linear dependencies in the regressor matrix cause the least square solution to be non-unique. Thus, for nonzero γ , an incorrect description of the open loop system is obtained. On the other hand, if the parameter $a = a^0$ was known *a priori*, then the process would be identifiable. Thus, the difficulty in closed-loop estimation increases with number of parameters in the model. Note, however, that the problem with lack of closed-loop identifiability, in this example, would be avoided if a higher order feedback controller is employed.

3.4 Adaptive Quadratic Dynamic Matrix Control

The objective of the QDMC algorithm is to calculate a set of input moves, $\Delta \mathbf{u}$, such that a quadratic objective function is optimized over a future prediction horizon in the presence of constraints (Garcia and Morshedi, 1986). The QDMC objective function is:

$$\varphi = (\hat{\mathbf{e}} - \mathbf{A}\Delta\mathbf{u})^T \mathbf{\Gamma}^T \mathbf{\Gamma} (\hat{\mathbf{e}} - \mathbf{A}\Delta\mathbf{u}) + \Delta\mathbf{u}^T \mathbf{\Lambda}^T \mathbf{\Lambda} \Delta\mathbf{u} \quad (3.22)$$

where $\hat{\mathbf{e}}$ represents the vector of projected deviations of the outputs from the setpoints and \mathbf{A} is a dynamic matrix, which relates the future projected outputs to the input move vector. The vector of future moves, $\Delta \mathbf{u}$, represents the solution that minimizes the objective function, φ . $\mathbf{\Gamma}^T \mathbf{\Gamma}$ weighs the output errors of the controlled variables. Excessive control moves are penalized by $\mathbf{\Lambda}^T \mathbf{\Lambda}$, the matrix of move suppression factors. QDMC implements hard process constraints by specifying linear inequalities as follows,

$$\begin{aligned} \Delta \mathbf{u}_{\min} &\leq \Delta \mathbf{u} \leq \Delta \mathbf{u}_{\max} \\ \mathbf{u}_{\min} &\leq \mathbf{u} \leq \mathbf{u}_{\max} \\ \mathbf{y}_{\min} &\leq \mathbf{y} \leq \mathbf{y}_{\max} \end{aligned} \quad (3.23)$$

The optimization problem represented by equations (3.22) and (3.23) constitutes a quadratic program. At each sampling instant, closed-loop control is achieved by implementing the first move of the optimal input profile $\Delta \mathbf{u}$. The entire cycle is repeated at each sampling instant.

In the remainder of this section, we present an analysis for closed-loop identification using DMC (Cutler and Ramaker, 1979) for a SISO system. The DMC control law is considered to be the solution of equation (3.22) while ignoring constraints in equation (3.23) completely. The conclusions drawn from our adaptive DMC analysis form the basis for our discussion of the adaptive QDMC (AQDMC) algorithm.

3.4.1 Closed-loop Identification with DMC

To study the effect of DMC algorithm on the parameter estimation scheme, consider the first-order process model:

$$y(k) = ay(k-1) + bu(k-1) \quad (3.24)$$

Although the parameter estimation is implemented incrementally using recursive least squares, the performance approaches that of the traditional, batch least squares solution,

$$\theta = (\Phi^T \Phi)^{-1} \Phi^T Y \quad (3.25)$$

where the regressor matrix, Φ , is defined as,

$$\Phi = \begin{bmatrix} y(1) & u(1) \\ y(2) & u(2) \\ \vdots & \vdots \\ y(t) & u(t) \end{bmatrix} \quad (3.26)$$

and \mathbf{Y} represents the vector of t measurements, that is,

$$\mathbf{Y} = [y(2) \quad y(3) \quad \cdots \quad y(t+1)]^T \quad (3.27)$$

The elements of \mathbf{Y} represent the process output measurements. The parameter vector, $\boldsymbol{\theta}$, consists of the unknowns a and b .

Uniqueness of the least squares solution is essential in obtaining the correct estimate of the parameter vector $\boldsymbol{\theta}$. This requires Φ to be of full rank. If the data used to construct Φ were collected during steady-state operation, the process outputs and inputs would be linearly dependent, causing Φ to be of rank one. Thus, the parameter estimator must be shut down during steady-state operation when the measurement data are no longer informative.

To investigate the effects of the adaptive scheme on the regressor matrix, we consider the DMC control law (Cutler and Ramaker, 1979):

$$\Delta \mathbf{u} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \hat{\mathbf{e}} \quad (3.28)$$

which minimizes the objective function in equation (3.22) with $\Gamma^T \Gamma$ and $\Lambda^T \Lambda$ taken as the identity and zero matrices respectively. Let us assume that the step-response coefficients in the dynamic matrix are updated by the adaptation scheme at every control instant. The $c \times p$ matrix, $(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T$ at the k^{th} instant may be represented as follows:

$$(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T(k) = [s_{ij}(a_{k-1}, b_{k-1})] \quad i = 1, \dots, c; \quad j = 1, \dots, p \quad (3.29)$$

where c and p represent the control and prediction horizons respectively. Note that elements s_{ij} are implicit functions of the model parameter estimates based on data up to $k-1$.

The vector of future projected deviations from the setpoint, $\hat{\mathbf{e}}$, can be evaluated using the parametric model in equation (3.24) as follows:

$$\hat{e}(k+i) = r_{k+i} - \left[a_{k-1}^i y(k) + \sum_{j=1}^i a_{k-1}^{i-j} b_{k-1} u(k-1) \right] - (y(k) - a_{k-1} y(k-1) - b_{k-1} u(k-1)) \quad (3.30)$$

where $i = 1, \dots, p$. Scalars r_{k+i} , refer to the reference trajectory. The term in the square bracket represents the projected model output i samples in the future based on past control actions. Thus, the step response model used by DMC to calculate the projected error is replaced by a first order ARX model. The last term in equation (3.30) represents the bias or current model-process mismatch.

Using equations (3.29) and (3.30), the DMC control law (equation (3.28)) in terms of the model parameters is,

$$u(k) + \left\{ \left(\sum_{i=1}^p s_{1i}(\alpha_{k-1}, b_{k-1}) \sum_{j=1}^{i-1} \alpha_{k-1}^{i-j} b_{k-1} \right) - 1 \right\} u(k-1) = - \left(\sum_{i=1}^p s_{1i}(\alpha_{k-1}, b_{k-1}) (\alpha_{k-1}^i + 1) \right) y(k) + \left(\alpha_{k-1} \sum_{i=1}^p s_{1i}(\alpha_{k-1}, b_{k-1}) \right) y(k-1) + \left(\sum_{i=1}^p s_{1i}(\alpha_{k-1}, b_{k-1}) r_{k+i} \right) \quad (3.31)$$

The following conclusions can be drawn from inspection of the control law:

- 1) The adaptive DMC controller is first order with respect to both the process output and input. Substitution of the control law in equation (3.26) provides the regressor matrix. The order of the controller suggests that under fairly general conditions, the columns of Φ will be independent when the data correspond to process response to deviations from the reference trajectory.
- 2) A similar analysis with a higher order SISO model,

$$y(z) = \frac{B(z^{-1})}{A(z^{-1})} u(z) \quad (3.32)$$

yields a controller with the following structure,

$$R(z^{-1})u(z) = T(z^{-1})r_{sp} - S(z^{-1})y(z) \quad (3.33)$$

where z^{-1} is the backward shift operator and r_{sp} is the setpoint. Note that r_{k+i} is the desired reference trajectory between the current measurement and the setpoint over the prediction horizon. The following relations between the degrees of the model and controller polynomials are satisfied:

$$\deg(R) = \deg(B) \text{ and } \deg(S) = \deg(A) \quad (3.34)$$

Thus, adaptive DMC has a two degree of freedom controller configuration. It is known from pole-placement design for self tuning controllers, that A and B can be uniquely determined only if polynomials R and S are of sufficiently high degree (Astrom and Wittenmark, 1995). Further, to achieve identifiability in closed-loop,

$$\deg(S) \geq \deg(A^\circ) \quad (3.35)$$

where A° is the denominator polynomial of the linear process. Equation (3.34) indicates that adaptive DMC satisfies the requirements for closed-loop identifiability provided the

selected identification model (equation (3.32)) has a degree greater than or equal to that of the underlying linear process.

3) The constant bias term ensures that the order of the controller equals the order of the estimation model. Since a high order controller is important for closed-loop identification, the bias term has a beneficial effect on parameter estimation.

4) If the biased projected future model output matches the reference trajectory, then $u(k) = u(k-1)$. During such operation, the parameter estimator must be shut down. Thus, process excitation in adaptive DMC using the model in equation (3.32) is realized only by presence of non-zero projected error over the future horizon.

5) No direct relationship emerges between parameter estimation and the prediction and control horizons. However, the control and prediction horizons serve as tuning parameters and therefore influence the model estimates.

3.4.2 Algorithm for adaptive QDMC

To ensure full rank of the regressor matrix, Φ , model adaptation should be performed only when the plant data exhibit adequate excitation. Astrom and Wittenmark (1995) discuss triggers for model updating based on calculation of covariances and spectra. However, simpler methods are often used in practice. A common method involves a criterion comparing the magnitudes of the variations in inputs and outputs with predetermined threshold values. In the current work, the recursive calculations are performed only if this criterion is satisfied. A drawback of this simple test lies in interpreting measurement noise as excitation. Another limitation is that the consistency and unbiasedness of least square estimates can be inferred only if the measurement noise is

white (Ljung, 1987; Mendel, 1995). It may be possible to alleviate these problems by designing suitable pre-whitening noise filters (Box and Jenkins, 1994) and triggers for model adaptation.

In the simulation examples presented at the end of this paper, the initial values of the model parameters are assumed to be zero. The AQDMC algorithm is initialized by a nominal control model. The nominal model represents the existing control model that is no longer fully descriptive of the underlying process. As parameter estimates begin to arrive sequentially, they are checked for convergence based on a predetermined tolerance on the rate of parameter change. If the convergence criterion is satisfied, the step response coefficients used by the DMC/QDMC controller are calculated from the estimated model and replace the nominal step response model. After the initial replacement of the nominal control model, the parameter adaptation is continued with the arrival of new input/output data, provided the excitation criterion is satisfied. The step response model is recalculated whenever a new converged parameter estimate becomes available. A schematic description of the various phases of the algorithm along a parameter estimation trajectory is described in Figure 3.2.

Use of an inadequate nominal model in the initial phase of the AQDMC algorithm may lead to poor control until the parameter convergence condition is satisfied and the adapted parameters employed. This is particularly significant if large number of parameters need to be estimated since they will typically require measurements, which are obtained sequentially, over a longer period to satisfy parameter convergence. In such

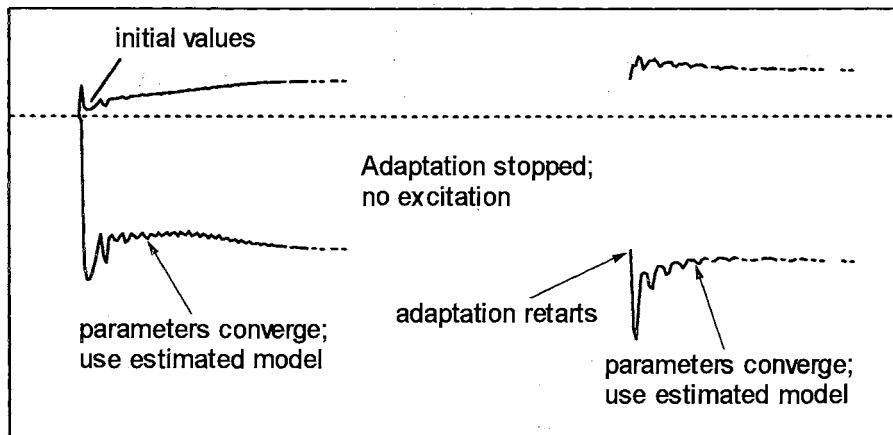


Figure 3.2: Use of estimated parameters is first made when they converge. The step response coefficients are calculated after this point whenever new converged estimates become available. Parameter adaptation is stopped when the data are no longer exciting.

cases, it may be beneficial to override the convergence criterion and use the adapted parameters for control after a pre-determined period.

Unlike DMC, the QDMC controller observes hard constraints and avoids aggressive control action by use of a move suppression factor (Garcia and Morshedi, 1986). From a parameter estimation standpoint, this has the undesirable effect of generating low energy input signals. This situation potentially slows down convergence of parameter estimates. However, move suppression and constraints are required from a process perspective and must be tolerated.

Reset action is implemented in DMC and QDMC by biasing the current predicted output to match the current measurement. This technique eliminates steady-state offset errors. In AQDMC, additional reset action is introduced by the parameter estimator. Since the parameter estimator attempts to determine the model of the process, while the controller endeavors to keep the controlled variables at their respective setpoints, it is expected that the adaptive controller will compensate for steady state errors, thereby minimizing the impact of the biasing operation. The flowchart depicting the adaptive QDMC algorithm is shown in Figure 3.3.

3.4.3 Conversion of identification model to step response model

The model for process identification can be described by equation (3.6) or (3.8) depending on the choice of model structure. Conversion to step response model is achieved by assuming all initial conditions to be zero and implementing a step input, $u(k) =$

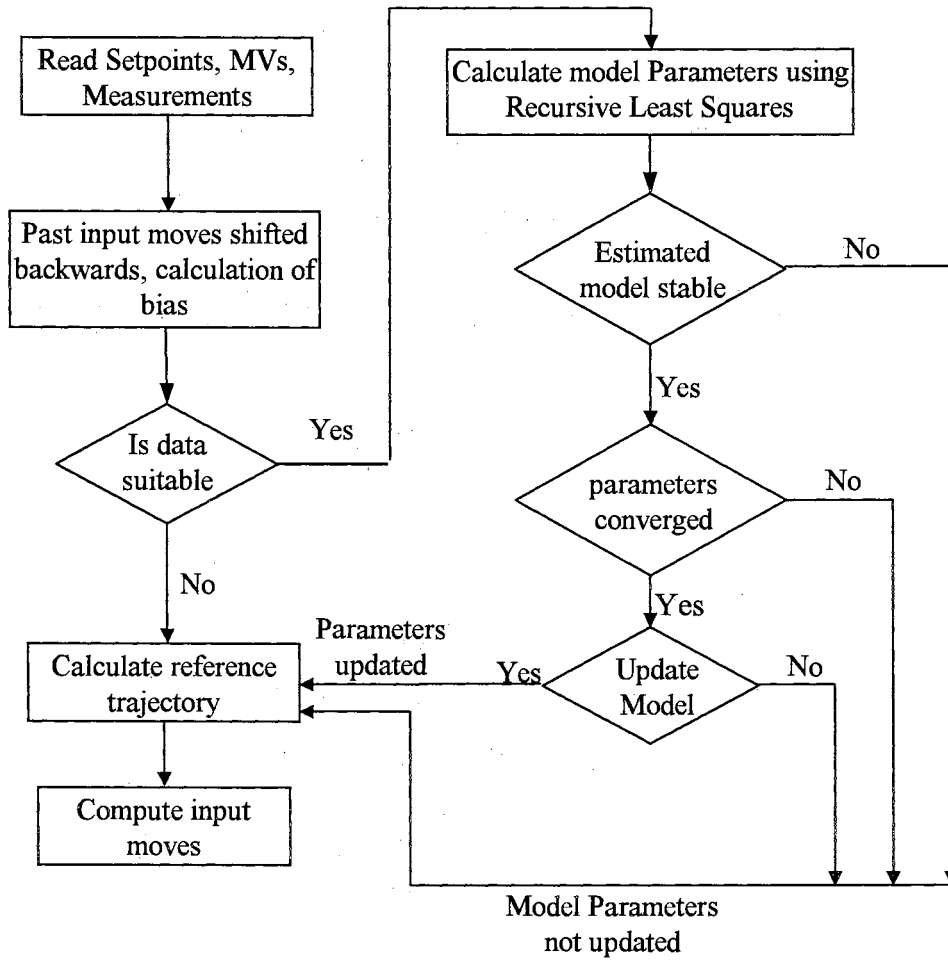


Figure 3.3: Adaptive QDMC algorithm.

1, $k = 0, 1, \dots$ to the process. The step response coefficients are then calculated recursively as the response to the input sequence.

3.5 Simulation Examples

The AQDMC algorithm was tested using simulations with MATLAB. The optimization of the quadratic objective function (equation (3.22)) including inequality constraints (equation (3.23)) was performed using the standard quadratic programming function (*qp.m*) available in MATLAB. In the following examples, the initial values of the unknown model parameters are assumed to be zero. The process can be approximated using a first-order-plus-time-delay model. The control horizon was set at 5 sample periods while the prediction horizon used was 80 sample periods. The error penalty and move suppression matrices were diagonal with identical non-zero elements of 1 and 0.02, respectively. Parameter estimation was performed only when the absolute value of change in either the input or output values exceeded 2E-04. The estimated model first replaces the initial nominal model when the variation in parameter estimates dropped to less than 2% of the previous value. Subsequent model updates are performed whenever the convergence criterion of 2% is satisfied. Model parameters are adapted whenever the excitation condition is satisfied.

Example 1: SISO and known time delay: Consider the SISO process where the actual process can be characterized as:

$$h_{\text{process}}(z) = \frac{0.0154z^{-4}}{1 - 0.8187z^{-1}} \quad (3.36)$$

For this case we assume it is known that the time delay of the process is 3 sample periods while the other parameters are unknown. The initial control model is:

$$h_{\text{initial model}}(z) = \frac{0.0028z^{-4}}{1 - 0.9623z^{-1}} \quad (3.37)$$

A setpoint change of 0.01 is introduced. The constraints are defined as:

$$0.25 \geq \Delta u_{k+j} \geq -0.25; \quad 1 \geq u_{k+j} \geq -1, \quad \text{for } j = 1, \dots, 5,$$

$$0.02 \geq y_{k+j} \geq -0.02, \quad \text{for } j = 1, \dots, 80$$

Figures 3.4(a) and 3.4(b) show the output response and input for the process. The dashed curve represents performance of the QDMC algorithm while the solid line represents the AQDMC algorithm. Figure 3.4(c) shows the estimation history for the model parameters. The nominal model (equation (3.37)) is replaced by the estimated model at $k = 25$ when the model parameters converge. Thus, the AQDMC and the QDMC results are identical until $k = 25$, after which the AQDMC controller adapts the model parameters whenever the excitation condition is satisfied. At $k = 71$, the process reaches steady state operation and the input-output data contain no new information. Here, the parameter estimator is shut down.

A comparison of the step test results of the estimated model with the actual process is shown in Figure 3.4(d). It is seen that the initial estimated model used at $k = 25$ does not approximate the process well. However, the quality of the estimates improves quickly. Since the process and the model used by the estimator conform to the same structure, it is possible to compare the estimated parameters with the actual values.

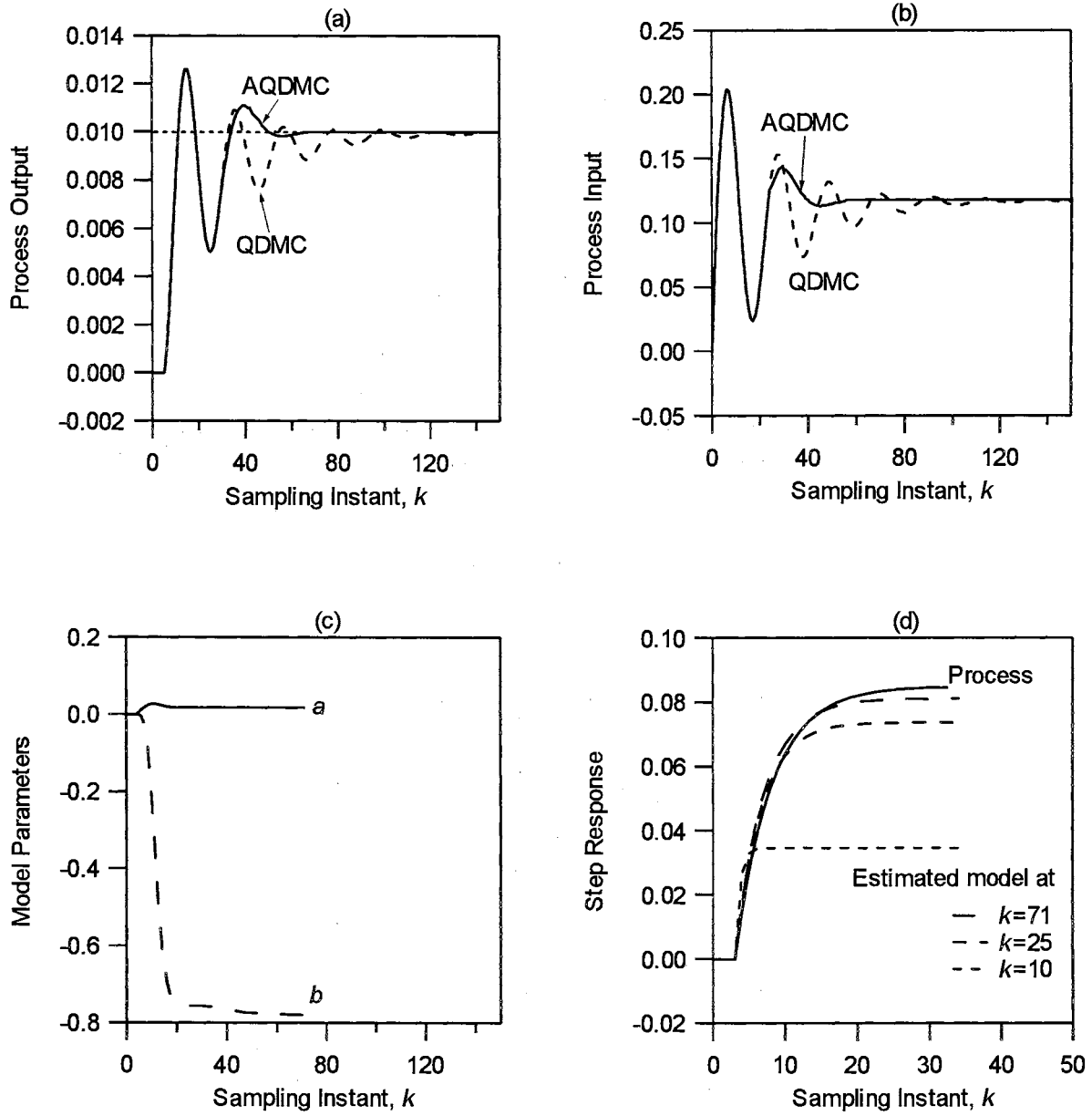


Figure 3.4: Example 1: Comparison of AQDMC and QDMC algorithms used to control a SISO process. Both the process and the model structure used by AQDMC are as shown in equation (3.5). The time delay of the process is known.

It is seen from Figure 3.4(c) that the estimated parameters approach the true values specified in equation (3.38). At $k = 71$, the estimated model is:

$$h_{\text{estimated model}}(z) = \frac{0.018z^{-4}}{1 - 0.779z^{-1}} \quad (3.38)$$

Thus, the ability of the AQDMC controller to effectively “learn” the process improves control performance as expected.

Example 2: SISO and unknown time delay with white noise. This example illustrates the performance of the AQDMC algorithm in the presence of white measurement noise and unknown time delay. Let the process and the initial model be represented by equations (3.36) and (3.37) as before. Also, let the constraints be the same as in the previous example. In this case, we also assume the time delay is unknown. Since the time delay, d , associated with the process is unknown, equation (3.6) is no longer linear in parameters and the model represented by equation (3.8) must be employed. Gaussian white noise with zero mean and standard deviation 0.001 is added to the output measurement.

The process outputs and inputs in response to a setpoint change of 0.01 are shown in figures 3.5(a) and 3.5(b), respectively. A measurement filter is not used. The parameter estimation history is shown in figure 3.5(c). The controller replaces the initial model with the estimated model at $k = 25$. Figure 3.5(d) illustrates the step response of the model at various stages of identification. The estimated model exhibits an inverse response due to a zero located outside the unit circle (at $-b_2/b_1 \approx 1.7$, for the estimated

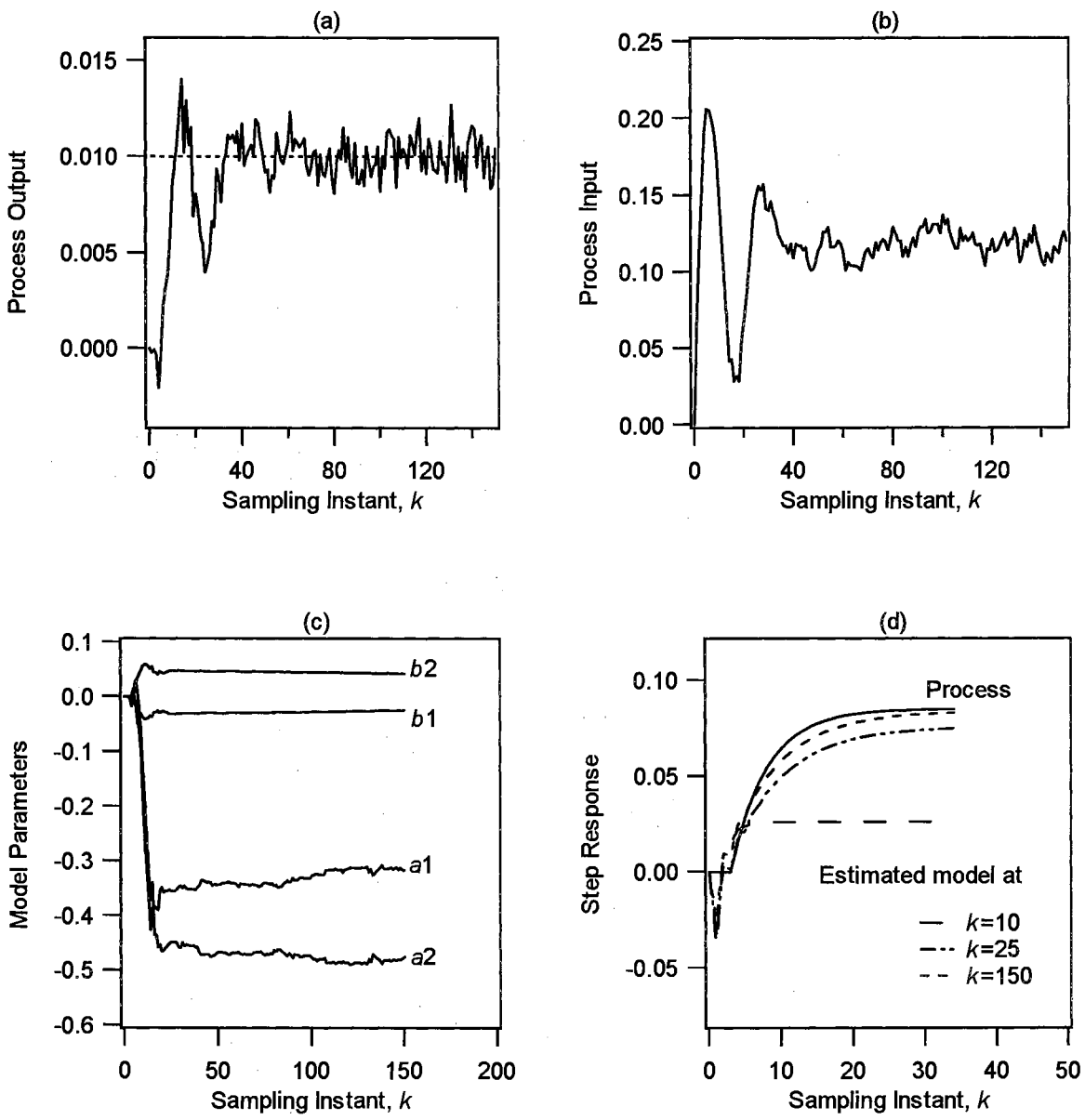


Figure 3.5: Example 2: Performance of the AQDMC algorithm in presence of white noise.

model at $k = 150$). This represents an undesirable trait introduced by use of overparameterized model structure. The estimated model is:

$$h_{\text{estimated model}}(z) = \frac{-0.025z^{-1} + 0.042z^{-2}}{1 - 0.316z^{-1} - 0.476z^{-2}} \quad (3.39)$$

Direct comparison of the estimated model to the actual process model is not possible because of the different model structures. However, the gains for both models can be calculated using the final value theorem. The gains for the estimated and actual process models are 0.082 and 0.085, respectively.

Example 3: MIMO and unknown time delay: Consider a 2x2 process whose dynamics are defined as:

$$\mathbf{G}_{\text{process}}(s) = \begin{bmatrix} \frac{0.006z^{-4}}{1 - 0.92z^{-1}} & \frac{-0.0043z^{-3}}{1 - 0.9355z^{-1}} \\ \frac{0.0096z^{-4}}{1 - 0.9184z^{-1}} & \frac{-0.0117z^{-3}}{1 - 0.9066z^{-1}} \end{bmatrix} \quad (3.40)$$

The process constraints are defined as:

$$\begin{aligned} 0.25 \geq \Delta u_{1,2k+j} \geq -0.25; \quad 2 \geq u_{1,2k+j} \geq -2, \quad \text{for } j = 1, \dots, 5, \\ 0.04 \geq y_{1,2k+j} \geq -0.04, \quad \text{for } j = 1, \dots, 80 \end{aligned}$$

We assume no information is known regarding the process delays. The parameterized model structure used in the identification step is based on equations (3.18) and (3.19).

The initial model used by the QDMC controller is

$$\mathbf{G}_{\text{initial model}}(s) = \begin{bmatrix} \frac{0.0022z^{-5}}{1-0.9592z^{-1}} & \frac{-0.0032z^{-2}}{1-0.9672z^{-1}} \\ \frac{0.005z^{-6}}{1-0.9535z^{-1}} & \frac{-0.0042z^{-2}}{1-0.9512z^{-1}} \end{bmatrix} \quad (3.41)$$

At $k = 0$, a setpoint change of 0.02 is introduced for y_1 and is stepped back to 0.0 at $k = 101$. Figures 3.6(a) and 3.6(b) show the response of the two process outputs controlled by the QDMC and the AQDMC controllers. Due to the large process-model mismatch, the QDMC controller is unable to control the process and the outputs settle at steady values due to non-availability of input resources as defined by the constraints (i.e. $u_{1,2 \min} = -2$). The AQDMC controller uses the initial model described by equation (3.41) until $k = 30$. Although the model estimates have not attained steady values (see Figure 3.7(a)), we override this criterion and employ the intermediate estimated model at each control execution. Now, the dynamic matrix is time varying and provides additional input excitation to the process. Although the process response moves towards the setpoint, the performance is sluggish. The process does not attain the desired output values until $k = 100$ when the new setpoint is implemented. Now, the adaptive system shows improved performance due to improved model parameter estimates based on a larger amount of data available to calculate the estimates.

The estimation trajectories of some of the model parameters are shown in Figure 3.7(a). The significant change in slopes at $k = 101$ shows the increase in the rate of estimation of parameters. This is due to the availability of informative input-output process data in response to the new setpoint change. The step response of the converged model at $k = 198$ is shown in Figure 3.7(b).

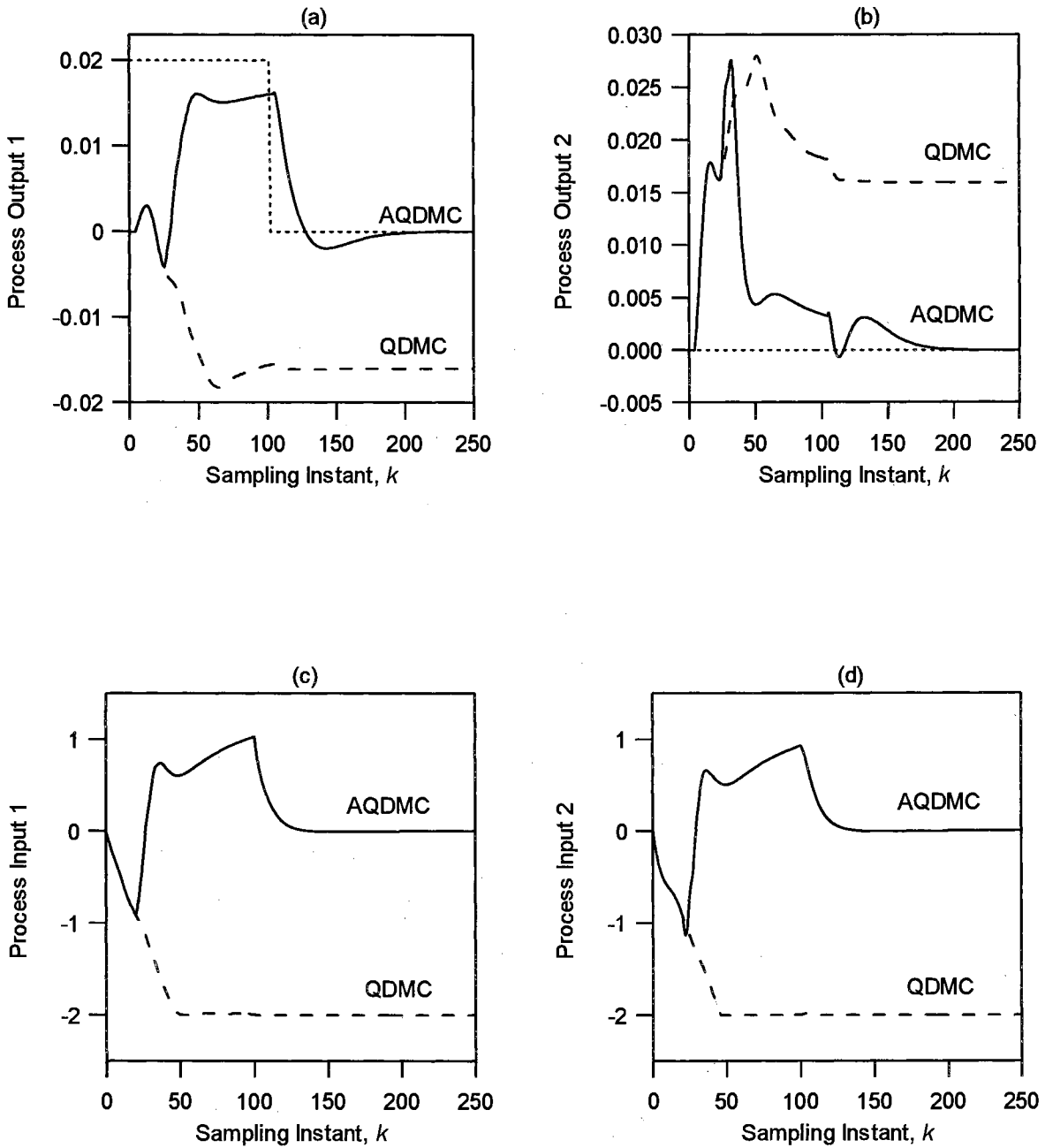


Figure 3.6: Example 3: Comparison of AQDMC and QDMC algorithms used in the control of a 2 x 2 process. At $k = 0$, the setpoint for output 1 is set to 0.02. At $k = 101$, the setpoint is stepped back to 0.0.

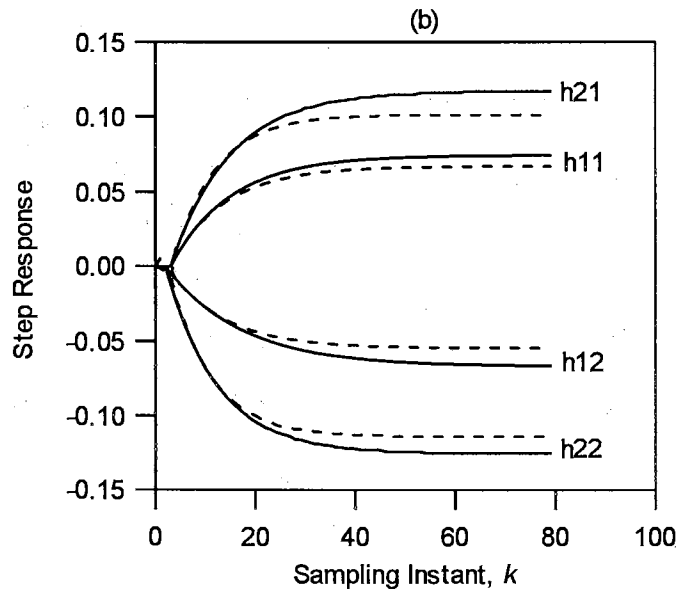
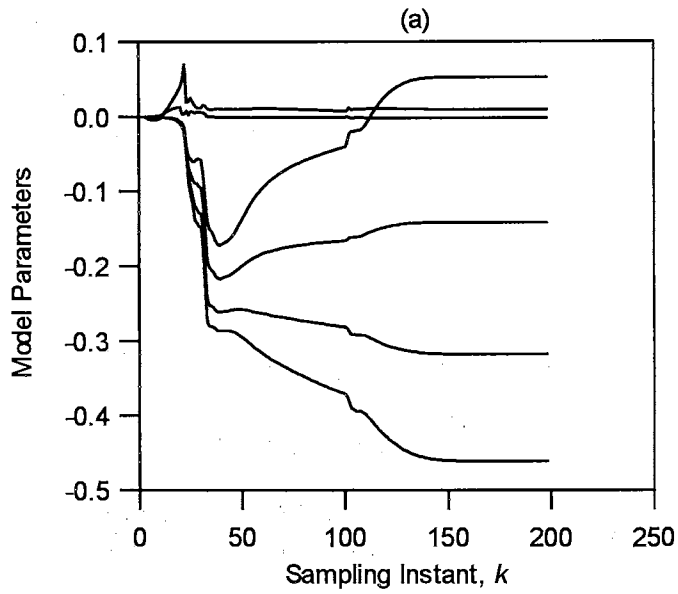


Figure 3.7: Example 3: Model parameter estimation by the adaptive QDMC algorithm: a) Estimation trajectory of some of the model parameters; b) Step response of the estimated model at $k = 198$.

Example 4: Nonlinear CSTR As a final example, we consider control of a non-adiabatic, continuous, stirred-tank reactor with a first order irreversible reaction. The heat of reaction is removed by circulation of cooling water in the reactor jacket. Uppal et al. [1974] describe the following system of equations that govern the process dynamics,

$$\dot{x}_1 = -x_1 + Da(1-x_1) \exp\left(\frac{x_2}{1+x_2/\gamma}\right) \quad (3.42a)$$

$$\dot{x}_2 = -x_2 + BDa(1-x_1) \exp\left(\frac{x_2}{1+x_2/\gamma}\right) + \beta(u-x_2) \quad (3.42b)$$

States x_1 and x_2 represent reactant conversion and a dimensionless reactor temperature. The manipulated variable, u , is a dimensionless temperature of the reactor jacket. The steady state characteristic of the reactor for parameters, $\beta = 3.0$, $\gamma = 40$, $B = 22$ and $Da = 0.082$, is shown in Figure 3.8. The process exhibits low gain at small values of conversion (1% to 4%) and considerably higher gain (in excess of 80 times the low gain) at higher conversions (>18%). A linear model is developed by conducting a step test that describes the relationship between u and x_1 at low conversions (in the vicinity of 1% to 4%),

$$h_{\text{initial model}}(z) = \frac{0.0063z^{-1}}{1-0.3886z^{-1}} \quad (3.43)$$

Performance of a DMC controller which employs the above model for various setpoint changes is shown in Figure 3.9(a). At low conversion values, the DMC controller shows adequate performance. However, when a new setpoint ($x_1 = 0.18$) is implemented, the DMC controller does not recognize the high plant gain in the new

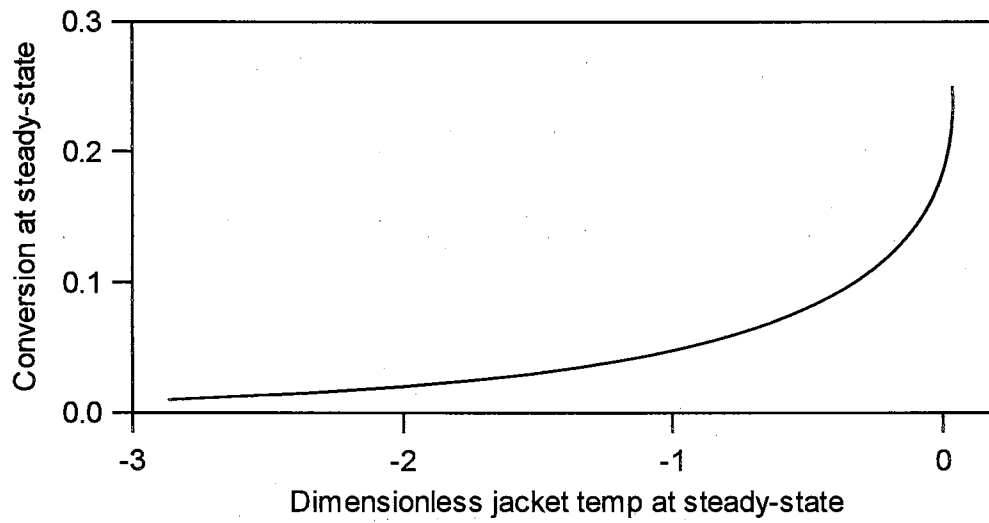


Figure 3.8: Example 4: Steady-state behavior of CSTR for $\beta = 3.0$, $\gamma = 40.0$, $B = 22$ and $Da = 0.082$.

region, resulting in aggressive control action. The system becomes unbounded when the next setpoint change (from 0.18 to 0.19) is implemented.

For purposes of adaptation, a model structure of equation (3.6) is selected. The process delay, d , is assumed to be zero. The solid curve in Figure 3.9(a) shows the corresponding behavior of the adaptive DMC controller. Model adaptation is triggered at $k = 300$ units, when the setpoint is changed to 0.18. The model parameters are estimated for the high-gain environment (see Figure 3.9(c)) and used for control. At $x_1 = 0.18$, the estimated model steady-state gain was 0.433 (at $k = 390$, when the adaptation was turned off due to lack of excitation) while the process steady-state gain = 0.55. The nominal model gain remains unchanged at 0.0103 which is inaccurate at the new setpoint. When the setpoint is changed to 0.19, the parameter adaptation continues and the controller successfully steers the plant to the new setpoint. Figure 3.9(d) shows the plant model mismatch for the non-adaptive and adaptive versions of DMC. The mismatch is small when the adapted model is used relative to standard DMC.

For adaptive control of the nonlinear CSTR problem, two modifications were made to the AQDMC algorithm vis-à-vis previous examples. (1) a forgetting factor of 0.92 was used with the recursive least squares algorithm, which discounted old measurements and emphasized recent ones, (2) the adapted FOPTD model in deviation form was constructed around the steady-state point ($x_{1,s} = 0.18$, $u_s = -0.007$).

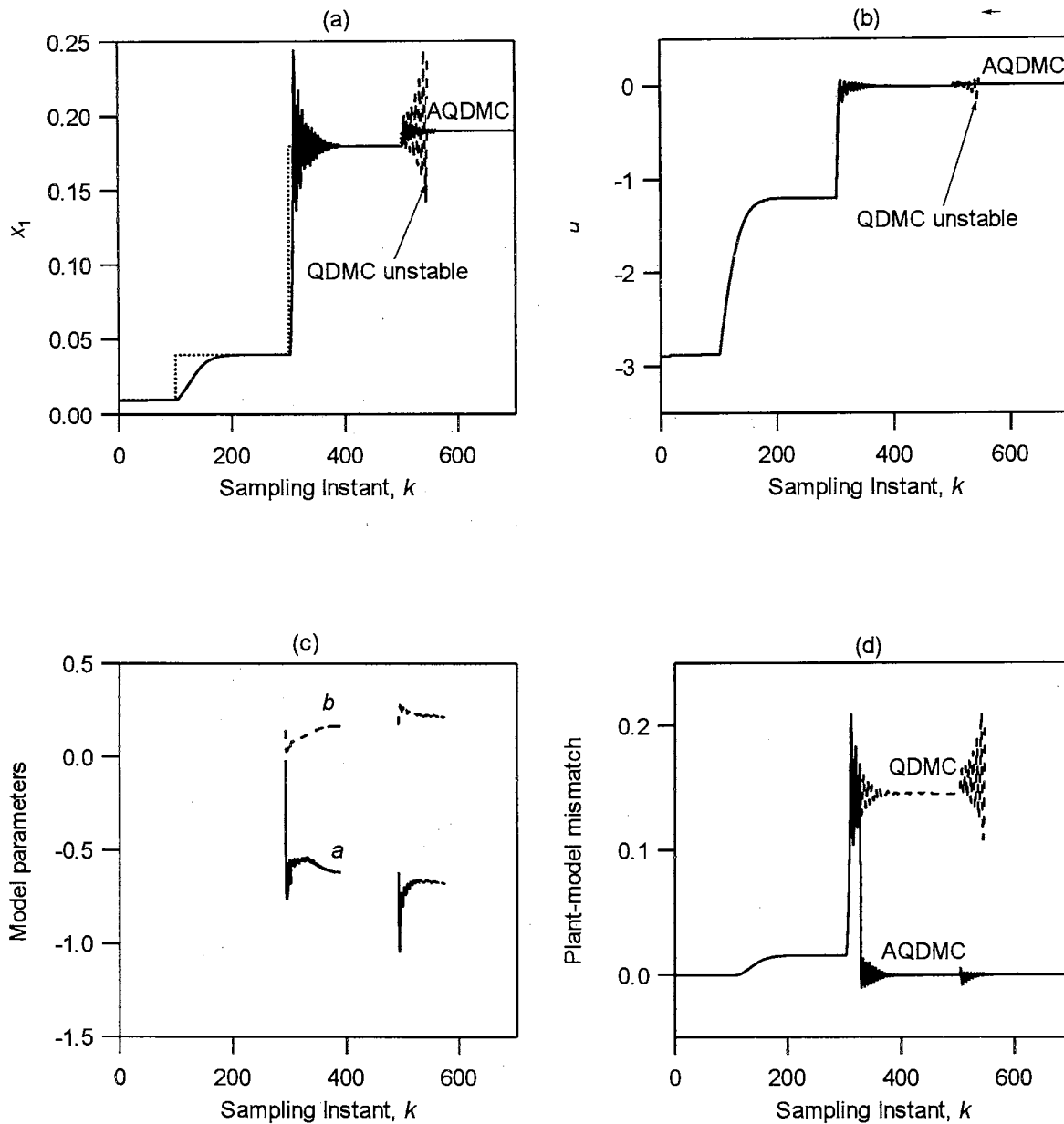


Figure 3.9: Example 4: Comparison of adaptive DMC and non-adaptive DMC performances for control of nonlinear CSTR. The nominal model was developed for low conversions (1 to 4% conversion). The DMC system response is unbounded for operation in high conversion regions while the adaptive DMC controller modifies controller parameters to reflect larger gain in the high conversion region.

3.6 Conclusions

Performance of the quadratic dynamic matrix control algorithm is sensitive to the presence of model parameter errors along with other sources of process-model mismatch. To overcome model parameter errors, we proposed an online model estimation technique using the recursive least square algorithm which can be integrated with QDMC to provide an adaptive QDMC algorithm. The usefulness of the algorithm was demonstrated through simulation examples.

An analysis of adaptive DMC was performed to show that the controller is closed-loop identifiable provided the identification model is of sufficiently high degree. In this study, processes well described by FOPTD models were considered. The efficacy of the adaptive technique for these processes was successfully demonstrated for linear and nonlinear SISO systems. The benefits of leveraging *a priori* information were also illustrated.

For MIMO systems, the use of AQDMC controllers is computationally more involved due to the large number of parameters required to describe such processes. Estimation of a large number of parameters requires a large data set. The data are obtained sequentially from the process and hence, the time required to obtain a reasonable estimate of the model is increased. However, this problem can be alleviated if model uncertainty affects only a smaller subset of the parameters which need to be estimated.

In this work, measurements were used for adaptation when variations in inputs and outputs were larger than a predetermined threshold value. This method worked well for the noise-free simulation examples. However, an issue that remains to be addressed is development of suitable adaptation triggers in presence of measurements corrupted by noise.

4 DEVELOPMENT OF INFERENTIAL MEASUREMENTS USING NEURAL NETWORKS

Chapter Overview

In many industrial processes, the most desirable variables to control are measured infrequently off-line in a quality control laboratory. In these situations, application of advanced control or optimization requires use of inferred measurements generated from correlations with measured process variables. For well-understood processes, the form of the correlation as well as the choice of inputs may be known. However, many industrial processes are too complex and the appropriate form of the correlation and choice of input measurements are not obvious. Here, process knowledge, operating experience, and statistical methods are crucial in development of correlations to be used for inferential measurements.

This chapter describes a systematic approach to development of nonlinear correlations for inferential measurements using neural networks. A three-step procedure is proposed. The first step consists of data collection and preprocessing. In the second step, the process variables are subjected to simple statistical analyses to identify a subset of measurements to be used in the inferential scheme. The third step involves generation of the inferential scheme. We demonstrate the methodology by inferring the ASTM 95% endpoint of a petroleum product using actual data from a U.S. domestic refinery.

4.1 Introduction

Methods for online control and optimization of processes are based on reliable and accurate measurements of key variables. However, not all important variables can be measured in real-time to allow for timely action based on their measurements. The lack of key measurements can be attributed to various factors (Marlin, 1995): (1) insufficient automation of some sensitive analyses without human intervention; (2) even if real-time measurement is possible, the cost of installing an additional sensor may not be economically attractive. The hard-to-measure variables usually represent product quality or are of direct economic interest. Often, these variables are inferred by correlations involving available measurements. The inferential model provides an estimate of the variable, which can then be incorporated in control and monitoring schemes.

Although inferential models are widely used in industry, only a few techniques of inferential model development have been discussed in open literature. Kresta et al. (1996) presented model development using partial least squares (PLS). The efficacy of the method was demonstrated by inference of the heavy key composition in the distillate. The independent latent variables were constructed using various temperature and flowrate measurements. Qin et al. (1997) constructed soft sensors using a principle component analysis (PCA) approach for continuous monitoring of emissions. Both of the above approaches are purely data-driven. The inputs to the inferential model are linear combinations of the measurements (latent variables in PLS, principle components in PCA) such that they describe the significant variability of the data set. Thus, while the

resulting correlation is often adequate, it fails to provide an intuitive sense of dependence of the inferred variable on the measurements.

In this work, we adopt the route of variable selection followed by regression. Thus, unlike the above approaches, only those measurements that have a significant influence on the inferred variable are included in the correlation. Selection and use of the measured variables in the inferential model requires considerable process insight. In case of physically large and highly integrated processes, enumeration of candidate variables based on process insight alone may not be feasible. Moreover, if the set of candidate measured variables is large, development of the correlation can easily become a time-consuming procedure. Identification of variables to be employed in the correlation is accomplished using simple statistical tools in conjunction with process knowledge. The correlation is then developed with the aid of regression using neural network models.

Section 4.2 discusses a three-step procedure to develop inferential measurements beginning with collection of data from the process. Section 4.3 describes a situation from the refining industry, which requires inferential measurements to control the process. Sections 4.4, 4.5, and 4.6 illustrate the three-step procedure to generate inferential measurements as applied to the situation described in Section 4.3. Finally, conclusions are presented in section 4.7.

4.2 Methodology

Large and complex processes contain a number of controlled and monitoring variables and a larger number of measured variables. If a desired variable y cannot be measured, then it must be inferred using a suitable subset of p measured variables, $\{x_i, i=1,p\}$ selected from the larger set of n candidate variables. The choice of the n measured variables may be based on the investigator's past experience or process insight. Thus, the problem of inferential measurement may be decomposed into three sub-problems:

- a) data collection and preprocessing, i.e. the variable we wish to infer and the candidate set of n measured variables;
- b) identification of a subset of p measured variables which will be used in the inference of the unmeasured variable, y ;
- c) approximation of the relationship between the inferred variable, y , and the subset identified in sub-problem (b), $\{x_i, i=1,p\}$, using neural networks.

Thus, we seek a correlation of the form:

$$y = f(\{x_i, i=1,p\}) \quad (4.1)$$

Modeling by linear regression uses a similar three-step procedure. The number of predictor variables included in the linear model fixes the number of model parameters and hence its complexity. For instance, a model with three predictors contains four parameters (including the intercept). However, in neural network models, the model complexity depends on the number of nodes in hidden layer, in addition to number of predictor variables. Sub-problem (b) keeps the model complexity in check with respect to number of predictor variables. In this work, the number of nodes in hidden layer is chosen by trial and error based on minimum mean square error.

In the remainder of this chapter, the unmeasured variable, y , will be referred to as the dependent or response variable and the measured variables may occasionally be referred to as the independent or regressor variables. The following three subsections describe each sub-problem.

4.2.1 Sub-problem (a): Data Collection and Preprocessing

The first step in the development of a correlation is collection of data consisting of the dependent and the candidate independent variables. Experience and process insights often guide the choice of the candidate independent variables. It is important to choose all variables, which can potentially influence the inferred dependent variable. It is expected that sub-problem (b) will screen out irrelevant independent variables from the subset of independent variables which will be employed in the correlation.

Further, the data required to develop the model must reflect the conditions under which it will be used. Let us assume that the correlation developed will be used to infer a variable during steady state operation of the process. Thus, due care must be taken to ensure that the data indeed reflects steady state conditions. To remove the effect of local transients, the sampled process variables like pressure, temperature and flow may be averaged over a suitable time period. Moreover, the observations must cover the entire range of operating conditions. The data set may be augmented by composite variables like stream enthalpies, heat duties of equipment, etc. that play an important role in characterizing process behavior. A schematic of the data matrix is shown in Figure 4.1.

	dependent	direct measurements		augmented variables	
	y	x_1	x_n
Obs No. 1					
Obs No. 2					
Obs No. m					

Figure 4.1: Schematic of the matrix of collected data. The first column refers to the inferred variable. The subsequent columns contain measurements of candidate independent variables. Augmented variables refer to variables such as enthalpy, product yield, etc. which cannot be measured but calculated using direct measurements.

The raw data thus collected may contain observations that are inconsistent with the statistical character of the remainder of the data set. These outlying observations can have an unwarranted influence on the model estimates (Cook and Weisberg, 1980). A difficulty in detecting such multivariate outliers is that the observation itself may not be extreme on any of the independent variables and therefore not apparent on the plots of two variables at a time. In this work, we employ principle components (PCs) in a fairly simple way to identify multivariate outliers as discussed by Jolliffe (1986). Principal components represent an orthogonal transformation of the data so that the variances of the derived coordinates are in decreasing order of magnitude. The first few principal components refer to directions associated with high variance of observations. The last few PCs refer to directions associated with small variance. Jolliffe suggests the use of following test statistics to identify outliers:

$$\begin{aligned}
 d_{1i}^2 &= \sum_{k=n-q+1}^n z_{ik}^2 \\
 d_{2i}^2 &= \sum_{k=n-q+1}^n \frac{z_{ik}^2}{l_k} \\
 d_{3i}^2 &= \sum_{k=1}^n l_k z_{ik}^2
 \end{aligned} \tag{4.2}$$

where z_{ik} is the value of the k^{th} PC for the i^{th} observation, n is the number of variables, q represents the low variance PCs (for example, variance < 1) and l_k is the variance of the k^{th} sample PC. Statistics d_{1i} and d_{2i} are designed to detect observations that do not conform to the correlational structure of the data. This is evident from the definition of the statistics since only the last few PCs are considered in their evaluation. Statistic d_{2i} penalizes observations associated with low variance PCs more heavily than d_{1i} . Statistic

d_{3i} is designed to detect observations that inflate the variance of the data set. All PCs are considered in evaluation of the d_{3i} statistic.

The resultant data set obtained after deletion of outlying observations is used in sub-problems (b) and (c) to obtain a correlation for prediction of y having the form of equation (4.1). Sub-problem (b) deals with the issue of selecting a suitable subset $\{x_i, i=1,p\}$ from the set of n candidate variables whereas sub-problem (c) involves approximation of the function f using neural networks.

4.2.2 Sub-problem (b): Identification of subset of variables for regression

Researchers often collect data on a response variable and several potential predictor variables during experiments. As discussed in section 4.2.1, additional predictor variables are frequently created by taking functions of the observed predictor variables. In the traditional method of obtaining a correlation using multiple linear regression, the hazard of using too many predictor variables is widely known. The addition of a variable to least square prediction equation almost always increases the variance of the predicted response (Walls and Weeks, 1969; Allen, 1971). Addition of predictor variables may decrease the squared bias, but this decrease is often small relative to the increase in variance. Consequently, the correlation based on large number of predictors is very sensitive to noise and results in a non-robust model. To overcome this problem, regression analysts focussed on determining methods to identify a subset of the original set of predictor variables (Hocking, 1983).

Often, the subset of regression variables may be partially identified by the investigator based on his/her experience and understanding of the process. Additionally, one may use statistical tools that identify relationships between groups of variables. This stage involves a certain degree of judgment and an art of “informal conversation” with the data. If the investigator does not make a judicious choice during variable selection but instead selects variable indiscriminately, the resulting model will be less robust and the irrelevant variables may mask or replace the effects of the more important variables. We discuss three techniques, viz., (a) scatter plots and simple correlation coefficients; (b) partial correlation coefficients; (c) Mallows’ C_P statistic.

4.2.2.1 Scatter Plots and Simple Correlation Coefficients

A graph of each regressor versus the dependent variable on a two-dimensional plot enables the investigator to visually search for underlying relationships. The pattern of points on the graph represents the relationship between the variables. Organization of the points along a straight line indicates linear relationship. A curved set of points may denote that the relationship is nonlinear. Absence of a pattern may denote that no significant relationship exists between the two variables. Simple correlation coefficients provide a measure of linear association between the two variables and thus aid in the selection of independent variables that exhibit a linear relationship with the dependent variable. It is observed that if the inherent relationship between the two variables is nonlinear, then the absolute value of the correlation coefficient may be far less than unity. However, our judgment will be made using scatter plots and comparative values of the

correlation coefficients between each of the independent variables and the dependent variable and not on the magnitude of the coefficient itself.

4.2.2.2 Partial Correlation Coefficient

Calculation of simple correlation coefficient between the independent and response variable ignores the effects of the other candidate independent variables. In a multivariable case, the value of the correlation coefficient may be masked by influences of other variables. In such situations, partial correlation coefficient (Steel and Torrie, 1980) may be employed to view linear association between two variables, say x_i and x_j , by “adjusting for” the effect of other variables, $(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_{j-1}, x_{j+1}, \dots, x_n)$. It is calculated using the matrix of simple correlation coefficients $[R_{ij}]$ as follows,

$$r_{ij,1,\dots,i-1,i+1,\dots,j-1,j+1,\dots,n} = \frac{-C_{ij}}{\sqrt{C_{ii}C_{jj}}} \quad (4.3)$$

where $r_{ij,1,\dots,i-1,i+1,\dots,n}$ is the partial correlation between the variables x_i and x_j . C_{ij} represents the ij^{th} element of the inverse of the simple correlation coefficient matrix $[R_{ij}]$. Thus, partial correlation coefficients have been employed in this work to identify those independent variables, which have significant prediction effect on the response variable, from the list of n independent variable. As with simple correlation coefficients, we seek those independent variables that exhibit dominant association with the dependent variable.

Based on scatter plots, simple and partial correlation coefficients a preliminary subset of r independent variables, $\{x_i, i=1,r; r < n\}$, is selected. If the number of candidate

independent variables, n , is large, the preliminary elimination of $(n-r)$ variables reduces the quantity of independent variables to a manageable number for further study. In the next step, the choice of the r regressor variables is further scrutinized.

4.2.2.3 Mallows' CP statistic

Partial correlation coefficient calculation eliminates the effect of other candidate regressors when studying the relationship between a single regressor variable x_i and independent variable, y . However, it does not consider selection of a subset of predictors from a larger set. Tasks designed to select a subset of predictor variables involve examining some criterion like the coefficient of determination, commonly known as R^2 . R^2 is defined as the fraction of the variation in the independent variable measurements explained by the regression model and equals unity if the fitted equation passes through all the data points. However, as discussed previously, the variance of the model response always increases with the inclusion of additional predictor variables. Thus, it cannot be used to determine the “best” choice of model subset when the number of variables in candidate subsets may vary. In such cases, criteria that penalize model complexity are more suitable for subset selection. Mallows's C_P statistic (Mallows, 1973) , Akaike information criterion (Akaike, 1974) and Bayesian information criterion (Schwarz, 1978) among others have been widely used to evaluate model complexity. Below, a brief description of the C_P statistic employed in the current work is provided.

Mallow suggests that the 'standardized total squared error' be used as a criterion and he developed an estimate of this quantity called the C_p statistic. It is defined as follows:

$$C_p = \left(\frac{\text{residual sum of squares for subset model with } p \text{ parameters including an intercept}}{\text{residual variance for full model}} \right) - (r - 2p) \quad (4.4)$$

where p denotes the number of predictor variables selected in the regression model from the larger set of r variables in the data set. Good models typically have the (p, C_p) coordinate close to the 45 degree line on a C_p v/s p plot. Since this method inspects all combinations of variables to provide a good subset of predictors to be used in the regression scheme, the number of possible subsets grows very rapidly with the number of variables. For a set of n candidate predictor variables, the total number of combinations is $2^n - 1$. Hence, the C_p statistic was calculated only for the $2^r - 1$ subsets that could be constructed from the preliminary subset identified in sub-problem (b). Methods to decrease computational effort in subset selection have been discussed by Hocking and Leslie (1967) and LaMotte and Hocking (1970). Use of the C_p statistic in this study was a means of refining the choice of the subset of predictors from an initial subset constructed by using simple and partial correlation coefficients.

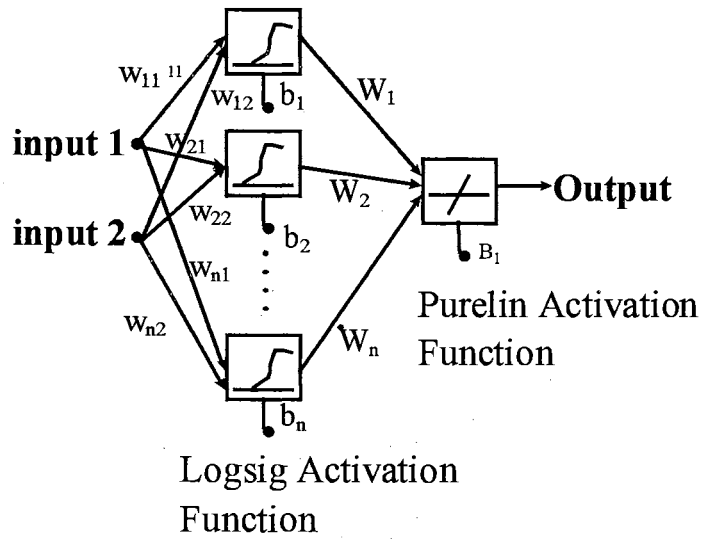
Using the above techniques, a set of regressor variables, $\{x_i, i=1, p\}$, from the set of n original measured variables is identified. In the final stage, the correlation is developed between the identified subset and the dependent variable using regression.

4.2.3 Sub-problem (c): Approximation of the relationship between the unmeasured variable, y , and the measured variables, $\{x_i, i=1,p\}$ using neural networks

The last step in development of the inferential measurement is building models using the identified set of $\{x_i, i=1,p\}$ to predict the unmeasured variable, y . If no specific *a priori* information exists about the model, one may take recourse to non-parametric modeling using neural networks. Good approximation of nonlinear functions also makes neural network the choice of regression. In this work, we considered 2-layer feedforward neural networks as shown in Figure 4.2. The network is trained using backpropagation algorithm. The performance of the network will be a reflection on the (i) success of outlier detection; (ii) adequacy of the selected subsets of variables; (iii) the architecture of the neural network. Equivalently, poor performance can be attributed to failure of any one or a combination of the three steps. This shows the coupled nature of each task undertaken in this work, thereby, making empirical model-building an iterative procedure. We briefly discuss two aspects of regression using neural networks: (a) training methodology; (b) regression refinement.

4.2.3.1 Training Methodology

Training neural networks is a data-analytic procedure and does not impose a stochastic framework on the training set. Under these conditions, it is necessary to stop training once an overfit is indicated. One way of doing this is with the use of cross-validation (Hush and Horne, 1993). The original data set is split into two subsets. One subset is used for training while the other is used for validation. The weights and biases obtained by using the training subset are applied to the validation subset to evaluate the



Inputs → **Layer 1** → **Layer 2** → **Output**

Figure 4.2: Architecture of multilayer perceptron neural network used in construction of inferential model. The inputs to the network are determined in sub-problem (b). Sub-problem (c) determines the optimal number of nodes in hidden layer .

performance in terms of the sum of squared errors. Typically, the sum of squared errors (SSE) for the training subset decreases with the number of iterations and perhaps levels off to some constant value when a local minimum is attained. In an overfit situation, SSE for the validation set decreases at first, but then comes to a minimum and later increases though the SSE of the training set continues to decrease. When the SSE of the validation subset increases, it is assumed that the regressions algorithm is over-fitting the training data. In the current work, the training was stopped as soon as SSE over the validation set began to increase.

4.2.3.2 Regression Refinement

The cross-validation training approach ensures that converged values of weights and biases are not strongly influenced by few outlying observations that may be present in the training set. Thus, a comparison of neural network prediction with the actual data set (training + validation) gives an indication of those observations that do not conform to regression. In the example presented in the following section, those observations whose network prediction error was large (greater than 2 to 3 standard deviations from the error mean) were deleted from the data set and the training procedure repeated.

It is assumed that the outlying observations detected by the cross-validation training method represent unsteady state measurements, which are inappropriate for model development. Moreover, such observations may significantly affect the regression parameters. However, an accompanying risk in automating outlier detection as described, is the possibility of deleting observations that indicate a bona fide operating

condition. The resulting data set (after deleting outliers detected by the network trained by cross-validation) may be used again to generate a new set of weights and biases by the training method described above.

The methods described above represent a few techniques from the vast volume of literature available. For instance, numerous techniques to detect outliers are described by Barnett (1994). Similarly, identification of the subset of regressors may be accomplished by several approaches like sequential search approaches, backward elimination, etc. (Johnson and Wichern, 1988; Hair, et al., 1995). The choice of backpropagation algorithm is also arbitrary and may be substituted by other networks such as radial basis function networks. In the remainder of this chapter, the three-step procedure is used to generate a correlation for a petroleum refinery based on real data.

4.3 An Example from Petroleum Refinery

In its native state, crude petroleum consists of a large number of hydrocarbons in addition to small quantities of inorganic compounds. The purpose of the refining process is the production of marketable products from crude. Fractionation towers and other processing equipment are employed in petroleum refineries. The separation of the crude into fractions possessing different properties leverages boiling range differences between desired products. Figure 4.3 depicts a typical fractionation tower in a refinery. Also shown are typical product streams drawn from the tower. The main feed comprises of preflash bottoms from the crude furnace. An auxiliary feed, preflash kerosene from the preflash tower, also enters the tower. A number of products are obtained. Lighter

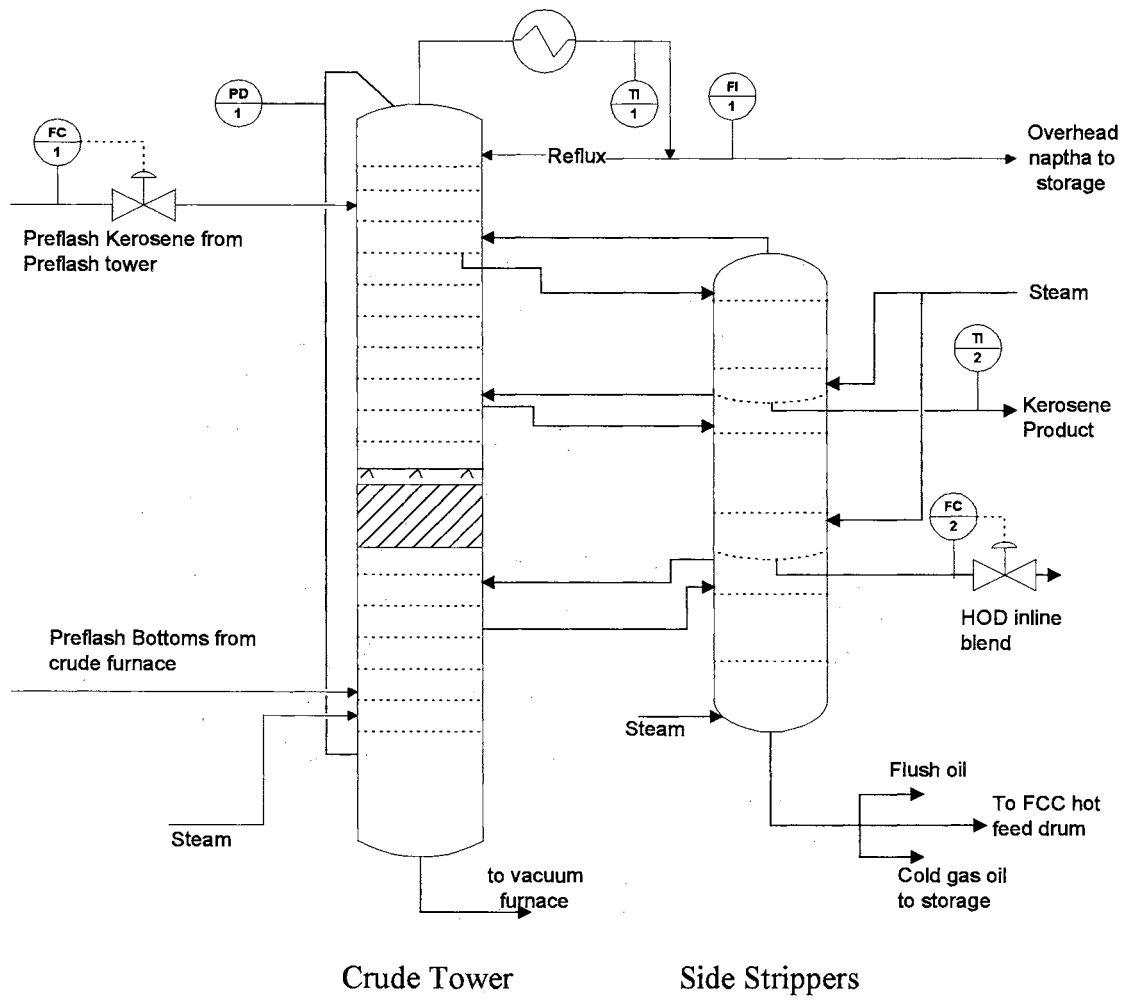


Figure 4.3: A schematic of a typical atmospheric distillation tower. Shown are two feed streams that enter the tower. Side streams are often fed to side-strippers to obtain intermediate products between the overhead product and bottoms.

products like naphtha are withdrawn from the top the tower while heavier products are collected at the bottom. Side-drawn products are often fed to stripping columns for further separation.

The temperature profile and the vapor traffic within the column are manipulated by using pumparound reflux. The large quantities of material and energy flow along with complex thermodynamic behavior of the components involved make the operation and control of the tower difficult. For profitable operation of the column, control of product quality is most important. In conventional distillation, product quality is expressed in terms of purity of components involved. The component purity may then be related to boiling points, which are used as control specifications. However, in crude petroleum fractionation, the products usually consist of a variety of components and a component by component analysis is not practically realizable. Instead, the products are subjected to rapid distillation procedures defined by American Society of Testing Materials (ASTM) during routine laboratory tests to yield an approximate measure of the composition in terms of ASTM distillation boiling ranges. As noted by Nelson (1941), routine tests to estimate the endpoints of the product form a “common basis of understanding between the refiner and the business world.”

A measure often used is the ASTM 95% endpoint of the product and refers to the temperature when 95% of the product is vaporized when distilled using the ASTM procedures. For example, kerosene has an ASTM boiling range of 325 °F to 550 °F. The customer may specify the desired kerosene ASTM 95% endpoint to be, say, 525 °F. In

this case, the column will need to be controlled such that the product quality specification is met. However, unlike conventional distillation, where the composition of the components in product streams may be measured reliably by on-stream analyzers, in petroleum fractionation, an online measurement of ASTM 95% endpoint is not generally feasible. The product sample is sent for routine laboratory tests to determine quality. Large lags of four to eight hours may occur before results of the lab analysis are known. During this time, a large amount of off-spec material may be produced leading to significant economic penalty. The potential for savings via more accurate product quality estimates forms the prime motivation for development of reliable inferential measurements of product quality. Quality control implementation on a real refinery using inferential models is discussed by San et al (1994a; 1994b). They also discuss the economic benefits realized from such a scheme.

In the next three sections, the three-step procedure of section 4.2 will be used to develop an empirical correlation to infer ASTM 95% endpoint of kerosene. Conventional lab methods to measure ASTM 95% endpoints use ASTM distillation apparatus and often yield varying results for the same sample and consequently have limited reliability. On the other hand measurements using simulated distillation (SimDist®) techniques exhibit reduced variability in measurements of the ASTM 95% endpoint of a given sample. SimDist is a liquid chromatographic procedure, which generates endpoint measurements with greater accuracy than traditional ASTM test procedures. However, both techniques have significant measurement lags associated with them making them unsuitable for use by the multivariable control system. Thus,

inference of ASTM 95% product endpoint is essential to enable control of the fractionation tower.

4.4 Identification and Collection of Candidate Independent Variables and Preprocessing of Data

The development of inferential measurement of ASTM 95% endpoint of kerosene begins with collecting the data required to construct a correlation. However, before collection of the data, the measurements that will be included in the data set must be decided. This decision is usually based on process insights and operational experience. Some factors considered in deciding the set of candidate independent variables for the fractionation tower example are discussed below.

4.4.1 Rationale for deciding set of candidate independent variables

The degree of separation between adjacent streams in a fractionation tower is strongly influenced by the internal operating conditions of the tower. The magnitude of separation is often measured in terms of (5-95) gap/overlap and is defined as the difference between the ASTM 5% initial point of the heavy stream and the ASTM 95% endpoint of the adjacent lighter stream (Watkins, 1979). Thus,

$$(5 - 95) \text{ Gap / Overlap} = T_{5\%, \text{ Heavy}} - T_{95\%, \text{ Light}}, \text{ ASTM} \quad (4.5)$$

Gap/overlap measures the degree of separation between two adjacent product streams. For example, an overlap of 60 °F represents poorer separation of adjacent products than an overlap of say, 10 °F. This is evident from the definition of (5-95) gap/overlap as shown in equation (4.4). An overlap of 60 °F represents a case where the ASTM 95%

endpoint of the lighter product stream is greater than the ASTM 5% initial point of the heavier stream by 60 °F. Thus, the ASTM boiling ranges of the two product streams overlap implying a poor separation between the two.

It is possible to estimate the ASTM 95% endpoint of a product based on product gap/overlap and true boiling point (TBP) cut point between the product stream and the adjacent heavier stream. As an example, consider the light product stream, naphtha, and the adjacent heavier product stream, kerosene. Based on material balances, the ASTM 95% endpoint of naphtha is evaluated as,

$$\text{ASTM 95\% endpoint Naphtha} = - \frac{\% \text{Kerosene}}{\% \text{Kerosene} + \% \text{Naphtha}} \text{gap/overlap} + \text{cut point} \quad (4.6)$$

where the yields of products, kerosene and naphtha, are defined as:

$$\text{product yield} = \frac{\text{product draw rate}}{\text{feed rate}} \quad (4.7)$$

The cut point used in equation (4.6) represents the whole crude TBP temperature corresponding to the yield point between two fractions. In the above example, the TBP cut point between naphtha and kerosene is calculated as follows:

$$\text{Cut point} = \text{Kerosene}_{50\%, \text{ASTM}} - \left(\frac{\% \text{Naphtha}}{\% \text{Naphtha} + \% \text{Kerosene}} \right) \times (\text{Kerosene}_{50\%, \text{ASTM}} - \text{Naphtha}_{50\%, \text{ASTM}}) \quad (4.8)$$

Based on equations (4.6), (4.7) and (4.8), it is proposed to infer the ASTM 95% endpoint of products as follows:

$$\text{ASTM 95\% endpoint} = g(\text{gap / overlap, product yields, ASTM 50\% points, operating variables}) \quad (4.9)$$

However, gap/overlap and ASTM 50% product points are lab measurements and not available online. It is decided to predict the value of gap/overlap so that it could be used to infer the ASTM 95% endpoint in accordance to equation (4.9). Empirical knowledge exists that relates the degree of separation as measured by gap/overlap to the separation capability of the system represented by an F-factor and the degree of difficulty of separation, $\Delta T(50\%)$ (Watkins, 1979). Thus, the following functional dependence is suggested:

$$\text{gap/overlap} = f_1(F - \text{factor}, \Delta T(50\%)) \quad (4.10)$$

However, F-factor is related to the internal reflux ratio of the tower while the parameter, $\Delta T(50\%)$, is a function of the ASTM 50% temperatures of the product streams. Thus, equation (4.10) can be reformulated as:

$$\text{gap/overlap} = f_2(\text{internal reflux ratio}, \text{ASTM 50\% point}) \quad (4.11)$$

Although, the internal reflux ratio cannot be measured explicitly, it is influenced by the operating conditions of the fractionation tower. Hence, gap/overlap is related to operating variables and ASTM 50% temperature ,

$$\text{gap/overlap} = f(\text{operating pressures, temperatures and flow rates}, \text{ASTM 50\% point}) \quad (4.12)$$

Equations (4.9) and (4.12) suggest a list of candidate independent variables. We intend to first approximate function f by a neural network NN1 to predict gap/overlap and subsequently use the predicted value as one of the inputs to network NN2 which will approximate function g to predict the desired ASTM 95% endpoint. Figure 4.4 shows a schematic description of this configuration.

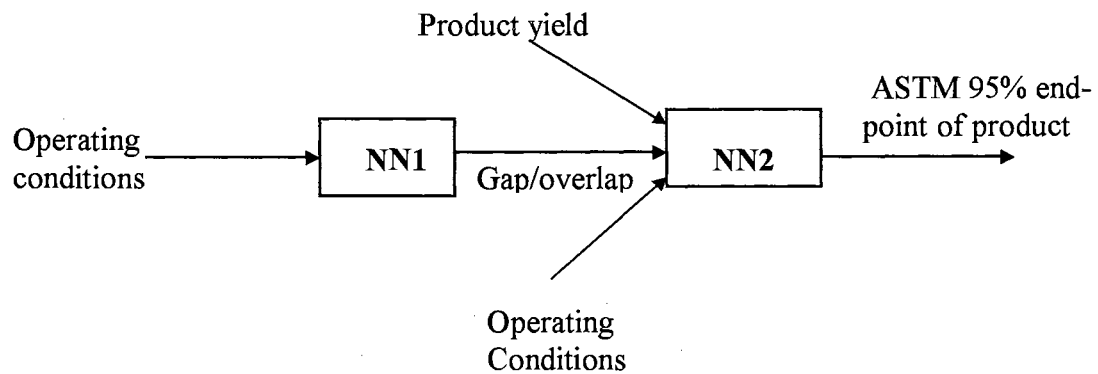


Figure 4.4: Illustration of regression scheme. Neural network NN1 predicts gap/overlap. Network NN2 uses the predicted gap/overlap as one of its inputs among others to estimate the unmeasured variable, ASTM 95% endpoint of kerosene.

4.4.2 Collection of Data

The models developed in the current work correspond to steady state operation of the fractionation tower. Thus, averaging of process variables was employed to remove the effect of local transients on the data set. Measurements of process variables like pressure, temperature and flow were hourly averages. For example, consider a product sample collected at 4:00 p.m. for measurement of ASTM distillation temperature. Then the corresponding observations of process variables at 4:00 p.m. are the averaged values from 3:00 p.m. to 4:00 p.m. On the other hand, the SimDist measurements on product quality were reflective of samples collected at a specific point in time. The data set was augmented by addition of variables like product yield calculated, various stream enthalpies and equipment heat duties. The data set was arranged in the style of Figure 4.1 and consisted of 59 variables and 546 observations representing the distillation unit operation for one year. Data on ASTM 95% endpoint, gap/overlap, and other quality variables were based on measurements using simulated distillation (SimDist[®]) of the product samples.

4.4.3 Data Preprocessing

Outliers were detected using the test statistics presented in equation (4.2). Those observations whose test statistic deviated by more than two standard deviations from the mean were deleted. Here, 11 outliers were detected by statistic d_{1i} , 29 by d_{2i} and 15 by d_{3i} . The total number of outliers deleted was 33. It is noted that the outliers detected by the three statistics may refer to the same observation and thus the total number of outlying observations that were deleted does not equal the sum of the outliers detected by

each statistic. After deletion of outliers, the remainder data set had 513 observations. All further analysis was based on the (513 observations x 59 variables) data matrix.

4.5 Identification of a Suitable Subset for Regression

As discussed in section 4.4.1, the estimation of ASTM 95% endpoint consists of predictions by two neural networks arranged in series as shown in Figure 4.4. Considerations on selection of inputs to these networks were discussed in 4.2.2. Despite using these guidelines, the list of candidate predictor variables was still very large (over 50 variables for both neural networks, NN1 and NN2). To keep the model simple, it is desirable to identify only those variables that significantly influence the dependent variables, viz. gap/overlap in NN1 and 95% end-point in NN2.

Scatter plots, simple and partial correlation coefficients were generated between each of the independent variable and the dependent variable. For purpose of illustration, the plot depicting gap/overlap between kerosene and the adjacent heavy oil distillate (HOD) versus pressure differential across the fractionation tower measured at the top (measurement tag PD1, in Figure 4.3) is reproduced in Figure 4.5. A visual inspection of the scatter plot and the correlation coefficient of -0.72 indicate a strong relationship between these two variables. Thus, the pressure difference across the tower is selected as a variable in the subset, which will be employed to predict kerosene/HOD gap/overlap. The identification of the variable, pressure differential in tower is reasonable since it governs vapor traffic and hence internal reflux ratio in the tower. The relationship between gap/overlap and internal reflux ratio was suggested by equation (4.11).

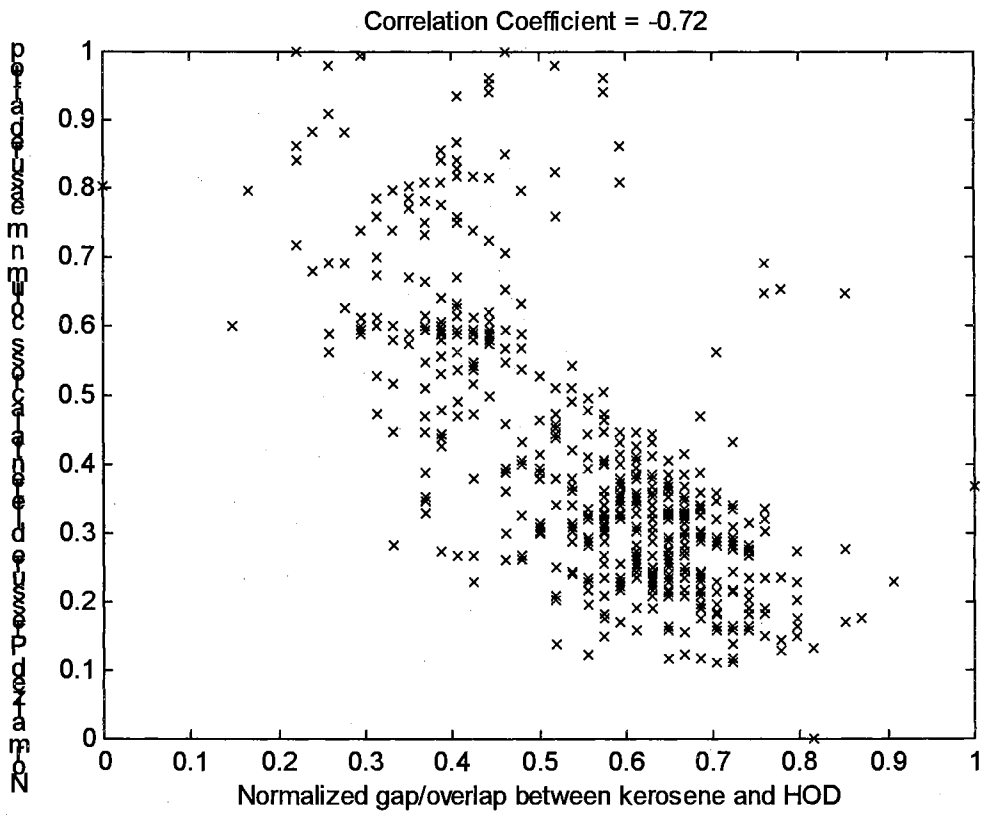


Figure 4.5: Scatter plot depicting relationship between kerosene/HOD gap/overlap and pressure differential across atmospheric tower (PD1 in Figure 4.1).

One of the other candidate independent variables in the preliminary list was the pressure differential across the tower measured at the bottom. The simple correlation coefficient between SimDist measurement of gap/overlap and the pressure difference across the tower measured at the bottom (measurement tag, PD2 in Figure 4.3) is -0.47, indicating a reasonably significant relationship between them, relative to other candidate variables. However, the partial correlation between the SimDist measurement of gap/overlap and PD2 is only -0.12, indicating that the individual influence of this measurement on gap/overlap is relatively weak. On the other hand the partial correlation coefficient between the SimDist measurement of gap/overlap and the pressure difference across the tower measured at top of tower (measurement tag: PD1) is -0.53. This situation indicates that the pressure difference measured at the top of the tower is a better indicator of the gap/overlap than the pressure difference across the tower measured at the bottom. The significant value of the simple correlation coefficient (-0.47) between PD2 and SimDist measurement of gap/overlap may be due to common effect of PD1 on both gap/overlap and PD2.

Based on scatter plots, simple and partial correlation coefficients a subset of eight variables was selected which forms a preliminary subset of variables used to approximate function f of equation (4.12) by neural network NN1. A description of the selected variables is provided in Table 4.1. A similar exercise was performed to identify inputs for neural network NN2. The scatter plot of the network output, viz. ASTM 95% temperature of kerosene, and one of the inputs, viz. gap/overlap, is shown in Figure 4.6. The correlation coefficient of absolute value 0.7 indicates a significant association

Table 4.1:

Variables influencing Kerosene/HOD gap/overlap as identified by scatter plots, simple and partial correlation coefficients.

Variable Number	Predictor Variable	Correlation Coefficient	Partial Correlation Coefficient
1	ASTM 50% cut point of kerosene	-0.48	0.09
2	differential of 50% points of adjacent light streams	-0.44	-0.27
3	differential of 50% points of adjacent heavier streams	-0.42	0.02
4	FC1, heavy product to storage	0.41	0.25
5	FC2, auxiliary feed	0.61	0.11
6	TI1, side-stripper bottoms temp.	-0.49	-0.27
7	PDI1, ΔP at top of column	-0.72	-0.53
8	PDI2, ΔP at bottom of column	-0.47	-0.12

between these two variables. This feature is consistent with the definition of (5-95) gap in equation (4.5) and was also suggested by equation (4.9). The corresponding partial correlation coefficient value is found to be -0.84 . However, as depicted in Figure 4.4, the input gap/overlap to network NN2 is a prediction by the network NN1 and not the SimDist measurement on which the current analysis is based. Thus, the quality of prediction of gap/overlap by NN1 limits the performance of the network NN2. Hence, a

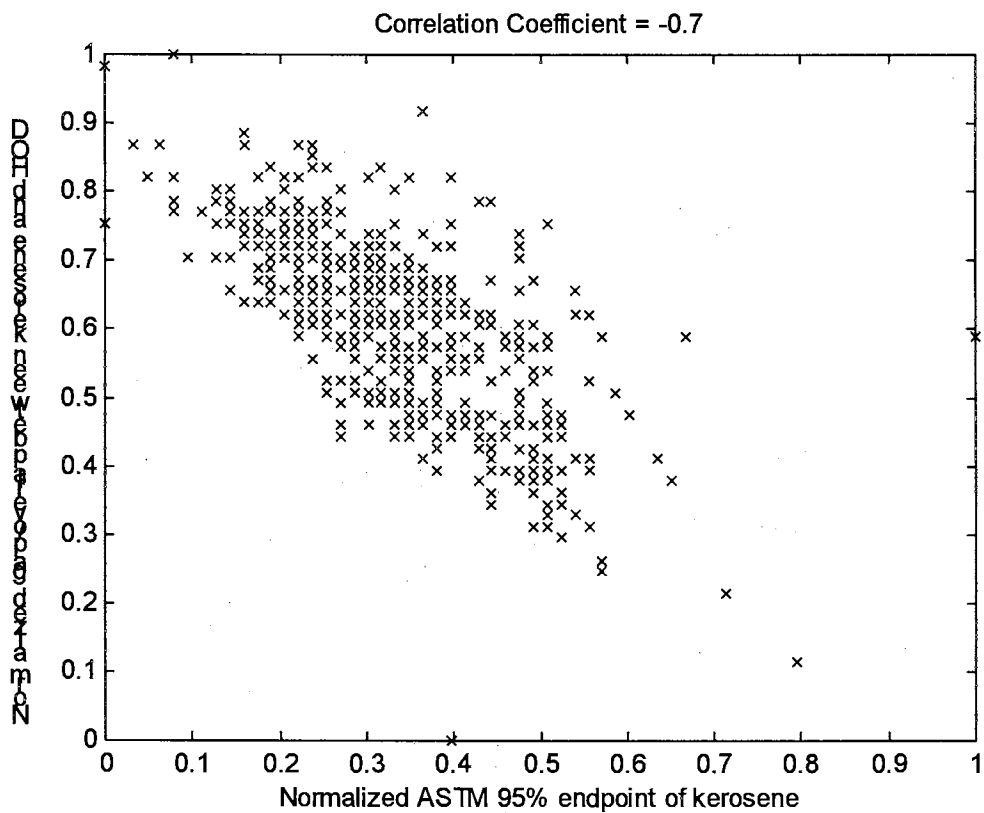


Figure 4.6: Scatter plot depicting relationship between ASTM 95% endpoint of kerosene and kerosene/HOD gap/overlap.

good prediction of gap/overlap is crucial to a good performance of the overall model. Based on scatter plots, simple and partial correlation coefficients a preliminary subset of nine independent variables is selected. See Table 4.2 for a brief description of these variables.

The preliminary subsets for NN1 (Table 4.1) and NN2 (Table 4.2) are then subjected to Mallows' C_p statistic analysis to study their suitability. The C_p statistic is calculated for all possible combinations of variables from Table 4.1. For purposes of illustration, the Mallow statistic calculated for certain subsets is displayed in Table 4.3. The variables considered in each subset in the table represent that combination corresponding to which the value of C_p is lowest for a fixed number of variables, p , in the model. It is further observed that if we plot the p v/s C_p graph, none of the points would lie on or close to the 45° line, indicating that none of these subsets are optimal.

Although, the model with eight variables ($C_p = 17$) may be the "best" based on C_p statistic, the subset with five variables ($C_p = 24$) may be considered to be more prudent, since it involves a fewer number of variables. The model with eight predictors contains three extra measurements over the model with five predictors, viz. ASTM 50% temperature of kerosene (variable number 1 in Table 4.1), difference between ASTM 50% points of HOD and kerosene (variable number 3 in Table 4.1) and pressure difference across the tower measured at the bottom of the tower (variable number 8 in Table 4.1). The partial correlations between these three variables and gap/overlap are seen to be the lowest when compared to the other five variables implying a relatively low degree of individual influence on the dependent variable, gap/overlap. Thus, based on

Table 4.2

Variables determined to be significant influences on ASTM 95% endpoint of kerosene as identified by scatter plots, simple, and partial correlation coefficients.

Variable Number	Predictor Variable	Correlation Coefficient	Partial Correlation Coefficient
1	kerosene and HOD gap/overlap	-0.70	-0.84
2	naphtha yield	-0.43	0.18
3	kerosene yield	0.42	-0.15
4	fraction of kerosene in product	-0.48	0.18
5	50% cutpoint of kerosene	0.69	0.39
6	50% cutpoint of HOD	0.57	0.62
7	HOD 50% - KERO 50%	0.77	0.13
8	ΔP at top of crude tower	0.46	0.13
9	enthalpy of naphtha stream	-0.45	-0.45

the combined results of Mallows's statistic and partial correlation, the model with five predictors was selected for prediction of gap/overlap by the neural network NN1. This exercised was repeated with the variables in Table 4.2 to identify inputs to neural network NN2 (see Figure 4.4). The C_P values for a few subsets are shown in Table 4.4. Based on the C_P values, the subset with four variables is selected.

Table 4.3

Mallow Statistic for certain Subsets of Variables of Table 4.1.

Number of Predict Variables in Subset, p	Subset of Variables, { Variables Number }*	C _p
1	{7}	283
2	{6,7}	208
3	{4,6,7}	96
4	{2,4,6,7}	32
5	{2,4,5,6,7}	24
6	{2,4,5,6,7,8}	18
7	{2,3,4,5,6,7,8}	19
8	{1,2,3,4,5,6,7,8}	17

* *To identify the variable number with measurement description, refer to Table 4.1.*

Statistical techniques focus attention on variables that have significant prediction power on the dependent variable. However, it is useful to provide a physical significance to each of the identified variables from a process standpoint.

4.6 Regression Using Neural Networks

The iterative method of training and regression refinement discussed in section 4.2.3 was applied to neural networks NN1 and NN2. The inputs to these neural networks can be referenced from Tables 4.3 and 4.4, respectively. During the regression

Table 4.4

Mallow Statistic for certain Subsets of Variables of Table 4.2.

Number of Predict Variables in Subset, p	Subset of Variables, { Variables Number }*	C_p
1	{1}	3834
2	{1,6}	933
3	{1,6,9}	282
4	{1,5,6,9}	47
5	{1,4,5,6,9}	11

* *To identify the variable number with measurement description, refer to Table 4.2.*

refinement phase, those observations that beyond 2.5 times the standard deviation of error from the error mean were deleted. The threshold 2.5 was arrived at by trial and error. Twice the standard deviation caused too many observations to be deleted while thrice the standard deviation caused too few observations to be deleted. This procedure was iterated a few times before arriving at the final values of weights and biases. It is assumed that such outlying observations detected by the neural network trained using cross-validation approach represent unsteady state measurements, which are inappropriate for model development. Moreover, such observations may adversely affect the regression parameters. In the current example, deletion of 25 observations, used in training of network NN1, afforded a significant decrease in the standard deviation of

error from 3.9° F (before deletion of outlying observations) to 2.7° F (after deletion of outlying observations). The performance of this network is shown in Figure 4.7.

A similar procedure was applied to neural network NN2 used in prediction of ASTM 95% kerosene endpoint. Note that for evaluation of the performance of NN2, the input gap/overlap used is the output of NN1 while the SimDist measured gap/overlap is used in training. Only one iteration of training and regression refinement was used during which 11 outliers were deleted. The comparison of the predicted value of ASTM 95% endpoints versus the SimDist measurements is shown in Figure 4.8.

4.6.1 A Benchmark Test

It is instructive to compare the predictions in Figure 4.8 with neural network NN2 predictions when the input gap/overlap refers to actual SimDist measurements rather than gap/overlap values predicted by network NN1. This comparison reflects the performance of the network NN1 and sets a limit to the accuracy of the prediction of ASTM 95% endpoint for the given data set using the scheme presented in Figure 4.4. Figure 4.9 shows the performance of the network, NN2 when SimDist values of gap/overlap are used to predict the ASTM 95% endpoint. The standard deviation of the prediction errors is 1.8 °F. The standard deviation of prediction errors when predicted gap/overlap was used as input to NN2 was 2.3 °F (see Figure 4.8). The superior performance is expected since actual SimDist measurements of gap/overlap are used as inputs to NN2 and hence errors in NN1 prediction are not reflected in the performance of network NN2. However, in implementation of the work-plan, no such measurements will be available online and

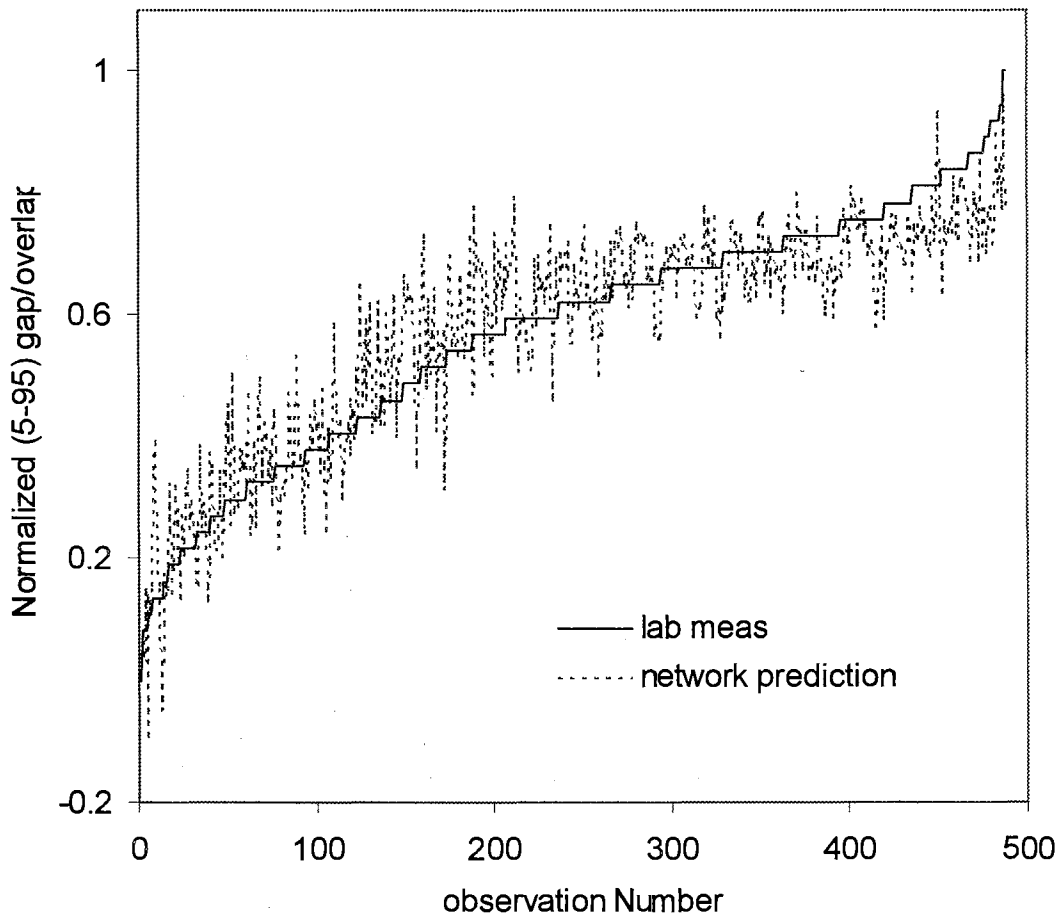


Figure 4.7: Performance of network NN1 when trained using trimmed data. The trimming of outliers was performed to refine regression (section 4.2.3.2). Five neurons are used in hidden layer.

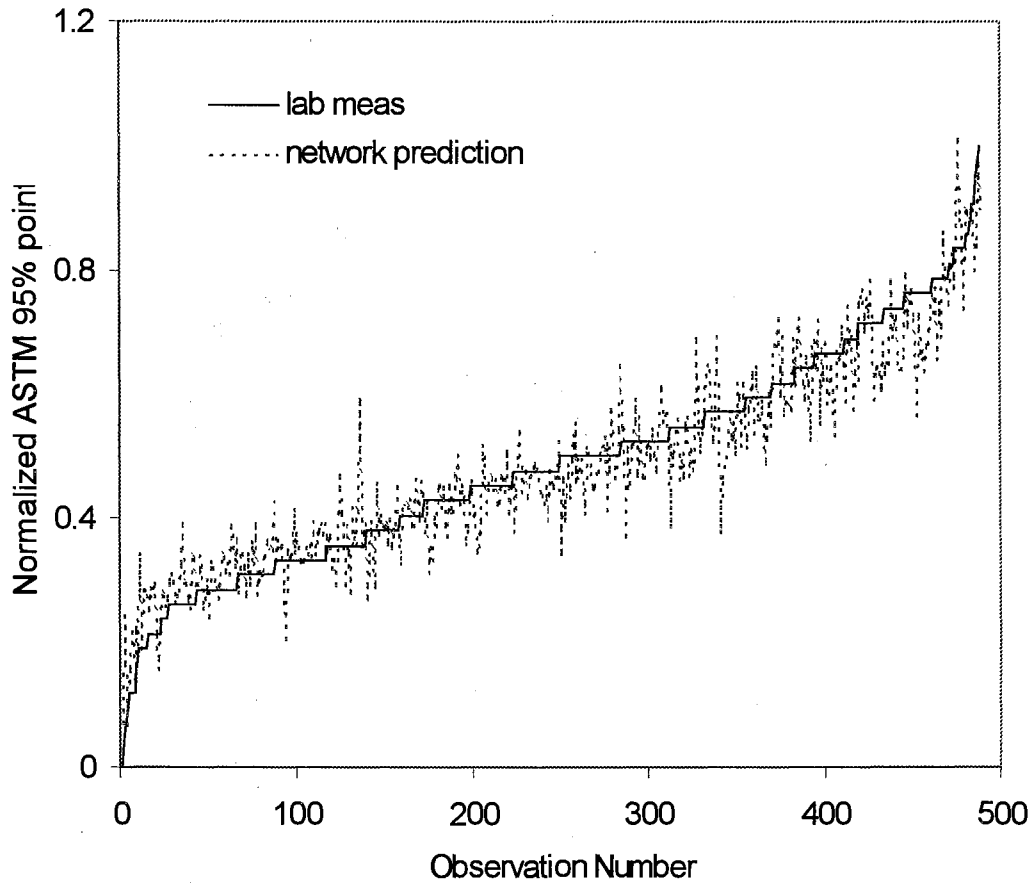


Figure 4.8: Performance of network NN2. The input gap/overlap is predicted by NN1 when it is trained using the trimmed data set (Figure 4.7). Four neurons are used in hidden layer.

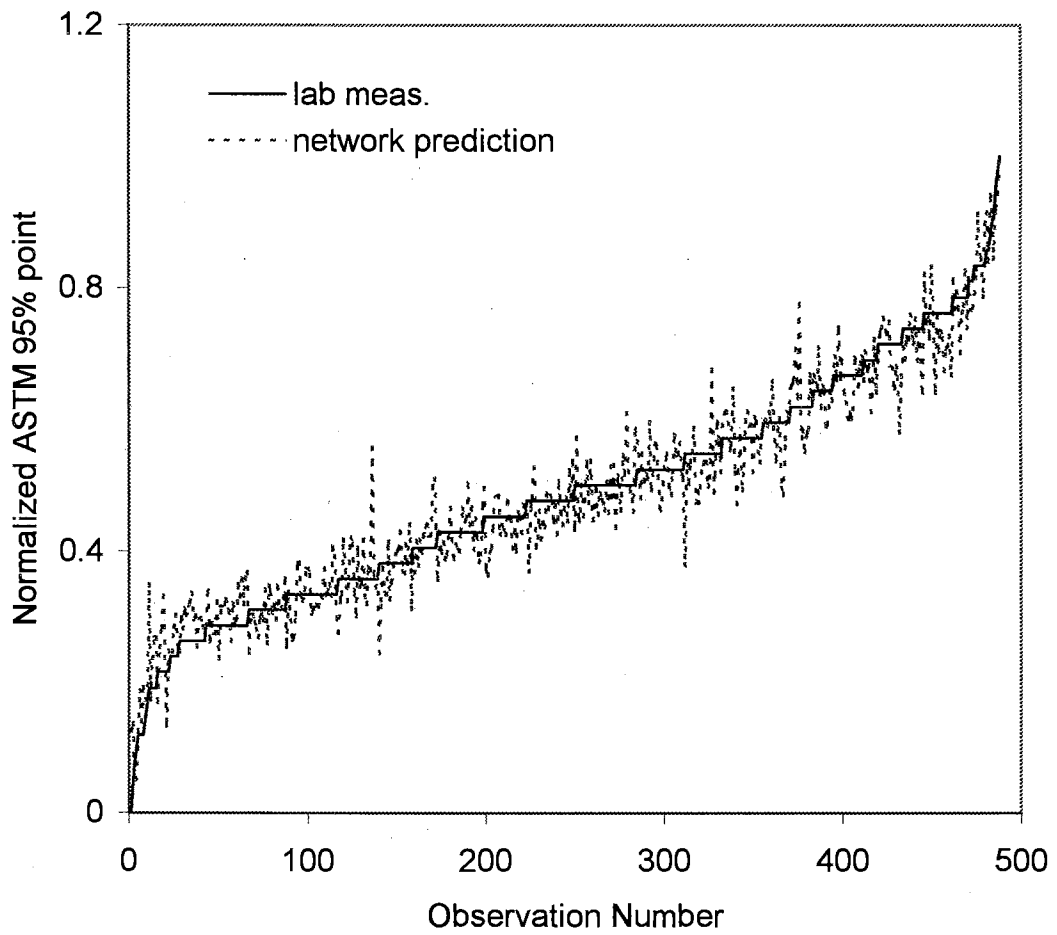


Figure 4.9: Performance of NN2 when trained by data set trimmed to refine regression. The input values of gap/overlap used to evaluate performance are the SimDist measurements and do not represent prediction by NN1 as in Figure 4.8.

will need to be predicted by NN1. This comparison merely serves as a benchmark test to determine the effect of performance of NN1 on the performance of NN2.

4.6.2 A Rearranged Scheme of Work

Analysis of the data set in section 4.5 had revealed a strong relationship between ASTM 95% endpoint and gap/overlap. This, in turn, motivated the idea of employing a neural network, NN1, to predict gap/overlap. Subsequently, this prediction along with other variables identified by sub-problem (b) would be used as inputs to a second neural network, NN2, to predict ASTM 95% endpoints. The configuration of the overall scheme is shown in Figure 4.10. Also shown is an alternative arrangement. Here, the eight inputs to the neural network, NN consist of the five NN1 inputs and three NN2 inputs (except gap/overlap).

A prime motivation for the rearranged scheme is the ease of implementation of endpoint inference technique. Although gap/overlap is not predicted explicitly as in Figure 4.4, it is expected that this information will be manifested in the five NN1 inputs and is therefore embedded in the regression scheme to predict ASTM 95% endpoint. It is noted that the rearranged scheme of work does not supplant the previous scheme, since all inputs to neural network NN were identified as inputs to NN1 and NN2 in the revised scheme. The rearrangement is merely a direct approach to ASTM 95% endpoint calculation. Thus, the rearranged scheme of work in Figure 4.11 is equivalent to the revised scheme of work in an input/output sense and only differs with respect to the internal structure of regression.

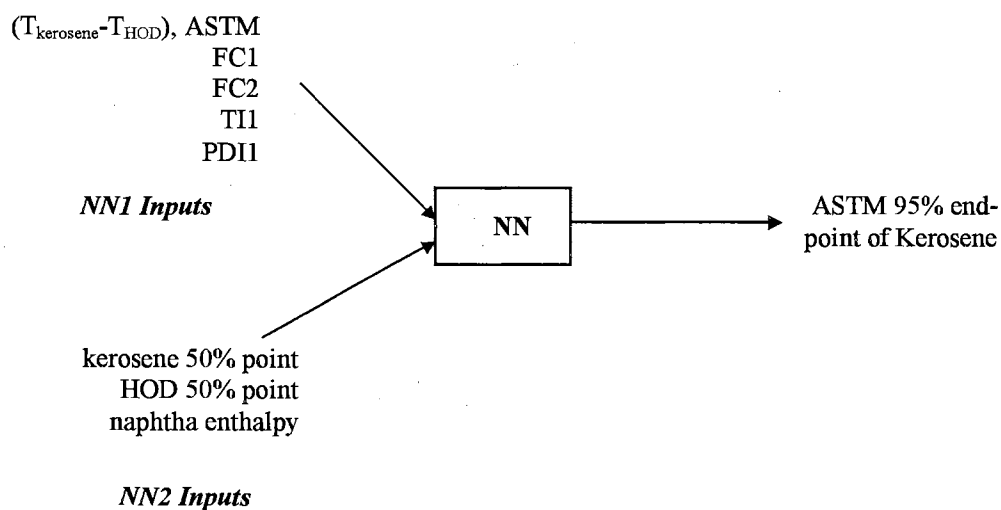
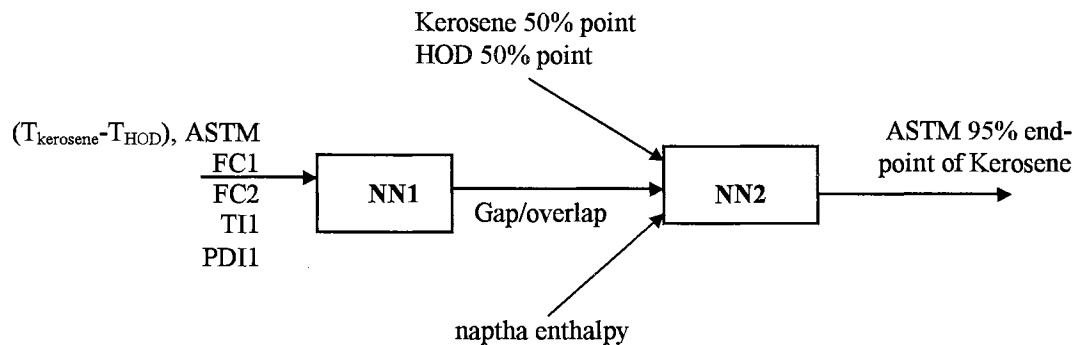


Figure 4.10: The top schematic shows the configuration of the inferential correlation in the original scheme. The inputs to the network were identified using procedures described in sub-problem (b). The bottom schematic shows the revised scheme of work with identified inputs. The inputs to NN1 and NN2 form the inputs to network NN (except the intermediate variable, gap/overlap).

The summary of performances of the rearranged network NN of figure 4.11 when 3, 4, 5, 6, and 7 neurons are employed in the hidden layer is shown in Table 4.5. Table 4.5 also shows the performance of the serial NN1/NN2 scheme. It is observed that each of the single neural network NN schemes exhibit superior performance when compared to the overall performance of the two network configuration. In fact, four of the five neural networks studied in the rearranged scheme of work, viz. when four, five, six and seven neurons are used in hidden layer, showed comparable performances relative to the predictions of NN2 in the benchmark test.

It was noted in the results of serial NN1/NN2 scheme of work that the errors in prediction of gap/overlap by NN1 adversely affected the estimation of ASTM 95% endpoint by NN2. This situation arose because the predicted gap/overlap was a direct input to NN2. The strong association between gap/overlap and ASTM 95% endpoint led to magnification of the errors in gap/overlap prediction. In the revised scheme of work no such accumulation of errors occurs since all inputs are fed directly to the network NN which predicts the ASTM 95% endpoint. This is also consistent with the fact that predictions of the rearranged scheme of work were comparable with NN2 estimates of ASTM 95% endpoint when SimDist measurements of gap/overlap are used (see section 4.6.1). It is observed from Table 4.5 that the neural network with five neurons in the hidden layer is optimal in the sense that it exhibits the smallest mean squared error (MSE). As the number of neurons increases from three to five in the hidden layer, MSE of predictions decreases. However, a further increase in hidden layer neurons causes MSE to increase. This situation may be attributed to the cross-validation method employed in training. As the number of neurons in the hidden layer increases, the neural

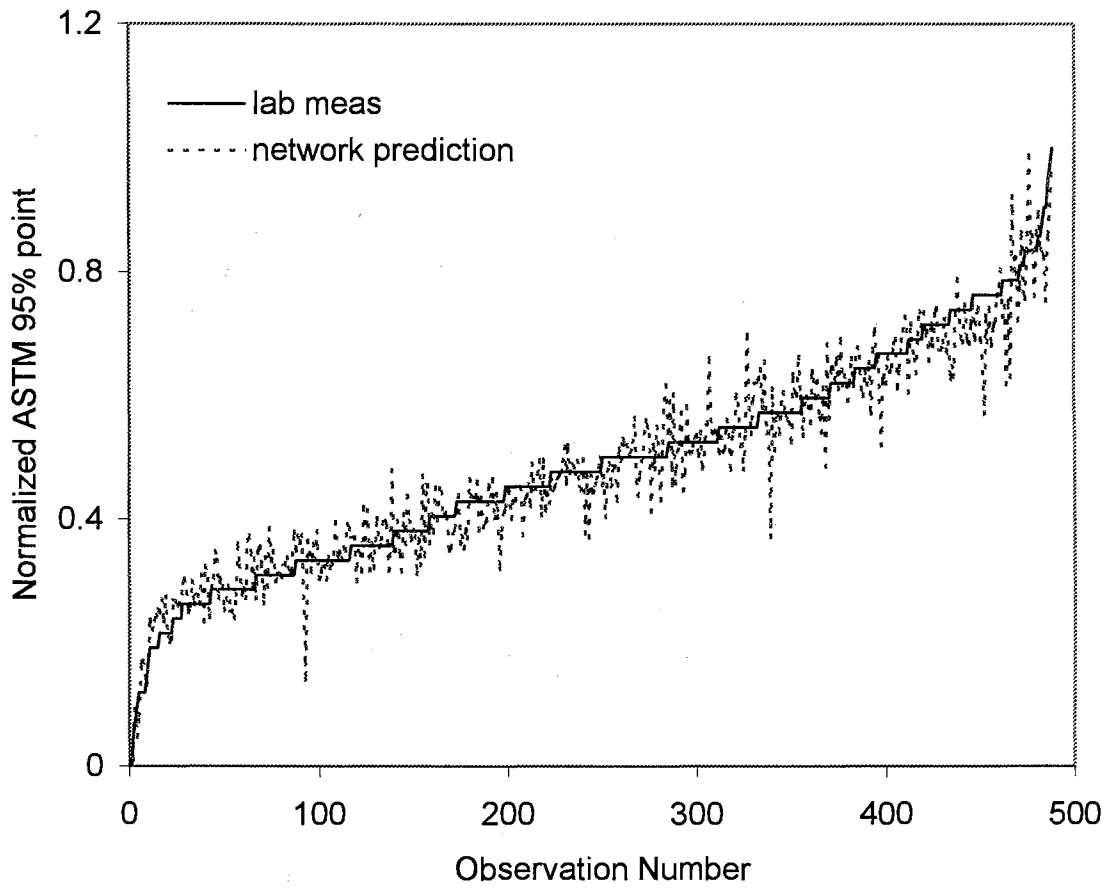


Figure 4.11: Performance of network NN when five neurons are used in hidden layer. The inputs and output are shown schematically in Figure 4.10. Performance measures are shown in Table 4.5.

Table 4.5

Performance measures for neural network NN (rearranged scheme of work) when different number of neurons are used in the hidden layer. Also the results of the revised scheme of work are shown.

Number of Neurons in hidden layer	Mean Squared Error in Prediction ($^{\circ}\text{F}^2$)	Standard Deviation of Prediction Error ($^{\circ}\text{F}$)	Minimum Error ($^{\circ}\text{F}$)	Maximum Error ($^{\circ}\text{F}$)
3	4.0	2.0	-7.2	7.8
4	3.3	1.8	-5.8	8.1
5	2.9	1.7	-5.7	4.1
6	3.1	1.8	-4.9	4.5
7	3.7	1.9	-5.4	7.7
Serial NN1/NN2 Scheme	5.6	2.3	-9.9	8.2
Benchmark Serial NN1/NN2 Test	3.3	1.8	-5.0	8.1

network possesses improved capability to mimic the training set data (due to the larger number of parameters available). This memorization of the training set causes larger prediction errors when the trained network is used with the validation set, thereby, increasing the overall mean squared error. The performance of the neural network NN with five neurons in the hidden layer is shown in Figure 4.11.

4.7 Conclusions

A unified framework to develop inferential measurements is presented. The three-step procedure is used to develop a correlation to infer ASTM 95% temperature of kerosene in a fractionation tower. The developed correlation predicts the ASTM 95% endpoint of kerosene with an error standard deviation of 1.7 °F. Success of the correlation development depends on the success of each of the three steps.

In the petroleum refinery example, it is shown that the identification of candidate independent variables demands an understanding of the process. Identification of the set of variables that will be employed in the regression scheme involves a certain degree of subjective judgment and process experience. The example also illustrates the possibility of improving estimates by rearranging the correlation.

5 A FACTORIZED APPROACH TO NONLINEAR MPC USING A RADIAL BASIS FUNCTION PROCESS MODEL

Chapter Overview

A computationally efficient approach for nonlinear model predictive control (NMPC) is presented. The new approach exploits the factorability of radial basis function (RBF) process models in a traditional model predictive control (MPC) framework. The key to the approach is formulation of the RBF process model in a manner capable of making nonlinear predictions across a p -step horizon without use of future unknown process measurements. The proposed RBF model avoids error propagation from use of model predictions as input in a recursive or iterative manner.

The resulting NMPC formulation using the RBF model provides analytic expressions for the gradient and Hessian of the controller's objective function in terms of the RBF network parameters. Solution of the NMPC optimization problem is significantly simplified by factorization of the RBF model output into terms containing only known and unknown parts of the process. The proposed NMPC approach is illustrated with simulation examples.

5.1 Introduction

Many chemical processes exhibit nonlinear behavior. Application of commercial linear model predictive control (MPC) technology is only partially successful in such cases. As process conditions deviate from the nominal operating point, model mismatch increases with a corresponding degradation in control performance. The problem is particularly severe in the process industries where the areas of a plant with the greatest economic incentives to apply MPC (e.g., a reactor system) typically exhibit strong nonlinearities. Plant operators frequently disable an MPC system when model mismatch compromises overall control performance. Recommissioning cannot be performed until the MPC models are updated or the operating conditions return to original design point.

The notion of using a nonlinear model to control a significantly nonlinear process within the model predictive control paradigm has led to an active interest in the development and application of nonlinear model predictive control. Nonlinear MPC (NMPC) adheres to the general MPC philosophy, that is, use of an explicit model to predict the process behavior over a future horizon, and implementation of control action that steers the process towards predetermined objectives in an optimal sense. NMPC uses a nonlinear model to provide a better approximation of the underlying nonlinear system. However, use of a nonlinear model presents additional challenges relative to linear MPC: 1) the complexity of nonlinear systems makes systematic development of nonlinear system identification techniques difficult (Pearson and Ogunnaike, 1997), and 2) nonlinear MPC requires solution of a nonlinear program at each sampling instant making implementation more involved (Henson, 1998).

Use of artificial neural networks as nonlinear dynamic models has been studied in recent years. When applied for predictive control most utilize a feedforward network architecture, while a few use the recurrent type. Hussain (1999) provides a summary of a number of applications reported in literature.

In this chapter, we propose a radial basis function (RBF) network based NMPC approach. The proposed RBF model is capable of providing non-iterative sequential predictions over a prediction horizon of length p . The factorability of Gaussian functions, employed by the RBF network nodes, is leveraged to formulate a compact representation of the model predictions over the prediction horizon. The traditional MPC controller objective function and the associated gradient and Hessian are then directly parameterized in terms of the network parameters. The resulting NMPC system is computationally efficient and provides the enhanced control expected from use of a nonlinear model. Simulation examples are provided to demonstrate identification and control with the proposed technique.

5.2 Nonlinear System Identification Using Neural Networks

A feedforward network can be regarded as a nonlinear autoregressive model with external inputs (NARX),

$$\hat{y}_k = F(y_{k-1}, \dots, y_{k-N_y}, u_{k-1}, \dots, u_{k-N_u}) \quad (5.1)$$

where a time-delay of unity is assumed between the model output, \hat{y}_k , and the previous process inputs, u_{k-1} . N_y and N_u are integers defined by the order of the model. Scalars

$u_{k-1}, \dots, u_{k-N_u}$ represent the sequence of inputs used by the model. Function F depends on the network architecture and the type of activation function employed by the nodes.

A feedforward network is typically trained to minimize the 1-step ahead prediction error,

$$E_{FFN} = \sum_{k=1}^N |\hat{y}_{k/k-1} - y_k|^2 \quad (5.2)$$

The subscript of $\hat{y}_{k/k-1}$ emphasizes the fact that the model prediction, \hat{y}_k , at sample k is based on measurements up to and including $k-1$. However, one of the primary purposes of MPC is to deal with complex dynamics over an extended horizon. Thus, an MPC model must predict the process dynamics over a prediction horizon, p , usually greater than unity. Equation (5.1) cannot be directly employed to provide the desired long-term predictions since future measurements needed in the computation are not available. However, a feedforward network can be cascaded to itself so that the model outputs are used as inputs for future predictions. Thus, the p predictions can be obtained as follows,

$$\begin{aligned} \hat{y}_{k+1/k} &= F(y_k, \dots, y_{k+1-N_y}, u_k, \dots, u_{k+1-N_u}) \\ \hat{y}_{k+2/k} &= F(\hat{y}_{k+1/k}, y_k, \dots, y_{k+2-N_y}, u_{k+1}, \dots, u_{k+2-N_u}) \\ &\vdots \\ \hat{y}_{k+p/k} &= F(\hat{y}_{k+p-1/k}, \dots, \hat{y}_{k+p-N_y/k}, u_{k+p-1}, \dots, u_{k+p-N_u}) \end{aligned} \quad (5.3)$$

In writing the above predictions, it has been assumed that $p > N_y$. We will refer to equation (5.3) as "cascaded 1-step" predictions.

Su and McAvoy (1997) tested the long-range predictive capability of a "cascaded 1-step" feedforward network on a biological wastewater treatment system. The results showed that a feedforward neural network makes poor predictions long-term when

compared to a recurrent neural network. The poor performance was attributed to the fact that the feedforward network training based on equation (5.2) does not take multi-step prediction into account. Thus, accumulation of prediction errors leads to deterioration of model performance as the prediction horizon increases.

In contrast a recurrent neural network is trained based on minimization of the following criterion,

$$E_{RNN} = \sum_{k=1}^p |\hat{y}_{k/0} - y_k|^2 \quad (5.4)$$

The training criterion simultaneously minimizes the prediction errors for 1-step, 2-step, and so forth up to p -steps in the future. Su and McAvoy reported good results using the recurrent network on the wastewater treatment plant. While appealing for use in an MPC system, recurrent networks are extremely difficult to train (Narendra and Parthasarathy, 1990). Until this problem is overcome, recurrent networks cannot be considered for general-purpose use in an MPC system.

In the remainder of this section, we discuss an alternative approach where predictions up to p -steps in the future can be made without requiring future (and yet unknown) process outputs. Thus, no cascading is necessary to provide the future p predictions and problems with accumulation and propagation of modeling errors are avoided. The proposed approach retains the simplicity of 1-step ahead training.

The cascaded 1-step model in equation (5.3) uses model predictions between $k+1$ and $k+j$ to predict future process outputs $k+j+i$. The dummy indices, j and i , assume

values in the range, $[2, p-1]$ and $[1, p-j]$, respectively. However, we want to avoid dependency of model predictions later in the control horizon p on previous model predictions. In a real-time control setting, measurements are available only up to the current instant, k . Thus, we require measurements input to our process model be limited to instant k or earlier. This may be accomplished by starting with the input-output model of equation (5.1) and applying successive iterations of this map until the measurements needed in the model input refer to available measurements.

To illustrate this idea, consider a model with a prediction horizon, $p = 3$, output order, $N_y = 2$, and input order, $N_u = 2$. Then equation (5.1) can be rewritten as,

$$\hat{y}_k = F(y_{k-1}, y_{k-2}, u_{k-1}, u_{k-2}) \quad (5.5)$$

Applying successive iterations of function F yields the following expressions,

$$\begin{aligned} \hat{y}_k &= F(F(y_{k-2}, y_{k-3}, u_{k-2}, u_{k-3}), F(y_{k-3}, y_{k-4}, u_{k-3}, u_{k-4}), u_{k-1}, u_{k-2}) \\ \hat{y}_k &= F(F(F(y_{k-3}, y_{k-4}, u_{k-3}, u_{k-4}), y_{k-3}, u_{k-2}, u_{k-3}), F(y_{k-3}, y_{k-4}, u_{k-3}, u_{k-4}), u_{k-1}, u_{k-2}) \end{aligned} \quad (5.6)$$

Equation (5.6) may be written as,

$$\hat{y}_{k/k-3} = \bar{F}(y_{k-3}, y_{k-4}, u_{k-1}, u_{k-2}, u_{k-3}, u_{k-4}) \quad (5.7)$$

The argument of the resulting composite function, \bar{F} , contains delayed process outputs and process inputs ranging from $k-p$ to $k-p+1-N_y$ (that is, $k-3$ to $k-4$) and $k-1$ to $k-p+1-N_u$ (that is, $k-1$ to $k-4$), respectively. Note that the model defined by equation (5.7) can be used to make $p = 3$ future predictions without needing future plant outputs,

$$\begin{aligned} \hat{y}_{k+1/k-2} &= \bar{F}(y_{k-2}, y_{k-3}, u_k, u_{k-1}, u_{k-2}, u_{k-3}) \\ \hat{y}_{k+2/k-1} &= \bar{F}(y_{k-1}, y_{k-2}, u_{k+1}, u_k, u_{k-1}, u_{k-2}) \\ \hat{y}_{k+3/k} &= \bar{F}(y_k, y_{k-1}, u_{k+2}, u_{k+1}, u_k, u_{k-1}) \end{aligned} \quad (5.8)$$

Thus, we have eliminated the need to use predicted model outputs to obtain future predictions. During prediction of \hat{y}_{k+j} in equation (5.8), factors that affect the process between the time $k+j-p$ and $k+j$ are accounted for by process inputs calculated for the interval $[k+j-1, k+j-p-N_u+1]$. Thus, the model in equation (5.7) maintains the causal relationship between the inputs and output. For the general case of a prediction horizon of p samples and input and output orders of N_u and N_y , respectively, equation (5.7) takes the form,

$$\hat{y}_{k/k-p} = \bar{F}(y_{k-p}, \dots, y_{k-p+1-N_y}, u_{k-1}, \dots, u_{k-p+1-N_u}) \quad (5.9)$$

We modify the above model to make it better suited for control by replacing the past p process inputs, u_{k-1}, \dots, u_{k-p} by the corresponding control moves, $\Delta u_{k-1}, \dots, \Delta u_{k-p}$, where,

$$\Delta u_k = u_k - u_{k-1} \quad (5.10)$$

Revisiting the model in equation (5.7), the inputs are then modified as follows,

$$\begin{aligned} \hat{y}_{k/k-3} &= \bar{F}(y_{k-3}, y_{k-4}, \Delta u_{k-1} + u_{k-2}, \Delta u_{k-2} + u_{k-3}, \Delta u_{k-3} + u_{k-4}, \Delta u_{k-3} + u_{k-4}, u_{k-4}) \\ \hat{y}_{k/k-3} &= \bar{G}(y_{k-3}, y_{k-4}, \Delta u_{k-1}, \Delta u_{k-2}, \Delta u_{k-3}, u_{k-4}) \end{aligned} \quad (5.11)$$

Thus for the general model in equation (5.9),

$$\hat{y}_{k/k-p} = \bar{G}(y_{k-p}, \dots, y_{k-p+1-N_y}, u_{k-p-1}, \dots, u_{k-p+1-N_u}, \Delta u_{k-p}, \dots, \Delta u_{k-1}) \quad (5.12)$$

Replacement of recent p control inputs by the corresponding control moves simplifies expressions presented later in the chapter. From a process response approximation point of view, the delayed input/output measurements, viz. $y_{k-p}, \dots, y_{k-p+1-N_y}$ and $u_{k-p-1}, \dots, u_{k-p-N_u}$, respectively, provide a reference to the state of the system p

samples in the past. Causality is provided by the most recent p input moves $\Delta u_{k-p}, \dots, \Delta u_{k-1}$. To ensure that delayed process inputs appear in the model, in addition to input moves and process outputs, $N_u \geq 2$. Since p future predictions can be made without use of model outputs, we will refer to equation (5.12) as the " p -step control model." The p future predictions with the p -step control model can be expressed as,

$$\begin{aligned} \hat{y}_{k+1/k-p+1} &= \overline{G}(y_{k+1-p}, \dots, y_{k+2-p-N_y}, u_{k-p}, \dots, u_{k+2-p-N_u}, \Delta u_{k+1-p}, \dots, \Delta u_k) \\ &\vdots \\ \hat{y}_{k+p/k} &= \overline{G}(y_k, \dots, y_{k+1-N_y}, u_{k-1}, \dots, u_{k+1-N_u}, \Delta u_k, \dots, \Delta u_{k+p-1}) \end{aligned} \quad (5.13)$$

To graphically illustrate the arrangement of the model inputs, consider a model with a prediction horizon, $p = 4$. Let us also assume that $N_y = 3$ and $N_u = 2$. Then the inputs to the networks will be as shown in Fig. 5.1. Note that the future four predictions require only current and past information on process inputs and outputs. As discussed later in Section 5.4, the unknown future input moves are calculated by the MPC controller. In the next section, we use the p -step control model formulated as a radial basis function network to predict \hat{y}_{k+i} . An example is provided that compares predictive performance of the p -step control model with the cascaded 1-step model in equation (5.3).

5.3 Dynamic Modeling Using RBF network

Feedforward RBF networks have been widely used as models of dynamic processes (Chen, et al., 1990; Pottmann and Seborg, 1997). The dynamics are

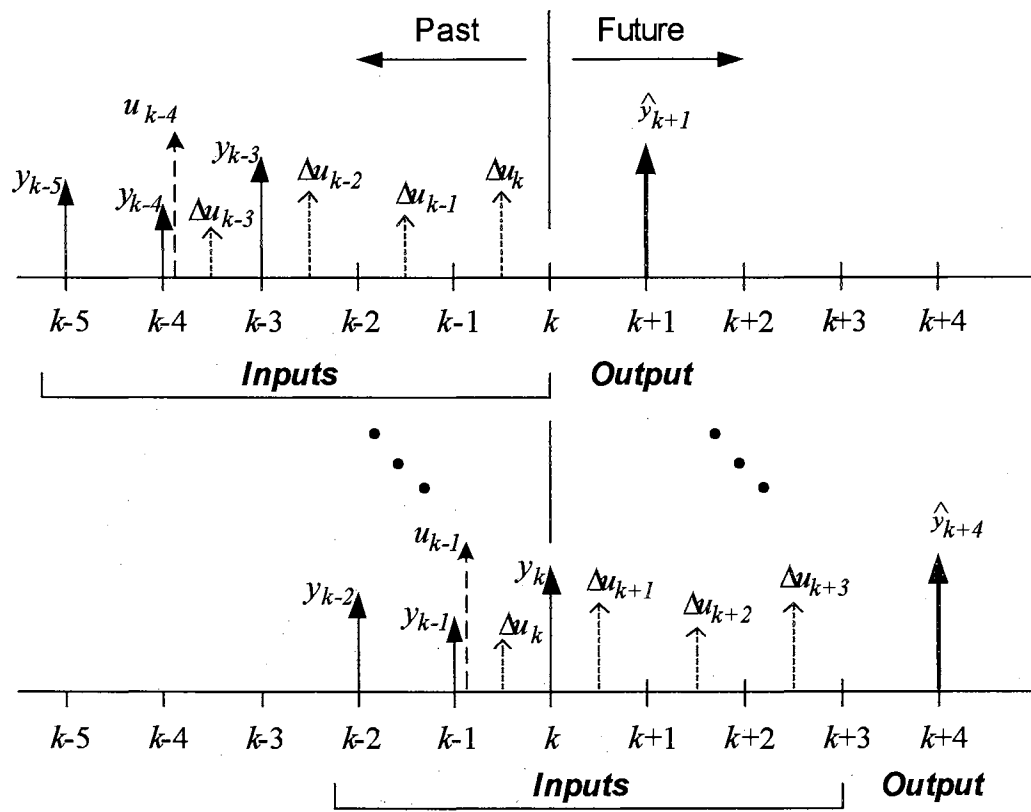


Figure 5.1: Timelines showing inputs to the p -step control model. For this example, $p = 4$, $N_y = 3$, and $N_u = 2$. Predicted outputs are generated from known information only, previous model predictions for y are not used in model input.

approximated by using past process input/output information in the network input. The RBF network is trained based on minimizing prediction error over the training trajectory (Moody and Darken, 1989). An important task in empirical model building is the selection of inputs.

Based on the p -step model for a single-input, single-output system, the input vector for an RBF network would be,

$$\mathbf{x}_k = [y_{k-p} \quad \dots \quad y_{k-p+1-N_y} \quad u_{k-p-1} \quad \dots \quad u_{k-p+1-N_u} \quad \Delta u_{k-p} \quad \dots \quad \Delta u_{k-1}] \quad (5.14)$$

Measurable disturbances can be accounted by augmenting the input vector \mathbf{x}_k to reflect current and past values of the disturbance variables. The RBF prediction of the process output at instant k is:

$$\hat{y}_k = \sum_{j=1}^m w_j \exp\left(-\frac{\|\mathbf{x}_k - \mathbf{t}_j\|^2}{\sigma^2}\right) \quad (5.15)$$

where m_1 represents the number of nodes, \mathbf{t}_j and w_j are the center and weight associated with the j^{th} node respectively. It is assumed that each node is of fixed width σ . Multiple-inputs and multiple outputs can be handled by including these in the RBF input vector, \mathbf{x}_k .

The RBF network parameters are determined through a two-step training procedure. First, the center locations, \mathbf{t}_j , and width, σ , are fixed by an unsupervised training algorithm, i.e. a clustering algorithm (Hush and Horne, 1993) (e.g. Kohonen feature map, k -means). The weights are obtained by regression with some form of regularization incorporated (Poggio and Girosi, 1990; German, et al., 1992).

To compare future predictions by the p -step control model with the cascaded 1-step model of equation (5.1), we consider a simulation example of a hot/cold water mixing process (Rhinehart, 1998). The process is shown schematically in Figure 5.2. Water at 80°C and 10°C enters through the hot and cold legs respectively. The mixed stream temperature, T_m , and flowrate, F_m , are controlled by the hot and cold leg control valves. The valves are regulated by control signals u_1 and u_2 whose range is 0-100%. The process is simulated by a first principles model that describes the flow dynamics in response to valve stem positions. The flowrate through each valve is a nonlinear function of the stem position. The temperature sensor is assumed to be located at the mixing point and is modeled as a third order response to the true mixing point temperature. The true mixing point temperature is calculated as a flowrate weighted average of the hot and cold leg temperatures. To emphasize the nonlinearity of the model, we choose to predict T_m and F_m rather than F_{hot} and F_{cold} , using signals u_1 and u_2 where:

$$T_m = \frac{F_{hot}(u_1, u_2)T_{hot} + F_{cold}(u_1, u_2)T_{cold}}{F_{hot}(u_1, u_2) + F_{cold}(u_1, u_2)} \quad (5.16)$$

and

$$F_m = F_{hot}(u_1, u_2) + F_{cold}(u_1, u_2) \quad (5.17)$$

A summary of the network configuration for the cascaded 1-step and the p -step models is provided in Table 5.1. The training and test set data were generated by exciting the process with random values of inputs u_1 and u_2 between 10% and 100% and holding the inputs for a random period between 5 and 60 seconds. The sample period was 5 seconds. In formulating the RBF input vector for the p -step control model, the prediction

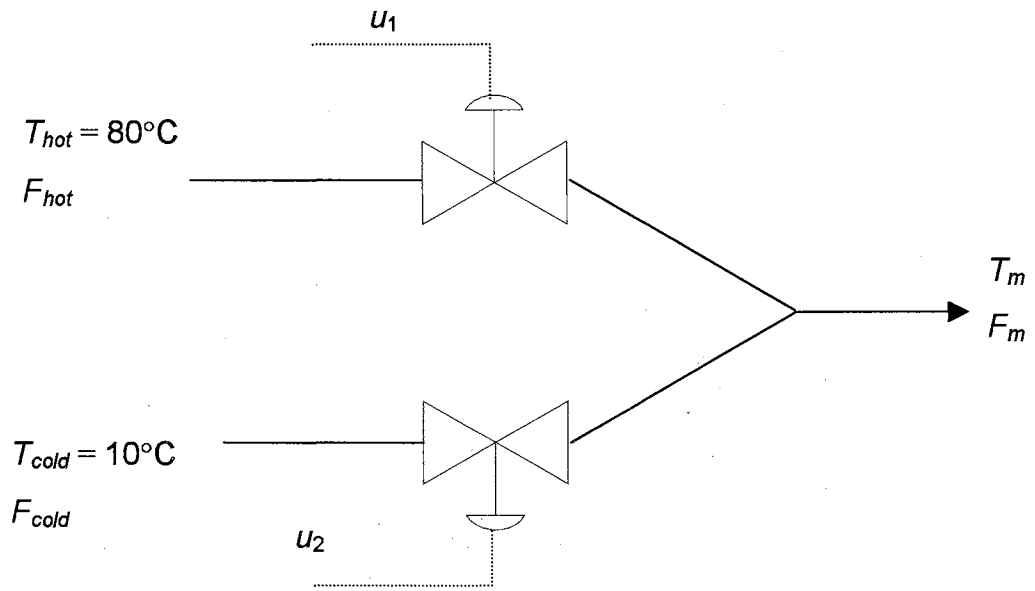


Figure 5.2: Schematic diagram of the hot/cold mixing process.

Table 5.1

RBF model summary for hot/cold mixing example.

	cascaded 1-step model ($p=1$)	p-step model ($p=8$)
Training set/Test set size	3000/2500	3000/2500
Number of hidden nodes	15	75
Number of input units	9	25
Model	N_y : 2 for T_m ; 1 for F_m	N_y : 2 for T_m ; 1 for F_m
	N_u : 1 for u_1 ; 1 for u_2	N_u : 1 for u_1 ; 1 for u_2
	N_d 1 for T_{hot} ; 1 for T_{cold}	N_d 1 for T_{hot} ; 1 for T_{cold}
Model inputs	$T_{m,k-1}, T_{m,k-2}, F_{m,k-1},$ $u_{1,k-1}, u_{2,k-1},$ $T_{hot,k-1}, T_{hot,k-2}, T_{cold,k-1}, T_{cold,k-2}$	$T_{m,k-8}, T_{m,k-9}, F_{m,k-8},$ $u_{1,k-9}, u_{2,k-9},$ $\Delta u_{1,k-8}, \dots, \Delta u_{1,k-1}, \Delta u_{2,k-8}, \dots, \Delta u_{2,k-1}$ $T_{hot,k-1}, T_{hot,k-2}, T_{cold,k-1}, T_{cold,k-2}$
Test set performance (Root mean square error)	0.8 °C for T_m 0.3 Kg/min for F_m	2.1 °C for T_m 1.1 Kg/min for F_m

horizon p was set to eight samples, the time taken by the process to reach the new steady state value. As noted in Table 5.1, the test set statistics for the 1-step model were better than those for the p -step model. This result was expected since the 1-step model had access to the previous actual output measurement, while the most recent output measurement employed by the p -step model was eight measurements in the past. As discussed below, the test set statistics are deceiving for situations where more than a single output prediction is needed.

A comparison of twenty consecutive future predictions of the cascaded 1-step and the p -step RBF models to a step change in the input u_2 from 60% to 40% is shown in Fig. 5.3. Input u_1 was held constant at 25%. The twenty future predictions by the cascaded 1-step and the p -step models were generated using equations (5.3) and (5.13) respectively. Note that the p -step control model uses only measurements available up to current instant to make future eight predictions while the cascaded model uses future predictions as

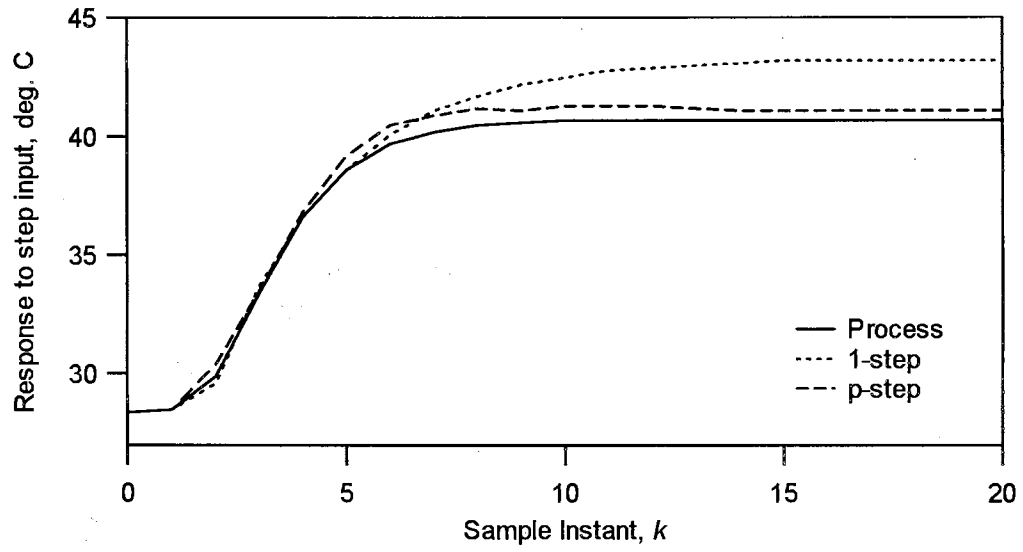


Figure 5.3: Comparison of cascaded 1-step predictions with the p -step model ($p = 8$).

inputs to the RBF network. It is evident from Figure 5.3 that the 1-step cascaded model performs well until $k = 6$. However, beyond $k = 6$ error accumulation becomes significant and the 1-step model predictions degrade. This trend is consistent with the observation made by Su and McAvoy. On the other hand, the p -step control model (with $p = 8$) performs more uniformly over the entire horizon of twenty samples. Thus, although the cascaded 1-step model provides excellent one-step predictions, long range predictions are problematic and better addressed using a p -step model. This advantage becomes more pronounced as the prediction horizon increases and would be beneficial in industrial applications of MPC which typically use large prediction horizon in the range of 20 to 50 (Marlin, 1995).

An RBF model can be easily manipulated to predict the steady-state gain at any point whenever the prediction horizon p exceeds the settling time for the process. In the absence of control moves between $k-p$ and $k-1$, the model prediction at k , \hat{y}_k , will correspond to the steady-state measurement in response to input u_{k-p-1} . Thus, the RBF model can predict steady state process output in a single computation step. Figure 5.4 illustrates the steady state prediction of mixed stream temperature by the RBF p -step model. The input to the hot leg valve was maintained at 25%, while the input to cold leg valve, u_2 , was varied from 20% to 100% in steps of 5% to generate the steady state RBF model predictions and process response. The control input moves, $\Delta u_{1,k-i}$ and $\Delta u_{2,k-i}$, $i=1, \dots, p$, were set to zero. The similarity between the model prediction and process steady state values confirms that information on nonlinear sensitivity of the process is embedded within the RBF model.

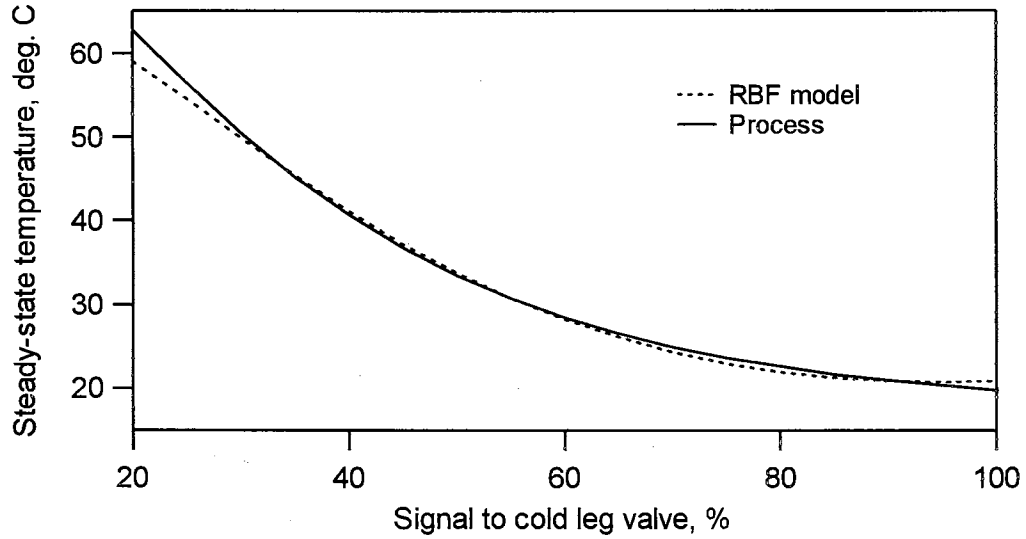


Figure 5.4: Comparison of steady state process output with RBF model predictions. Summary of RBF network is provided in Table 5.1. The signal to hot leg valve was maintained at a fixed value of 25%. The cold leg valve input was varied in steps of 5%. Each RBF prediction was obtained in one computational step.

Finally, to illustrate approximation of process dynamics by the RBF model, we present results of a simulation exercise in Figure 5.5. The figure shows the results of a step test run on the process equations and the RBF model. The cold water valve was stepped up from 50% to 75% in one simulation run and stepped down from 50% to 25% in the other. In both runs, the hot leg valve was maintained at 25%. Excellent performance is indicated in both cases. In the following section, we parameterize the MPC control problem in terms of the p -step control model.

5.4 MPC Using p -step Control Model

Model predictive control algorithms compute a manipulated variable profile over a control horizon by optimizing an objective function defined over the prediction horizon, subject to constraints. Only the first move is implemented and the procedure is repeated at every sampling instant. The optimization function reflects the process objectives that must be achieved, including minimization of overall cost. However, economic optimization is often performed by a higher level system, which determines the optimal setpoints. We utilize the traditional MPC optimization function that penalizes deviation of future model predictions, \hat{y}_{k+j} , from setpoints, r_{k+j} , while minimizing future control moves, Δu_{k+i} :

$$\phi = \sum_{i=1}^P \Gamma_i (r_{k+i} - \hat{y}_{k+i})^2 + \sum_{i=0}^{c-1} \Lambda_i (\Delta u_{k+i})^2 \quad (5.18)$$

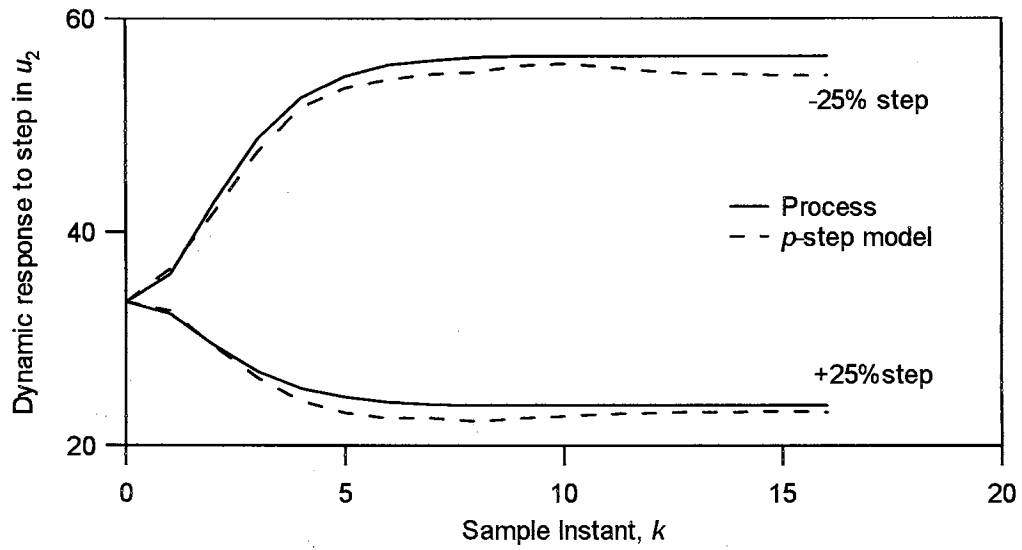


Figure 5.5: Performance of RBF model for approximation of dynamic response. The two cases shown depict response of temperature to steps in cold leg valve from 50% to 75% and from 50% to 25%. The hot leg valve was maintained at 25%.

Variables p and c represent the prediction and control horizons, respectively. Γ_i and Λ_i denote the error penalty and move suppression factors at the i^{th} instant. Then, the MPC control law can be stated as,

$$\begin{aligned} & \arg(\min_{\Delta u_k, \Delta u_{k+1}, \dots, \Delta u_{k+c-1}} \phi) \quad \text{such that} \\ & y_{\min} \leq \hat{y}_{k+i} \leq y_{\max} \\ & \Delta u_{\min} \leq \Delta u_{k+i} \leq \Delta u_{\max} \\ & u_{\min} \leq u_{k+i} \leq u_{\max} \end{aligned} \quad (5.19)$$

Based on the p -step control model, the future model predictions, \hat{y}_{k+i} , are seen to depend on past control moves and the future control move variables, Δu_{k+i} (see equations (5.14) and (5.15)). The future control moves, $\Delta u_k, \dots, \Delta u_{k+c-1}$, represent the decision variables for the optimization problem in equation (5.19). For calculation purposes, it is desirable to express \hat{y}_{k+i} such that the unknown decision variables appear explicitly in the objective function. Since Gaussian functions are factorable, it is possible to express the model prediction, \hat{y}_{k+i} , as an inner product of two vectors. The unknown decision variables are contained in one vector and all other known past quantities, including the network weights in the other. Thus, the RBF output can be rearranged as follows:

$$\hat{y}_{k+i} = \sum_{j=1}^{m_1} w_j \exp(\text{past} + \text{future}) = \begin{bmatrix} w_1 \exp(\text{past}) \\ \vdots \\ w_{m_1} \exp(\text{past}) \end{bmatrix}^T \begin{bmatrix} \exp(\text{future}) \\ \vdots \\ \exp(\text{future}) \end{bmatrix} \quad (5.20)$$

or

$$\hat{y}_{k+i} = \hat{\mathbf{y}}_{\mathbf{p},k+i}^T \hat{\mathbf{y}}_{\mathbf{f},k+i} \quad (5.21)$$

Subscripts **p** and **f** refer to the fact that the corresponding factors contain all known (**past**) and unknown (**future**) terms, respectively. Thus, only $\hat{\mathbf{y}}_{\mathbf{f},k+i}$ needs to be computed during every function call by the optimization algorithm.

As an example of RBF output factorization, consider the p -step control model with a prediction horizon, $p = 2$, output order, $N_y = 1$, and input order, $N_u = 2$. Then, the output of a 3-node RBF network, \hat{y}_{k+1} , in response to input vector (see equation (5.14)),

$$\mathbf{x}_{k+1} = [y_{k-1} \quad u_{k-2} \quad \Delta u_{k-1} \quad \Delta u_k] \quad (5.22)$$

can be written as,

$$\hat{y}_{k+1} = \sum_{j=1}^3 w_j \exp\left(-\frac{\|\mathbf{x}_{k+1} - \mathbf{t}_j\|^2}{\sigma^2}\right) \quad (5.23)$$

Let $t_{j,l}$ represent the l^{th} element of the node center vector, \mathbf{t}_j . Then, equation (5.23) can be rewritten as the product of two vectors as follows,

$$\hat{y}_{k+1} = \begin{bmatrix} w_1 \exp\left(\frac{(y_{k-1} - t_{1,1})^2 + (u_{k-2} - t_{1,2})^2 + (\Delta u_{k-1} - t_{1,3})^2}{\sigma^2}\right) \\ w_2 \exp\left(\frac{(y_{k-1} - t_{2,1})^2 + (u_{k-2} - t_{2,2})^2 + (\Delta u_{k-1} - t_{2,3})^2}{\sigma^2}\right) \\ w_3 \exp\left(\frac{(y_{k-1} - t_{3,1})^2 + (u_{k-2} - t_{3,2})^2 + (\Delta u_{k-1} - t_{3,3})^2}{\sigma^2}\right) \end{bmatrix}^T \begin{bmatrix} \exp\left(\frac{(\Delta u_k - t_{1,4})^2}{\sigma^2}\right) \\ \exp\left(\frac{(\Delta u_k - t_{2,4})^2}{\sigma^2}\right) \\ \exp\left(\frac{(\Delta u_k - t_{3,4})^2}{\sigma^2}\right) \end{bmatrix} \quad (5.24)$$

As discussed previously, the first factor on the right hand side represents $\hat{\mathbf{y}}_{\mathbf{p},k+1}$ and contains all known measurements (including the current measurement) and the known past inputs applied to the process. The second vector represents $\hat{\mathbf{y}}_{\mathbf{f},k+1}$ consists of the unknown control move, Δu_k , which is a decision variable of the optimization program presented in equation (5.19). All future p predictions can be expressed in a similar way.

As shown later, this factorized form facilitates analytic expressions for the gradient and Hessian of the objective function.

To illustrate the factorization of the RBF model prediction based on the general p -step control model structure, consider the model prediction at $k+1$. Let the center, \mathbf{t} , associated with a given node be partitioned as follows,

$$\mathbf{t} = [\mathbf{t}^y \mid \mathbf{t}^u \mid \mathbf{t}^{\Delta u}]^T \quad (5.25)$$

Vectors \mathbf{t}^y and \mathbf{t}^u contain elements corresponding to the delayed process outputs and inputs while $\mathbf{t}^{\Delta u}$ corresponds to the elements of the input moves. Then the factors $\hat{\mathbf{y}}_{\mathbf{p},k+1}$ and $\hat{\mathbf{y}}_{\mathbf{t},k+1}$ can be written as follows:

$$\hat{\mathbf{y}}_{\mathbf{p},k+1} = \begin{bmatrix} w_1 \exp \left[-\frac{1}{\sigma^2} \left\{ (y_{k-p+1} - t_{1,1}^y)^2 + \dots + (y_{k-p-N_y+1} - t_{N_y,1}^y)^2 + (u_{k-p} - t_{1,1}^u)^2 + \dots + (u_{k-p+2-N_u} - t_{N_u-1,1}^u) + (\Delta u_{k-p+1} - t_{1,1}^{\Delta u})^2 + \dots + (\Delta u_{k-1} - t_{p-1,1}^{\Delta u})^2 \right\} \right] \\ \vdots \\ w_{m_1} \exp \left[-\frac{1}{\sigma^2} \left\{ (y_{k-p+1} - t_{1,m_1}^y)^2 + \dots + (y_{k-p+1-N_y} - t_{N_y,m_1}^y)^2 + (u_{k-p} - t_{1,m_1}^u)^2 + \dots + (u_{k-p+2-N_u} - t_{N_u-1,m_1}^u) + (\Delta u_{k-p+1} - t_{1,m_1}^{\Delta u})^2 + \dots + (\Delta u_{k-1} - t_{p-1,m_1}^{\Delta u})^2 \right\} \right] \end{bmatrix} \quad (5.26)$$

and

$$\hat{\mathbf{y}}_{\mathbf{t},k+1} = \begin{bmatrix} \exp \left[-\frac{1}{\sigma^2} \left\{ \Delta u_k - t_{p,1}^{\Delta u} \right\}^2 \right] \\ \vdots \\ \exp \left[-\frac{1}{\sigma^2} \left\{ \Delta u_k - t_{p,m_1}^{\Delta u} \right\}^2 \right] \end{bmatrix} \quad (5.27)$$

The center element, $t_{l,m}$ corresponds to the m^{th} component of the l^{th} node center. Thus, $\hat{\mathbf{y}}_{\mathbf{t},k+i}$ contains all the input moves the optimizer must calculate, while $\hat{\mathbf{y}}_{\mathbf{p},k+i}$ is formed by

completely known quantities including past measurements, prior control moves and network weights.

As in linear MPC, we assume zero input moves after a control horizon of length c . Then, performing a similar exercise as above, the factors for the predictions, $\hat{y}_{k+1}, \dots, \hat{y}_{k+p}$, based on c future moves, $\Delta u_k, \dots, \Delta u_{k+c-1}$, take the following form:

$$\hat{y}_{p,k+i} = \mathbf{w}.*\exp\left[-\frac{1}{\sigma^2}\left(\sum_{j=0}^{N_y-1}(y_{k-p-j+i}\mathbf{1}-\mathbf{t}_{j+1}^y)^2 + \sum_{j=1}^{N_u-1}(u_{k-p-j+i}\mathbf{1}-\mathbf{t}_j^u)^2 + \sum_{j=0}^{p-1}(\Delta u_{k-p+j+i}\mathbf{1}-\mathbf{t}_{j+1})^2\right)\right] \quad (5.28)$$

where i ranges from 1 to p . $\mathbf{1}$ represents a column vector of size m_1 with unity elements. The operator " $*$ " is used to denote element by element multiplication. Note that when the index, i , equals p , the final summation drops out since j varies from 0 to -1 . The factor $\hat{y}_{\mathbf{t},k+i}$ that contains the future moves the optimization algorithm must calculate is:

$$\hat{y}_{\mathbf{t},k+i} = \exp\left[-\frac{1}{\sigma^2}\sum_{j=0}^{i-1}(\Delta u_{k+i-j-1}\mathbf{1}-\mathbf{t}_{p-j}^{\Delta u})^2\right], \quad i = 1, \dots, c \quad (5.29a)$$

$$\hat{y}_{\mathbf{t},k+i} = \exp\left[-\frac{1}{\sigma^2}\sum_{j=i-c}^{i-1}(\Delta u_{k+i-j-1}\mathbf{1}-\mathbf{t}_{p-j}^{\Delta u})^2\right], \quad i = c+1, \dots, p \quad (5.29b)$$

The exponential function in equations (5.28) and (5.29) implies a term by term application to each element of the column vector in the square brackets. Column vector \mathbf{t}_j is constructed by using the j^{th} element of all m_1 centers. Thus, any future prediction, \hat{y}_{k+i} , can be computed using equations (5.21), (5.28) and (5.29).

The objective function in equation (5.18) can be parameterized in terms of the network weights and the decision variables as follows:

$$\phi = \sum_{i=1}^p \Gamma_i \left(r_{k+i} - \hat{\mathbf{y}}_{\mathbf{p},k+i}^T \hat{\mathbf{y}}_{\mathbf{f},k+i} (\Delta \mathbf{u}) \right)^2 + \sum_{i=0}^{c-1} \Lambda_i (\Delta u_{k+i})^2 \quad (5.30)$$

The gradient of the objective function can be computed analytically as follows:

$$(\nabla \phi)_m = -2 \sum_{i=1}^p \Gamma_i \left(r_{k+i} - \hat{\mathbf{y}}_{\mathbf{p},k+i}^T \hat{\mathbf{y}}_{\mathbf{f},k+i} \right) \hat{\mathbf{y}}_{\mathbf{p},k+i}^T \frac{\partial \hat{\mathbf{y}}_{\mathbf{f},k+i}}{\partial \Delta u_m} + 2 \Lambda_m \Delta u_{k+m} \quad (5.31)$$

where $(\nabla \phi)_m$ denotes the m^{th} component of the gradient and $m = 0, \dots, c-1$. The partial derivative of $\hat{\mathbf{y}}_{\mathbf{f},k+i}$ is evaluated from equation (5.29) as,

$$\frac{\partial \hat{\mathbf{y}}_{\mathbf{f},k+i}}{\partial \Delta u_m} = \begin{cases} -\left(\frac{2}{\sigma^2} \right) \hat{\mathbf{y}}_{\mathbf{f},k+i}^* \left(\Delta u_{k+m} \mathbf{1} - \mathbf{t}_{p-i+m+1}^{\Delta u} \right), & \text{for } m+1 < i \\ 0, & \text{for } m+1 \geq i \end{cases} \quad (5.32)$$

Similarly, the (m,n) component of the Hessian matrix can be computed as follows,

$$(\nabla^2 \phi)_{m,n} = -2 \sum_{i=1}^p \left\{ \Gamma_i \left(r_{k+i} - \hat{\mathbf{y}}_{\mathbf{p},k+i}^T \hat{\mathbf{y}}_{\mathbf{f},k+i} \right) \left(\hat{\mathbf{y}}_{\mathbf{p},k+i}^T \frac{\partial^2 \hat{\mathbf{y}}_{\mathbf{f},k+i}}{\partial \Delta u_m \partial \Delta u_n} \right) - \left(\hat{\mathbf{y}}_{\mathbf{p},k+i}^T \frac{\partial \hat{\mathbf{y}}_{\mathbf{f},k+i}}{\partial \Delta u_m} \right) \left(\hat{\mathbf{y}}_{\mathbf{p},k+i}^T \frac{\partial \hat{\mathbf{y}}_{\mathbf{f},k+i}}{\partial \Delta u_n} \right) \right\} \quad (5.33)$$

The second order partial derivative in the above expression is calculated by,

$$\frac{\partial^2 \hat{\mathbf{y}}_{\mathbf{f},k+i}}{\partial \Delta u_m \partial \Delta u_n} = -\left(\frac{2}{\sigma^2} \right) \left(\hat{\mathbf{y}}_{\mathbf{f},k+i}^* \delta_{mn} \mathbf{1} + \frac{\partial \hat{\mathbf{y}}_{\mathbf{f},k+i}}{\partial \Delta u_n} \left(\Delta u_{k+m} \mathbf{1} - \mathbf{t}_{p-j} \right) \right) \quad (5.34)$$

when $(m+1), (n+1) \leq i$ and zero otherwise.

Using the above expressions for the gradient and Hessian, the optimization of the objective function in equation (5.30) can be performed using sequential quadratic

programming (SQP). The nonlinear output constraint in equation (5.19) can be written in terms of the factors of the model prediction and linearized using equation (5.32). The input constraints can be converted to input move constraints as in quadratic dynamic matrix control (Garcia and Morshedi, 1986). Analytical expressions for the gradient and Hessian greatly reduce the number of function calls and hence the computational burden during optimization. In addition, the separation of the decision variables in the model prediction ensures that only the unknown parts of the objective function and the gradient and Hessian required by the SQP algorithm are recalculated during optimization. Although the above expressions are derived for a single-input-single output system, similar expressions can be written for multiple-input multiple-output (MIMO) system by augmenting the objective function, $\hat{\mathbf{y}}_p$ and $\hat{\mathbf{y}}_f$ to include the multiple input/output variables.

5.5 Simulation Examples

To illustrate the performance of our proposed NMPC approach for a MIMO problem, consider control of the 2 x 2 hot/cold water mixing process discussed previously. In implementing the factorized RBF based NMPC for a MIMO process with N_o outputs, the network weights are stored in a $m \times N_o$ matrix, \mathbf{W} .

$$\mathbf{W} = [\mathbf{w}^1 \quad \mathbf{w}^2 \quad \dots \quad \mathbf{w}^{N_o}] \quad (5.35)$$

Equation (5.28) is then modified to provide $\hat{\mathbf{y}}_p$ for the l^{th} process outputs as follows:

$$\hat{\mathbf{y}}'_{p,k+i} = \mathbf{w}^l \cdot \exp \left[-\frac{1}{\sigma^2} \left(\sum_{j=0}^{N_y-1} (y_{k-p-j+i} \mathbf{1} - \mathbf{t}_{j+1}^y)^2 + \sum_{j=1}^{N_u-1} (u_{k-p-j+i} \mathbf{1} - \mathbf{t}_j^u)^2 + \sum_{j=0}^{p-1} (\Delta u_{k-p+j+i} \mathbf{1} - \mathbf{t}_{j+1})^2 \right) \right] \quad (5.36)$$

where l ranges from 1 to the number of process outputs, N_o . Vector $\hat{\mathbf{y}}_f$, which contains the future input moves, remains unchanged. The summary for the p -step control model is shown in Table 5.1. The error penalties Γ_i , $i = 1, \dots, p$ were set to 0.5 and 1.1 for the temperature and flow, respectively, with Λ_j , $j = 1, \dots, c$ set to 0.7 for both the inputs, u_1 and u_2 . The prediction and control horizons were assumed to be 8 and 3 sample intervals, respectively. At each control step, $\hat{\mathbf{y}}_p$ was calculated only once. During optimization, each objective function call involved calculation of $\hat{\mathbf{y}}_f$ and the computation of the model predictions based on equation (5.21). Optimization was performed using MATLAB's SQP function, *constr*, with analytical values of the gradient generated by equations (5.31) and (5.32). MATLAB's SQP function calculated the Hessian by finite difference. Inputs u_1 and u_2 were constrained to lie between 10% and 100%. No constraints were imposed on the outputs, the mixed stream temperature and flowrate. As in traditional linear MPC, the future p predictions are biased by the current value of the mismatch at each control execution step.

Control of the mixed stream temperature and flowrate for multiple setpoint changes is presented in Fig. 5.6. For the purpose of comparison, control of the mixing process by QDMC is also shown. The linear model used by QDMC is identified by step tests in the 40% to 60% region of the hot and cold valve signals. Disturbances enter the process at $k = 50, 125, 200$ and 375 by step changes in the hot and cold leg temperatures as shown in the figure. The RBF model accounts for hot and cold fluid temperatures changes by incorporating these measurements in the RBF input vector as shown below,

$$\mathbf{x}_{k+i}^{\text{augmented}} = \left[\mathbf{x}_{k+i} \quad \vdots \quad T_{hot,k} \quad T_{hot,k-1} \quad T_{cold,k} \quad T_{cold,k-1} \right] \quad (5.37)$$

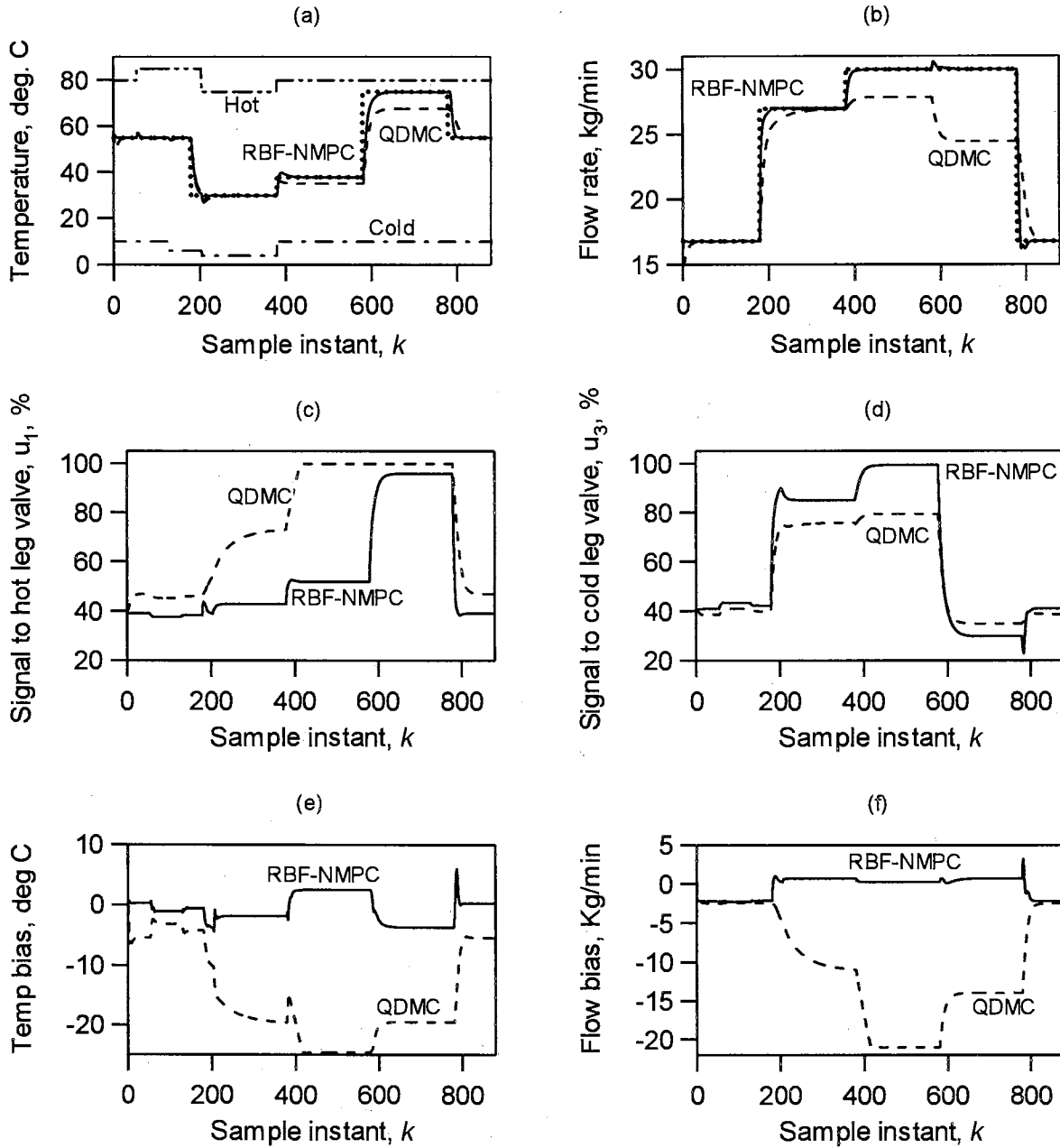


Figure 5.6: (a) and (b) illustrate control of the 2x2 mixing process in presence of measured disturbances in hot leg (at $k = 50, 200,$ and 375) and cold leg (at $k = 125, 200,$ and 375) temperatures by the RBF based NMPC and linear QDMC. Setpoint changes were made at $k=175, 375, 575$ and 775 for temperature and at $k=175, 375$ and 775 for flowrate. (c) and (d) show control action implemented by the NMPC and QDMC controllers. (e) and (f) process model mismatch for temperature and flowrate

Similarly, the center vector is also augmented to include center elements corresponding to the disturbance measurements. During the future prediction phase, it is assumed that the current values of the hot and cold leg temperatures remain constant over the prediction horizon. In this example, the QDMC also uses a model to account for the temperature disturbances. Both, the RBF based NMPC and QDMC successfully reject these measured disturbances by use of measurement bias.

At $k = 175$, the temperature setpoint is changed from the initial value of 55°C to 30°C while the flow setpoint is changed from 17 kg/min to 27 kg/min . The QDMC controller responds by making large positive changes in the cold leg valve to decrease the temperature of the mixed stream. The hot leg valve has a higher throughput (nearly double) than the cold leg valve at a given stem position. Thus, the QDMC controller also opens the hot leg valve to allow for increased flow rate. However, at lower temperatures the mixed stream temperature becomes increasingly sensitive to hot water flow. When the flow rate setpoint is further increased at $k = 375$, the hot leg valve further opens till it saturates at 100% and the process is no longer maintained at the respective setpoints. At $k = 775$, the setpoints are brought to the region of linear model development and the linear QDMC controller is once again able to control the plant. On the other hand, the RBF based NMPC controller exhibits excellent control over the entire operating region. Also shown in Fig. 5.6 is the process-model mismatch for the two outputs. Tight control by the RBF based NMPC controller is a consequence of good predictions by the RBF model over the entire range of operation.

It is of practical interest to investigate the computational requirements for the factorized RBF model based NMPC algorithm. To evaluate computational benefits, the computation time needed by the factorized RBF model based NMPC for the above problem is compared with a non-factorized RBF based NMPC algorithm which also uses the p -step control model. Unlike the factorized approach where \hat{y}_p is calculated only once during each control execution, the non-factorized algorithm computes the entire expression for RBF model predictions (similar to equation (5.2)) during each iteration of the nonlinear program at every control execution. Also, the gradient information is calculated numerically with the non-factorized approach. Table 5.2 documents the results. The results were generated by using the *tic-toc* command in MATLAB and represent the actual time needed by the computer to complete the controller-related calculations. As evident from Table 5.2, the factorized RBF based NMPC is an order of

Table 5.2

Comparison of computation time needed in control of hot/cold mixing example.
(simulation for 900 samples or 4450 seconds)

	Factorized RBF based NMPC	Non-factorized RBF based NMPC (gradients evaluated numerically)	QDMC
Real-time needed for computation (seconds)	268	2984	17

magnitude efficient than the non-factorized approach. This is a significant reduction considering the non-factorized NMPC approach is two orders of magnitude more demanding than linear QDMC.

To test the RBF model based NMPC algorithm in presence of unmeasured disturbances, we again consider control of the hot/cold mixing process. However, unlike the previous example, the RBF model does not utilize the hot and cold leg temperatures. Additionally, various process non-idealities, including valve stiction, drifts in process parameters, drifts in temperature, and drifts and spikes in upstream pressure drops of the hot and cold legs, are built into the governing equations for the process (Rhinehart, 1998). The control simulation results are shown in Figure 5.7. The mixed stream flowrate measurement was filtered using a CUSUM filter (Rhinehart, 1992) prior to input to the MPC scheme. No filtering was employed for the temperature measurement. The RBF network was trained on the unfiltered noisy data. Use of regularization during training ensured that the network does not overfit the noisy data. The number of nodes and the regularization parameter were selected so that similar statistics are obtained for network performances on the training and test sets. Based on the simulation results, it is observed that RBF based MPC exhibits tight control of the mixing processes in face of the non-idealities described above.

As a final example, we consider control of a non-adiabatic, continuous, stirred-tank reactor with a first order irreversible reaction. The heat of reaction is removed by circulation of cooling water in the reactor jacket. Uppal et al. (1974) describe the

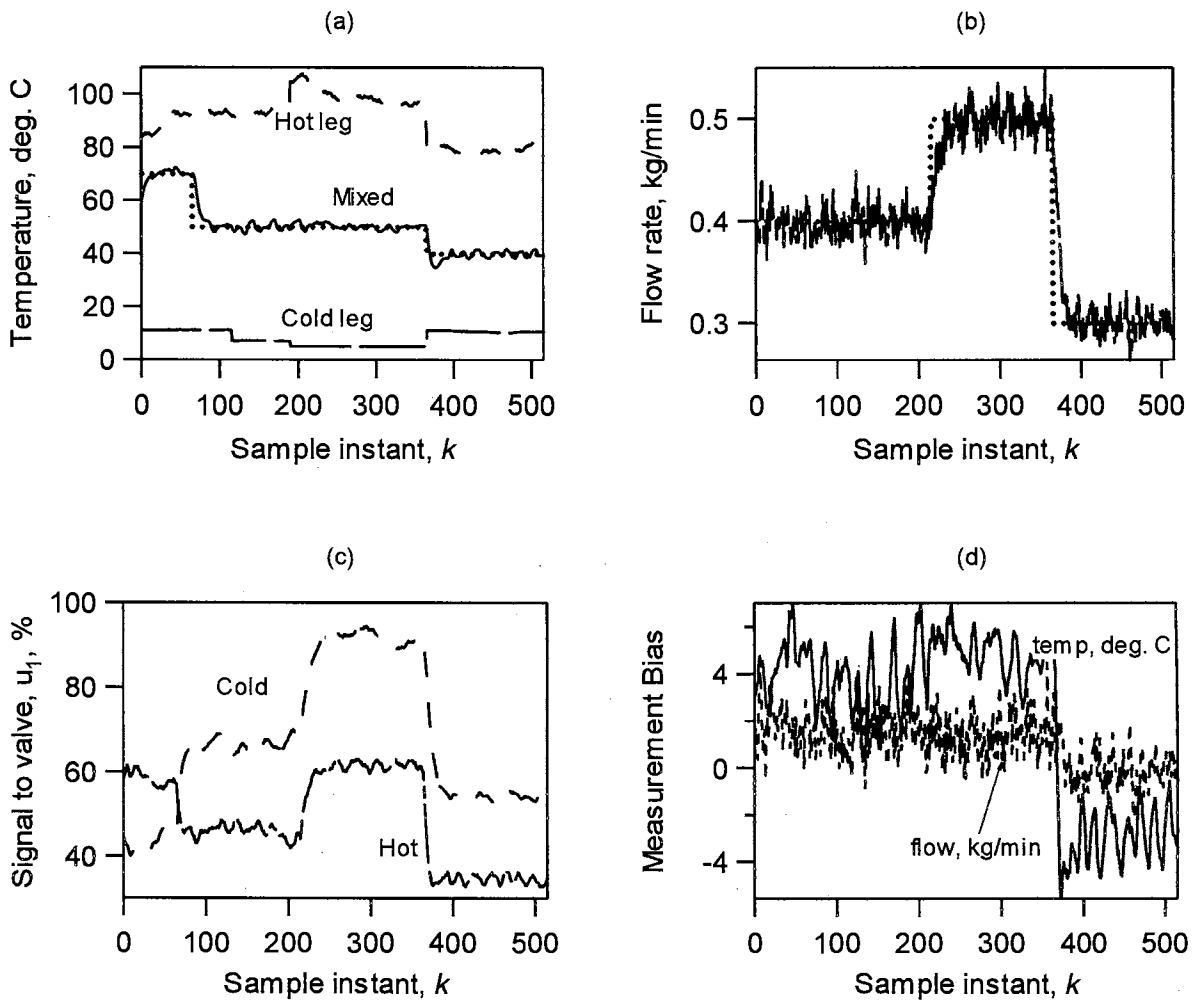


Figure 5.7: (a) and (b) illustrate the control of the 2x2 mixing process in presence of unmeasured disturbances in hot and cold leg temperatures (steps, drifts and spikes). Setpoint changes were made at $k=0, 65$ and 365 for temperature and at $k=0, 200, 365$ for flowrate. (c) control action implemented by the NMPC controller. (d) process model mismatch for temperature and flowrate.

following system of equations that govern the process dynamics,

$$\dot{x}_1 = -x_1 + Da(1-x_1)\exp\left(\frac{x_2}{1+x_2/\gamma}\right) \quad (5.38a)$$

$$\dot{x}_2 = -x_2 + BDa(1-x_1)\exp\left(\frac{x_2}{1+x_2/\gamma}\right) + \beta(u-x_2) \quad (5.38b)$$

States x_1 and x_2 represent reactant conversion and a dimensionless reactor temperature. The manipulated variable, u , is a dimensionless reactor jacket temperature. The steady state characteristic of the reactor for parameters, $\beta = 3.0$, $\gamma = 40$, $B = 22$ and $Da = 0.082$, is shown in Fig. 5.8. The process exhibits low gain at small values of conversion (1% to 4%) and considerably higher gain (in excess of 80 times the low gain) at higher conversions (>18%). The performance of the RBF based NMPC and linear QDMC is shown in Fig. 5.8. A 125-node RBF network was trained to emulate process behavior over the range of 1% to 20% of conversion. For the QDMC controller, a linear model was developed by conducting a step test at low conversions (1% to 4%). A prediction and control horizon of 9 and 3 samples, respectively, was used for both algorithms. At low conversion, the QDMC controller shows adequate performance. However, when a new setpoint ($x_1 = 0.18$) is implemented, the QDMC controller does not recognize the high plant gain in this new region, resulting in aggressive control action. The system becomes unbounded when the next setpoint change (from 0.18 to 0.19) is implemented. On the other hand, the RBF based NMPC controller provides tight control over the entire range of operation.

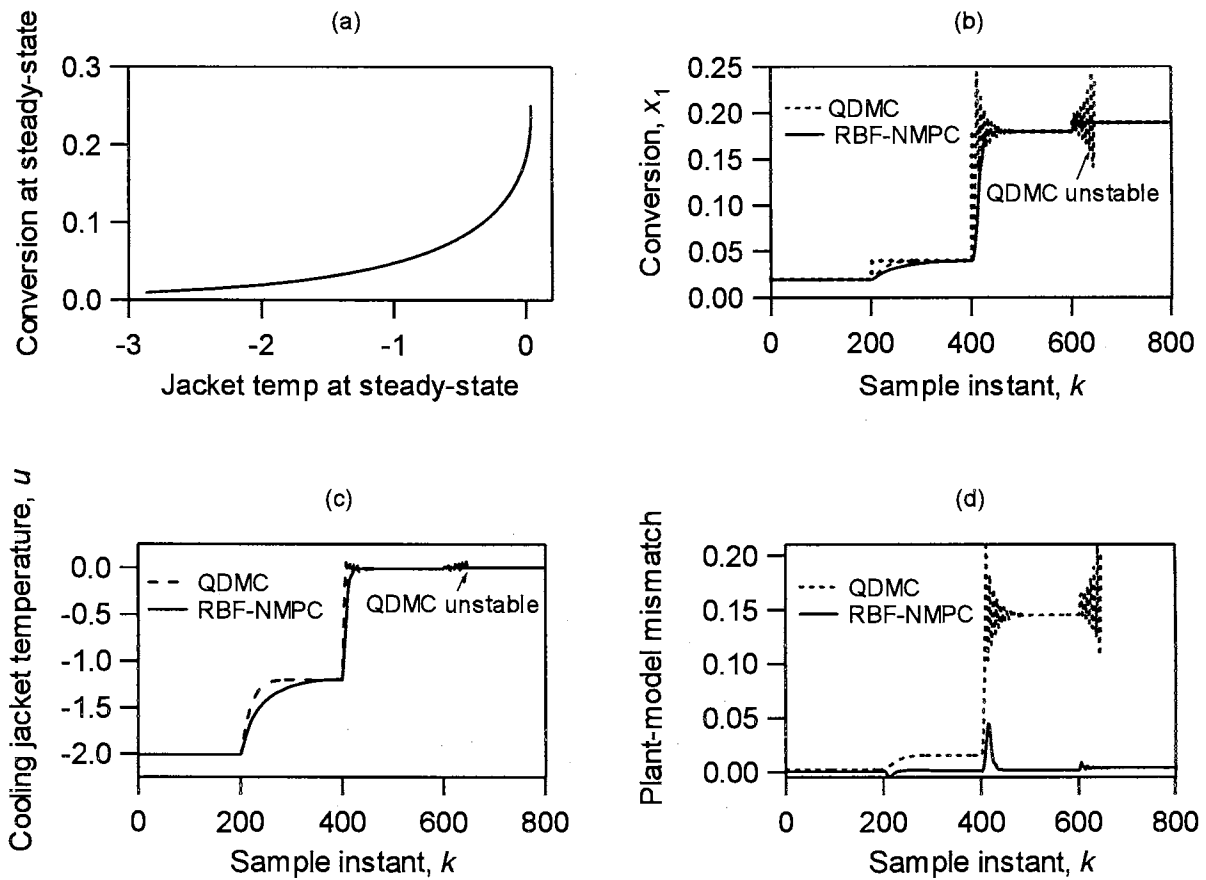


Figure 5.8: (a) shows the steady state characteristics of the CSTR process. (b) illustrates the control of the CSTR process using a linear QDMC controller and the RBF based NMPC algorithm. Setpoint changes were made at $k=200$, 400 , and 600 . The linear QDMC control system becomes unstable in high gain region ($k > 646$). The RBF-NMPC controller provides tight control in both, the low and high gain regions. (c) control action. (d) process model mismatch.

5.6 Conclusions

The most significant contribution of the proposed methodology is the ability to provide nonlinear control. The proposed NMPC technique integrates two well-accepted concepts in the modeling and control communities, RBF networks and Model Predictive Control. RBF-based NMPC controller design offers the potential of a generic methodology for a large number of industrial processes as many process nonlinearities can be expressed in terms a set of radial basis functions (Hartman et al., 1990).

The methodology can be applied to multivariable systems as illustrated by application to the 2 x 2 hot/cold water mixing simulation. Conceptually, the methodology can be applied to any $m \times n$ system. However, the use of RBF networks for the process model introduces practical questions of scale. As evident from the development presented in Section 5.4, the number of nodes used to model a multivariable system directly impacts the computational resources required to implement the proposed methodology. There is an obvious premium on efficient modeling to minimize the total number of RBF nodes. Additional work is required to determine the point at which scale becomes a problem. The potential to compensate for reduced control model fidelity with additional computational effort is clearly an issue of interest in this situation.

The potential problems of scale mentioned previously have been mitigated in part by leveraging the factorability of the Gaussian functions used in RBF networks. This property was exploited to express model predictions as an inner product of two vectors, one containing the decision variables of the MPC optimization program with the other

made up entirely of known past quantities. This minimizes computational effort since only the unknown parts of the objective function need to be re-evaluated during each optimization call. Additional computational benefits are realized due to the compact representations for the MPC controller objective function and the availability of analytic expressions for the gradient and Hessian. The objective function takes the form of a sum of weighted Gaussian functions. Opportunities may exist to further exploit the radial symmetry of the Gaussian functions to tailor more efficient MPC optimization algorithms.

In the proposed NMPC scheme, the choice of the prediction horizon p influences both the controller performance as well as the control model. The p -step model specifically eliminates iterative dependency on model predictions. However, there is a cost. The most recent output measurement available for use by the model is always p steps prior in time. The p -step model was proposed due to problems with cascaded 1-step models using industrial magnitude prediction horizons. Performance of p -step models using these relatively long prediction horizons needs to be demonstrated.

6 APPLICATION OF FACTORIZED RBF BASED MPC TO THE EASTMAN CHALLENGE PROBLEM

Chapter Overview

The purpose of this paper is to explore the application of a factorized radial basis function (RBF) network model based nonlinear MPC (NMPC) algorithm (Bhartiya and Whiteley, 2000) for control of the Eastman process. The algorithm derives its computational efficiency by factorizing the model response. Control inputs are calculated based on optimization of a nonlinear objective function using the sequential quadratic programming technique. A brief description of the algorithm is included in the paper. Key elements of a plantwide control strategy outlined by McAvoy and Ye (1994) are discussed. A subset (4x4) is selected for control by the RBF based NMPC algorithm with the remaining plant uses the McAvoy and Ye scheme. An RBF model is then developed for this subset and finally, results using factorized RBF based NMPC approach for control of the Eastman process are presented.

6.1 Introduction

The Eastman challenge problem (Downs and Vogel, 1993) has been used extensively to evaluate different control strategies. In this paper, we demonstrate successful application of a new nonlinear model predictive control (NMPC) algorithm (Bhartiya and Whiteley, 2000) to the Eastman process. The NMPC algorithm employs a nonlinear process model in the form of a radial basis function (RBF) network. The algorithm exploits the factorability of RBF models in a traditional model predictive

control framework. The Eastman challenge problem provides an ideal testbed to evaluate the computational and nonlinear control benefits of the proposed NMPC algorithm.

Model predictive control (MPC) is used extensively to control high value, constrained, multivariable industrial processes (Qin and Badgwell, 1997). However, the current generation of commercially available MPC packages generally relies on linear process models. Excellent performance is realized as long as the plant operates close to the conditions used to create the linear approximation of the process. The goal for the next generation of control software is to provide similar capability across the whole range of possible plant operating conditions. MPC can potentially provide this capability if a more accurate nonlinear model of the process is employed.

However, use of a nonlinear model presents additional challenges relative to linear MPC: 1) the complexity of nonlinear systems makes systematic development of nonlinear system identification technique difficult (Pearson and Ogunnaike, 1997), and 2) nonlinear MPC requires solution of a nonlinear program at each sampling instant making implementation more involved (Henson, 1998). Both of these problems have been investigated by a number of researchers. The following paragraphs provide a brief account of developments reported in literature.

A straightforward extension of linear theory to nonlinear system identification is generally difficult due to the complexity of nonlinear systems. Cook (1986) indicates that because of the large number of different types of nonlinearities can occur in practice,

extending a basic control scheme to account for all possibilities is unrealistic. One way of tackling the general nonlinear problem is to employ a framework, within which a large number of nonlinear processes can be adequately approximated. Volterra and Hammerstein models (Agarwal and Seborg, 1987) and neural networks (Hussain, 1999) have been studied as nonlinear modeling tools. As an alternative, the NMPC controller may be based on a fundamental model which is derived from conservation laws and constitutive equations. The continuous time differential equations are discretized by some method (e.g., orthogonal collocation on finite elements (Meadows and Rawlings, 1997) to allow incorporation in the NMPC scheme.

A common approach to the nonlinear optimization required in NMPC is based on successive linearization of nonlinear models. Garcia (1984) proposed a nonlinear QDMC algorithm, a simple extension of DMC/QDMC based on online successive linearization of a mechanistic nonlinear model. Nonlinear MPC using closed-loop state estimation by an extended Kalman filter has been proposed by Lee and Ricker (1994). Gattu and Zafiriou (1995) augmented the system states with stochastic states to account for modeling errors and disturbances. Banerjee et al. (1997) describe a method of state estimation for nonlinear systems that are subject to multiple operation regimes and make transitions between them. The nonlinear process is approximated by a linear parameter varying system which consists of local linear models. Krishnan and Kosanovich (1998) also present a multiple model based MPC scheme. The linear time invariant models are computed offline along a pre-defined reference trajectory of a batch process. Each of the above nonlinear MPC techniques use the standard quadratic programming optimization

method to obtain control inputs. Use of nonlinear programming techniques have also been used (Bequette, 1991). Mayne (1996) argues that model constraints corresponding to satisfaction of model equations over the prediction horizon, generally, result in a nonconvex optimization. Various solution methods of solving the online finite horizon nonlinear control problem are available (Mayne, 1995; Santos *et al.*, 1995). Staus *et. al* (1996) study a class of nonlinear problems for which the global optimum can be computed online. A similar study is reported by Srinivas and Arkun (1995).

In the past few years, renewed interest has been paid to neural network models because of their simple structure and effective computational performance. In particular, artificial neural networks have been used for inferential modeling (Bhide *et al.*, 1995), fault diagnosis (Venkatasubramanian *et al.*, 1990), process identification (Chen *et al.*, 1990) and model based control (Bhat and McAvoy, 1990; Su and McAvoy, 1997). Hussain (1999) provides a summary of a number of applications reported in literature. When applied for predictive control most utilize a feedforward network architecture, while a few use the recurrent type. Among the various neural network choices, multi-layer perceptron and radial basis function networks are the most popular in control and identification applications. Both of these networks are capable of universal approximation (Cybenko, 1987; Hartman *et al.*, 1990).

The Eastman process has been used for many purposes including evaluation of various linear and nonlinear MPC schemes. McAvoy and Ye (1994) outlined a multiple single loop strategy. The loop pairings were determined based on the relative gain array,

Niederlinski index and nonlinear disturbance and saturation analyses. Banerjee and Arkun (1995) proposed a two-tier control configuration procedure to design the SISO loops. Kanadibhotla and Riggs (1995) applied Generic Model Control (GMC) to the reactor temperature loop and a nonlinear steady-state compensating controller to the stripper composition loop. The remainder of the plant was controlled using standard PI controllers. Other SISO strategies have also been outlined by Desai and Rivera (1993), Lyman and Georgakis (1995), Luyben (1996), and Ricker (1996). Tyreus (1999) used a partial control structure in which the controlled variables were identified by a thermodynamically motivated dominant variable method.

Palavajhala et al. (1993) compared a SISO strategy with a DMC implementation. Ricker and Lee (1995b) proposed a nonlinear (mechanistic), dynamic model of the Eastman process consisting of 15 adjustable parameters. In a later publication, Ricker and Lee (1995a) presented a nonlinear MPC for an 8x8 subset of the Eastman process using their nonlinear mechanistic model of the plant. The remaining controlled variables were stabilized by SISO feedback loops in a cascade structure. Variables manipulated by the MPC controller were setpoints to the SISO loops. The model was linearized at every control execution step and process inputs calculated. The nonlinear model predictive control (NMPC) showed good results over the entire spectrum of plant operation. While they did not provide details, Ricker and Lee stated that they tested MIMO strategies employing time-invariant models such as DMC, QDMC, IDCOM, etc. and noted that these models were too sensitive to gain variations and could not be tuned for robust performance. The DMC implementation of Palavajhala et al. violated the $\pm 5\%$

variability specification on product compositions. Srinivas and Arkun (1997) used linear input output models with MPC in a supervisory mode to control the Eastman process. The current work differs from those listed above in that MPC is based on a nonlinear RBF model identified from input-output data.

The remainder of this paper is organized as follows. The essential points for the proposed nonlinear MPC algorithm are presented in Section 2. A brief overview of the Eastman process is presented in Section 3. Application of the NMPC algorithm for control of the Eastman process is described in Section 4. Results are presented in Section 5 followed by conclusions in Section 6.

6.2 RBF-Based Nonlinear MPC Algorithm

A complete description of the algorithm is presented in (Bhartiya and Whiteley, 2000). From an MPC standpoint, the most important point is the use of a radial basis function (RBF) neural network to predict future process behavior. Feedforward RBF networks have been widely used as models of dynamic processes (Chen *et al.*, 1990; Pottmann and Seborg, 1997). A number of applications using the RBF model with model predictive control (MPC) have been reported in the literature. Hunt and Sbarbaro (1992) use an RBF model for pH control of a neutralizing tank. Pottmann and Seborg (1997) train a RBF network to directly emulate a predictive controller.

The radial basis function (RBF) network consists of an input layer, a hidden layer and an output layer. In the input layer, unweighted inputs, \mathbf{x} , are directly transmitted to

the hidden layer nodes. Each hidden layer node consists of a radial basis function. In the current work, the Gaussian function is employed,

$$g_j(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{t}_j\|^2}{\sigma^2}\right) \quad (6.1)$$

where \mathbf{t}_j is the center of the j^{th} hidden node. It is assumed that each node is of fixed width σ . Other radial functions such as inverse multiquadric have also been used (Pottman and Seborg, 1992). The output layer performs a weighted summation of hidden node outputs to give the network output,

$$\hat{y}_i(\mathbf{x}) = \sum_{j=1}^{m_1} w_{ij} g_j(\mathbf{x}) \quad (6.2)$$

where m_1 represents the number of hidden nodes and w_{ij} is the weight between the i^{th} output node and the j^{th} hidden node. RBF network training consists of fixing the hidden node centers and their width, and the weights to approximate the input-output mapping provided by the data. Often, the training data consists of noisy measurements. This may cause the minimization of prediction error to be an ill-posed problem. Regularization theory is used to address this problem by incorporating smoothness constraints or *a priori* knowledge (Poggio and Girosi, 1990) in the minimization problem. Regularization may also be used to incorporate first-principles knowledge (Gurumoorthy and Kosanovich, 1998) directly in the training algorithm.

The proposed NMPC algorithm employs an RBF model to provide non-iterative sequential predictions over a prediction horizon of length p . The model is structured to avoid dependency of future model predictions on previous model predictions. Future

predictions, \hat{y}_k , are related to past measurements y_i and inputs u_i , delayed by p samples and input moves, Δu_i , as follows,

$$\hat{y}_{k/k-p} = F\left(y_{k-p}, \dots, y_{k-p+1-N_y}, u_{k-p-1}, \dots, u_{k-p+1-N_u}, \Delta u_{k-p}, \dots, \Delta u_{k-1}\right) \quad (6.3)$$

where the control move at the k^{th} instant,

$$\Delta u_k = u_k - u_{k-1} \quad (6.4)$$

N_y and N_u represent the orders of output and input, respectively. Function F is defined by the RBF network. From a process response approximation point of view, the delayed input/output measurements, viz. $y_{k-p}, \dots, y_{k-p+1-N_y}$ and $u_{k-p-1}, \dots, u_{k-p-N_u}$, respectively, provide a reference to the state of the system p samples in the past. Note that subscripts now refer to instances in time rather than connectivity within the RBF network.

Based on the model in equation (6.3), the following form of the input vector is chosen for a single-input, single-output system,

$$\mathbf{x}_k = \left[y_{k-p} \quad \dots \quad y_{k-p+1-N_y} \quad u_{k-p-1} \quad \dots \quad u_{k-p+1-N_u} \quad \Delta u_{k-p} \quad \dots \quad \Delta u_{k-1} \right] \quad (6.5)$$

Figure 6.1 illustrates the input vector with elements located on a timeline. Measurable disturbances can be accounted by augmenting the input vector \mathbf{x} to include the disturbance variables. The RBF prediction of the process output at instant k is then obtained by equation (6.2). Multiple-inputs and multiple outputs can be handled by including these in the RBF input vector, \mathbf{x} .

Control moves are determined using the traditional MPC approach. The algorithm computes the manipulated variable profile over a control horizon by optimizing an objective function defined over the prediction horizon, subject to constraints. Only the

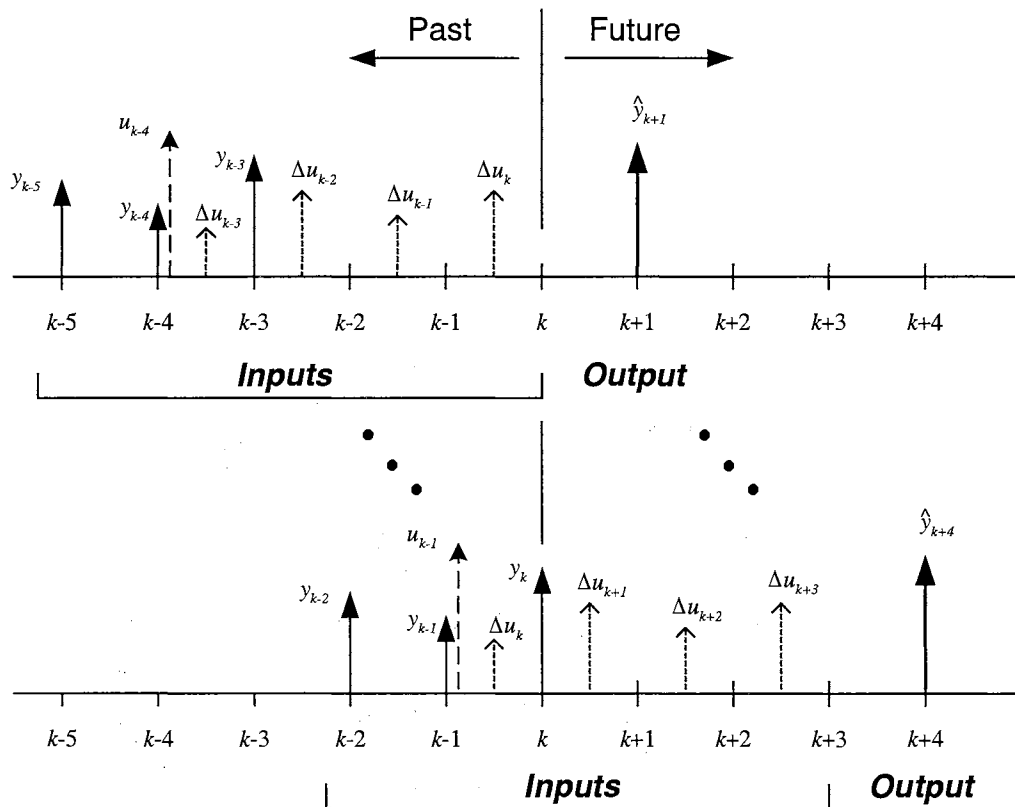


Figure 6.1: Timelines showing inputs to the p -step control model. For this example, $p = 4$, $N_y = 3$, and $N_u = 2$. Predicted outputs are generated from known information only, previous model predictions for y are not used in model input.

first move is implemented and the procedure is repeated at every sampling instant. We use the following objective function:

$$\phi = \sum_{i=1}^p \Gamma_i (r_{k+i} - \hat{y}_{k+i})^2 + \sum_{i=0}^{c-1} \Lambda_i (\Delta u_{k+i})^2 \quad (6.6)$$

Variables p and c represent the prediction and control horizons, respectively. Γ_i and Λ_i denote the error penalty and move suppression factors at the i^{th} instant. The MPC control law can be stated as,

$$\begin{aligned} & \arg(\min \phi) \quad \text{such that} \\ & \Delta u_k, \Delta u_{k+1}, \dots, \Delta u_{k+c-1} \\ & y_{\min} \leq \hat{y}_{k+i} \leq y_{\max} \\ & \Delta u_{\min} \leq \Delta u_{k+i} \leq \Delta u_{\max} \\ & u_{\min} \leq u_{k+i} \leq u_{\max} \end{aligned} \quad (6.7)$$

The future model predictions, \hat{y}_{k+i} , depend on past control moves and the future control move variables, Δu_{k+i} (see equation (6.3)). The future control moves, $\Delta u_k, \dots, \Delta u_{k+c-1}$, represent the decision variables for the optimization problem in equation (6.7). For calculation purposes, it is desirable to express \hat{y}_{k+i} such that the unknown decision variables appear explicitly in the objective function. The key idea in the factorized RBF model lies in expressing the model prediction, \hat{y}_{k+i} , as an inner product of two vectors. The unknown decision variables of the nonlinear program (equation (6.7)) are contained in one vector and all other known past quantities, including the network weights in the other. Thus, the RBF output can be rearranged as follows:

$$\hat{y}_{k+i} = \sum_{j=1}^{m_1} w_j \exp(\text{past} + \text{future}) = \begin{bmatrix} w_1 \exp(\text{past}) \\ \vdots \\ w_{m_1} \exp(\text{past}) \end{bmatrix}^T \begin{bmatrix} \exp(\text{future}) \\ \vdots \\ \exp(\text{future}) \end{bmatrix} \quad (6.8)$$

or,

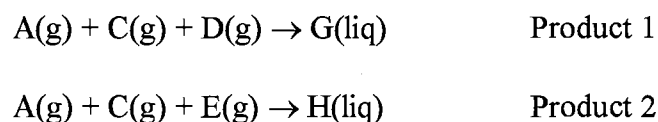
$$\hat{y}_{k+i} = \hat{y}_{\mathbf{p},k+i}^T \hat{y}_{\mathbf{f},k+i} \quad (6.9)$$

Subscripts **p** and **f** refer to the fact that the corresponding factors contain all known (**past**) and unknown (**future**) terms, respectively. Thus, only $\hat{y}_{\mathbf{f},k+i}$ needs to be computed during every function call by the optimization algorithm. All future p predictions can be expressed in a similar way. The factorized form provides analytic expressions for the gradient and Hessian of the objective function (Bhartiya and Whiteley, 2000). As demonstrated later, the availability of analytic expressions significantly reduces the computational requirements associated with the algorithm. In addition, the separation of the decision variables in the model prediction ensures that only the unknown parts of the objective function and the gradient and Hessian required by the sequential quadratic programming (SQP) algorithm are recalculated during optimization.

The nonlinear output constraint in equation (6.7) can be written in terms of the factors of the model prediction and linearized. However, output constraints have not been used in the current work. The input constraints can be converted to input move constraints as in quadratic dynamic matrix control (Garcia and Morshedi, 1986).

6.3 Overview of Eastman Process

The process consists of producing two products from four reactants by the following reactions (Downs and Vogel, 1993),





In addition to the reactants, an inert B is also present. Five unit operations, viz. an exothermic, two-phase reactor, a product condenser, a vapor-liquid separator, a compressor, and a stripper column with a reboiler are employed. A schematic of the Eastman process is shown in Figure 6.2. The gaseous reactants react to form the liquid products. The heat of reaction is removed by an internal cooling bundle. Although a large holdup of products G and H exists in the reactor, there is no liquid effluent stream. Unreacted feed along with the vaporized product leaves the reactor through a partial condenser to the vapor-liquid separator. The separated liquid contains most of the products G and H and small amounts of reactants D, E and byproduct F. Unreacted reactants, A and C and the inert B are essentially noncondensibles and are recycled back to the reactor by a compressor. A purge stream is provided to avoid buildup of inert B.

Figure 6.2

Finally, the separated liquid enters a stripper column to recover the unreacted reactants. Steam is provided as the heat source to recover the volatile components. Products G and H exit the stripper base and are separated in a downstream unit.

The main control objective is to maintain product rate and compositions at their respective setpoints ($\pm 5\%$ for production rate and ± 5 mol % G) and minimizing variability of A, D and C feed streams due to small available holdup. This must be accomplished while keeping other process variables within operational constraints to

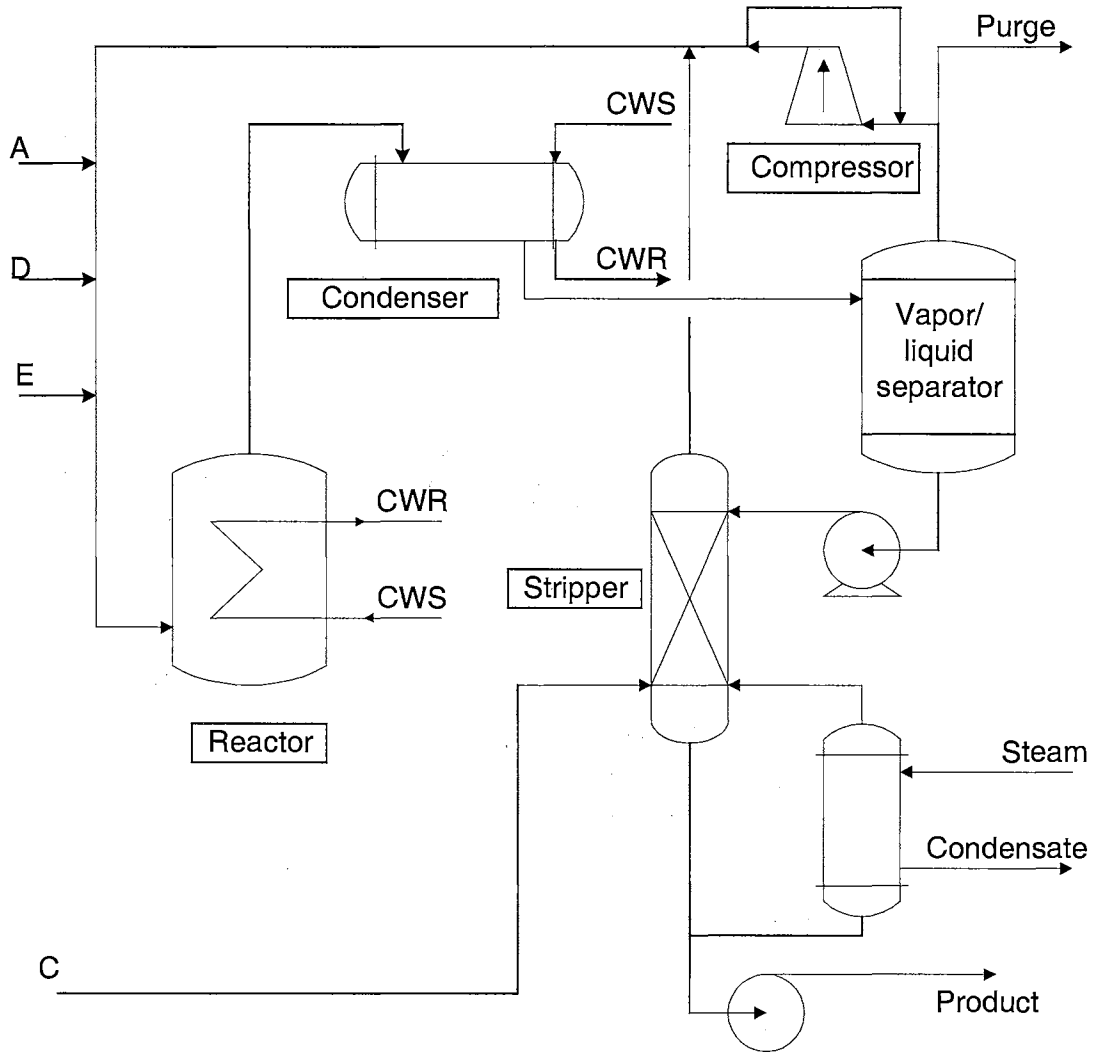


Figure 6.2: Schematic of the Eastman challenge process.

ensure equipment protection. Additional details are available in Downs and Vogel (1993).

6.4 Application of RBF-Based NMPC to Eastman Problem

6.4.1 Plantwide Control Strategy

As a starting point, we used the SISO strategy previously reported by McAvoy and Ye (1994) for the base regulatory control system. The SISO loop pairings recommended by McAvoy and Ye are presented in Table 6.1. Two minor modifications were made to McAvoy and Ye's scheme:

- 1) Setpoint of condenser cooling water return is used to control the condenser temperature (Gain = 5.0 %/°C; integral time = 50 minutes). Recycle flow was not controlled.
- 2) The flow controller for the A & C stream (stream 4) was tuned more aggressively (Gain = 0.3 %/kscmh; integral time = 0.09 minutes).

Selection of the manipulated and controlled variables associated with the NMPC controller was made based on the control objectives specified in the Eastman problem statement. Downs and Vogel suggested the following setpoint changes from the base case conditions:

- 1) production rate step change (-15%),
- 2) product mix step change (50G/50H to 40G/60H on mass basis),
- 3) reactor operating pressure step change (-60 kPa)
- 4) purge gas composition of component B step change (+2 mol %).

Table 6.1**Input-output pairing determined by McAvoy and Ye (McAvoy and Ye, 1994)**

Controlled Variable	Manipulated Variable
Reactor temperature	Reactor coolant temperature setpoint
Reactor pressure	A feed rate setpoint
Reactor level	E feed rate setpoint
Separator level	Underflow rate setpoint
Stripper temperature	Steam flow setpoint
Stripper level	Stripper underflow rate setpoint
Production rate	A & C feed rate setpoint
G/H composition in product	D/E ratio setpoint
E composition in product	Stripper temperature setpoint
Inert B composition in purge stream	Purge flow rate setpoint
Recycle flow	Condenser cooling water temperature setpoint
Compressor power	Recycle valve

We therefore chose to control the production rate, reactor pressure, and purge gas composition. Since the PI controller in the McAvoy and Ye scheme performs adequately for control of product quality (within the tolerance of ± 5 mol % G as suggested by Downs and Vogel), product quality was not included in the NMPC scheme.

A fourth controlled variable was selected however. Disturbance IDV(1) is defined by a step change in the A/C feed ratio in stream 4. This disturbance upsets the reaction stoichiometry causing the gaseous reactants to accumulate in the reactor-recycle loop with an associated increase in reactor pressure. A similar situation is encountered with disturbance IDV(8) which relates to random variations in the A, B and C composition of stream 4. In order to address reactor stoichiometry disturbances, the A/C

mole ratio in the reactor feed was included as an additional controlled variable in the NMPC scheme.

The manipulated variables were selected based on the steady-state gain matrix presented by McAvoy and Ye (1994) along with the following observations.

- 1) The A & C feed stream consists of reactant C and the bulk of reactant A, both of which are needed to form products, G and H. Thus, this stream directly affects the production rate. It is also the carrier of inert B and hence directly influences the purge gas composition of component B.
- 2) The reactor temperature strongly influences the reactor pressure. A decrease in reaction temperature quenches the reaction. In this case, the unreacted reactants, which are essentially noncondensibles, accumulate in the recycle loop causing the reactor pressure to rise. On the other hand, an increase in reaction temperature increases the rate of formation and subsequent vaporization of products. The vaporized products are then condensed leading to a drop in the recycle loop pressure.
- 3) Stream A has a much smaller throughput compared to the A & C stream and supplies the remainder of component A needed for the reaction. Thus, this stream can be effectively used for control of the A/C ratio in the reactor feed.
- 4) Purge rate is a logical choice for control of component B in purge. It also affects the pressure by avoiding accumulation of noncondensibles.

Based on the above considerations, we chose A feed, A & C feed, reactor temperature and purge rate as the set of manipulated variables used by NMPC scheme.

Of the twenty load disturbances suggested by Downs and Vogel, seven {IDV(4, 5, 7, 11, 12, 14 and 15)} are directly addressed by the base SISO regulatory control system. Disturbance IDV(2) is defined by a step increase in the B composition while maintaining A/C composition constant in stream 4. Since B is an inert, it does not upset the stoichiometry in the reactor. The presence of the disturbance is detected by the consequent rise in the purge gas composition of component B and should be addressed by the NMPC controller. Disturbances IDV(3, 9 and 10) relate to temperature changes in feed streams 2 and 4. These disturbances manifest themselves by changing the heat content of the reactor. Thus, these disturbances can be rejected by manipulation of the reactor temperature setpoint.

A summary of the resulting 4x4 subset selected for control by NMPC is provided in Table 6.2. PI controller tuning parameter values used are as in McAvoy and Ye except for the NMPC subset and the two modifications noted previously. Note that each of the variables manipulated by the NMPC controller are setpoints for the lower level PI controls.

Table 6.2

Controlled and manipulated variables for control by the RBF based MPC controller.

MPC controlled variables	MPC manipulated variables
Reactor pressure	Reactor temperature setpoint
Composition of B in purge	Purge rate setpoint
Production rate	A & C feed rate setpoint
A/C mole ratio in reactor feed	A feed rate setpoint

6.4.2 Development of the RBF Model

The choice of control and prediction horizons and the sample interval determines the dimension of the input vector to the RBF model. We used a sample interval as 3 minutes. The prediction and control horizons were chosen as one hour ($p=20$) and 15 minutes ($c=5$), respectively.

The prediction and control horizons affect the number of hidden nodes required by the RBF model. Typically, an increase in the input space size (due to large dimension) requires greater number of hidden nodes to span it. While increasing the number of nodes is a potential solution, it leads to a concomitant increase in model complexity and demands on the test and training set size.

Training data were generated by perturbing the NMPC manipulated variables around the base case. Table 6.3 gives the range of the perturbations. The magnitude of the steps was selected randomly within the ranges specified in Table 6.3. The duration of each step was selected randomly between 45 minutes and 2 hours. The simulator was run multiple times to generate the desired data. A total of 35 runs with simulation time varying between 16 hours and 175 hours were made. Each run was terminated when process conditions approached known operating constraints for the process (e.g., high pressure shutdown). All disturbances were turned off during each run.

Before generating network input patterns, the data were normalized to zero mean and unity standard deviation. The arrangement of measurements in each network input

pattern, x_k is shown in Table 6.4. Note that in addition to the control inputs, the input vector also contains measurements of D feed, E feed and the total reactor feed rates. Use of D feed rate and E feed rate as inputs to the RBF model was necessary since implementation of the product mix setpoint change entailed large manipulations of the D and E feed rate to the reactor. These manipulations in turn potentially influence variables controlled by the NMPC controller and therefore must be modeled by the RBF network.

Table 6.3

Operating region represented in network training.

MPC Manipulated Variable	Maximum value	Minimum value
Reactor temperature (°C)	116	128
Purge rate (kscmh)	0.05	0.85
A & C feed rate (kscmh)	7.5	11
A feed	0.05	1.05

Table 6.4

Elements of RBF input pattern vector x_k .

Delayed CVs	Pressure _{,k-p}	% B in purge _{,k-p}	Product rate _{,k-p}	A/C mole ratio _{,k-p}
Delayed MV	A feed _{,k-p-1}	A & C feed _{,k-p-1}	Purge rate _{,k-p-1}	Temperature _{,k-p-1}
Past input moves	ΔA feed _{,k-p} , ..., ΔA feed _{,k-1}	ΔC feed _{,k-p} , ..., ΔC feed _{,k-1}	Δ Purge _{,k-p} , ..., Δ Purge _{,k-1}	Δ Temp _{,k-p} , ..., Δ Temp _{,k-1}
Disturbance inputs	D feed rate _{,k}	E feed rate _{,k}	Reactor feed rate _{,k}	

A total of 10,000 network input patterns were selected by uniformly choosing from the entire set of approximately 40,000 available patterns. These covered the range of operating region summarized in Table 6.3. A 230 hidden node RBF network was

trained using 5000 input patterns. The network was then validated on the remaining 5000 patterns. Training and test set error statistics are provided in Table 6.5.

To illustrate RBF model predictions within and outside of training region, the following experiment was performed. At time zero, the A & C feed rate was stepped down from its base value of $9.3477 \text{ m}^3\text{h}^{-1}$ to $9.0 \text{ m}^3\text{h}^{-1}$ for a period of two hours. It was then decreased in steps of $0.5 \text{ m}^3\text{h}^{-1}$ at hourly intervals to $7.5 \text{ m}^3\text{h}^{-1}$. After five hours, the A & C feed rate was stepped to $6.0 \text{ m}^3\text{h}^{-1}$, which lies outside of the training data range (see Table 6.3). During this exercise, the A feed rate, purge rate and reactor temperature were maintained at their base values. A comparison the actual and predicted behavior is shown in Figure 6.3. As expected, good agreement is observed until after five hours when the model begins to operate outside the range of the training data. However, the model does continue to correctly predict long term trends.

Table 6.5

Training and test set error statistics.

		Reactor pressure, kPa	Product flow rate, m^3/hr	Component B in purge, mol %	(A/C) mole ratio in reactor feed
Training set	mean error	-0.0024	-0.0001	0.0000	0.0000
	Standard deviation of error	8.3387	0.1186	0.1038	0.0337
Test set	mean error	0.0042	0.0006	-0.0039	0.0008
	standard deviation of error	8.7749	0.1207	0.1089	0.0331

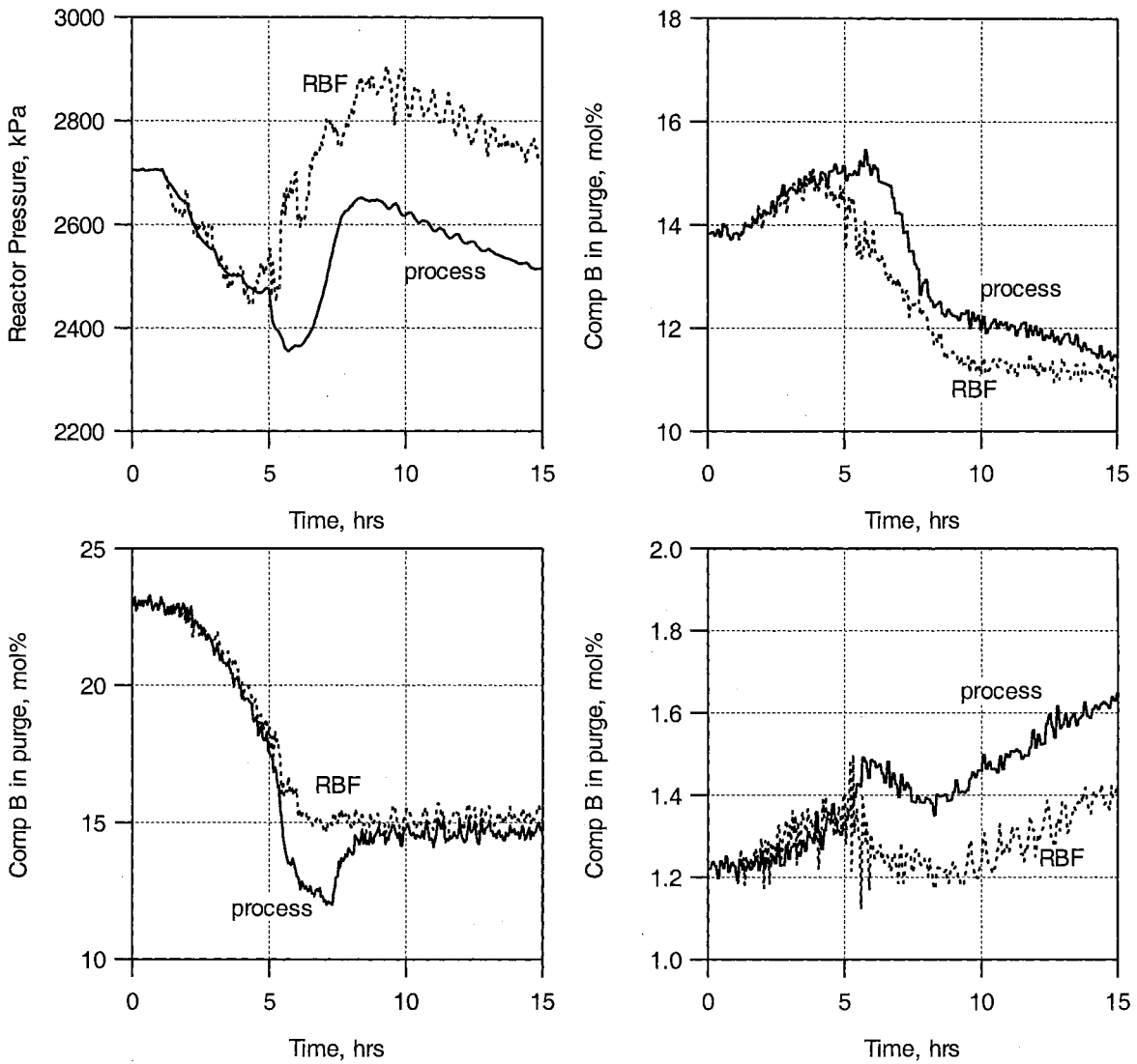


Figure 6.3: Comparison of RBF model predictions with plant measurements for step changes in A & C feed rate. All other variables are maintained at their base values. After 5 hours, the A & C feed rate is brought to $6 \text{ m}^3/\text{h}$, which lies outside the lower limit of training data of $7.5 \text{ m}^3/\text{h}$.

6.4.3 NMPC Controller Settings

Table 6.6 shows the error penalty and move suppression factors employed in the current work. Considerations used in the choice of the weighting factors are as follows:

- 1) Large deviations in A/C mole ratio in the reactor feed from the base case value of 1.22 can potentially lead to plant shutdown due to violation of pressure limits. Thus, A/C mole ratio must be controlled tightly.
- 2) The product rate must be kept near its setpoint.
- 3) No tolerance limits are specified on reactor pressure and purge gas composition of component B. Thus, smaller penalties can be applied to lower priority for their control.
- 4) Downs and Vogel suggest changes in flow rate of the A & C stream (stream 4) are undesirable.
- 5) Large changes in purge rate can change the reactor feed rate (by changing the recycle rate) leading to upsets in the production rate.

Table 6.6

Weights used in the RBF based MPC simulations.

Controlled Variable	Error penalty	Manipulated variable	Move suppression
Reactor pressure	0.16	A feed setpoint	0.7
Product flow rate	0.42	C feed setpoint	1.0
Component B in purge	0.28	Purge rate	0.8
A/C mole ratio in reactor feed	0.35	Reactor temperature	1.4

Hard constraints were implemented only on the manipulated variables. Table 6.3 documents the constraints used.

6.5 NMPC Controller Performance

In all simulation results reported below, setpoint changes or disturbances were implemented after two hours of simulation at the base case conditions. Further, all flow and pressure measurements were filtered using a CUSUM filter (Rhinehart, 1992) prior to input to the NMPC scheme. In their paper, Downs and Vogel (1993) suggest the following setpoint changes and disturbances to evaluate the control scheme.

- 1) Production rate (step change -15%)
- 2) Product mix (step change from 50G/50H to 40G/60H)
- 3) Pressure change (step change -60 kPa)
- 4) Composition of B in purge (step change 2%)
- 5) IDV(1) (step change A/C feed ratio in stream 4)
- 6) IDV(4) (step change reactor cooling water inlet temperature)
- 7) IDV(8) (random variation A, B, C feed composition in stream 4)
- 8) IDV(12) + IDV(15) (simultaneous random variation of condenser cooling water inlet temperature and valve sticking).

6.5.1 Servo Response

Results for setpoint changes are presented in Figures 6.4 to 6.7. The product mix setpoint was implemented through the PI controller that used the D/E feed ratio as the manipulated variable. D and E feed measurements were also used as inputs to the RBF network model. All four of the setpoint changes reflected operation in region represented in the training data. Consequently, the RBF model provided good predictions and tight control was achieved.

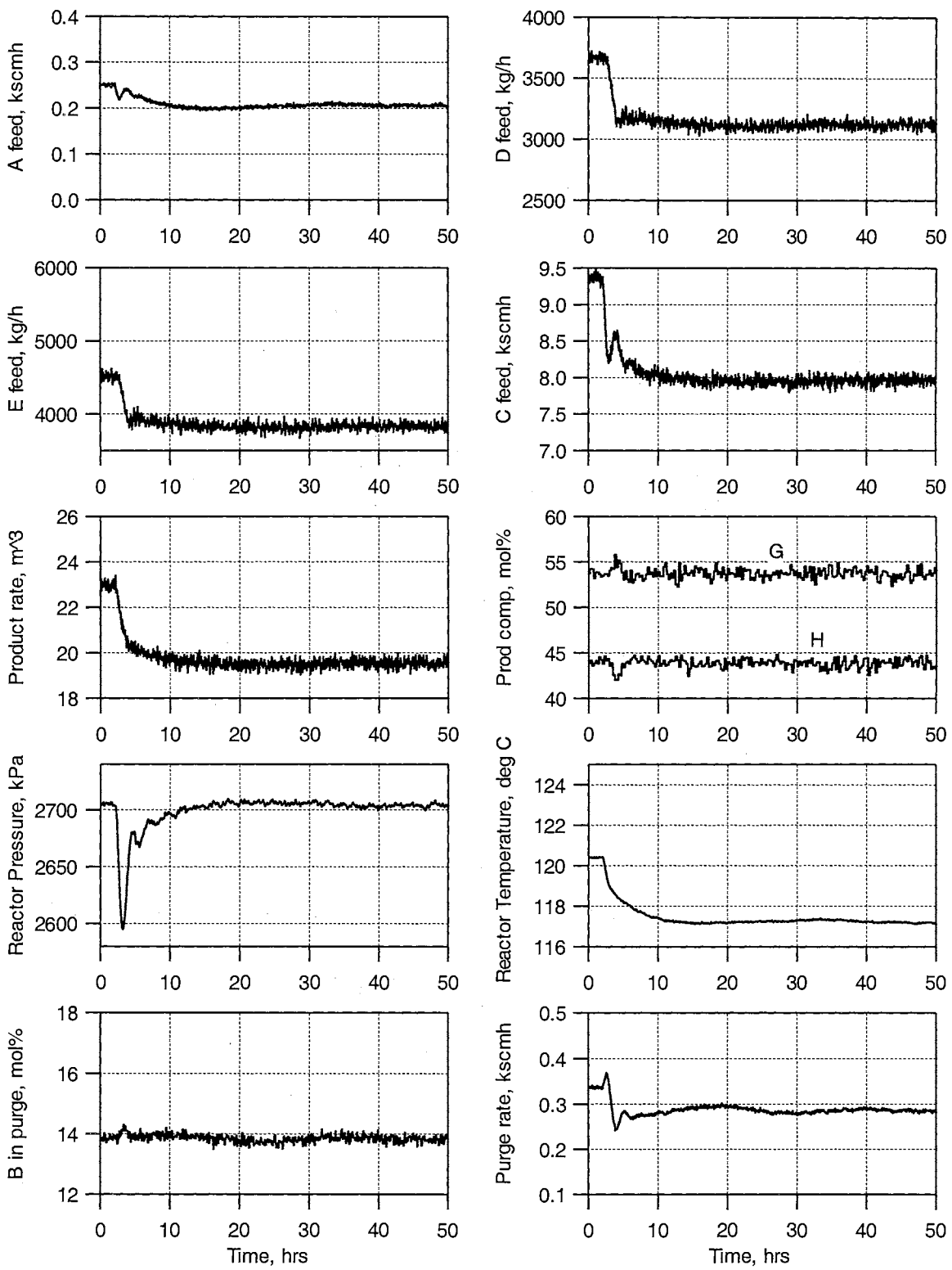


Figure 6.4: Product flowrate setpoint change (-15%).

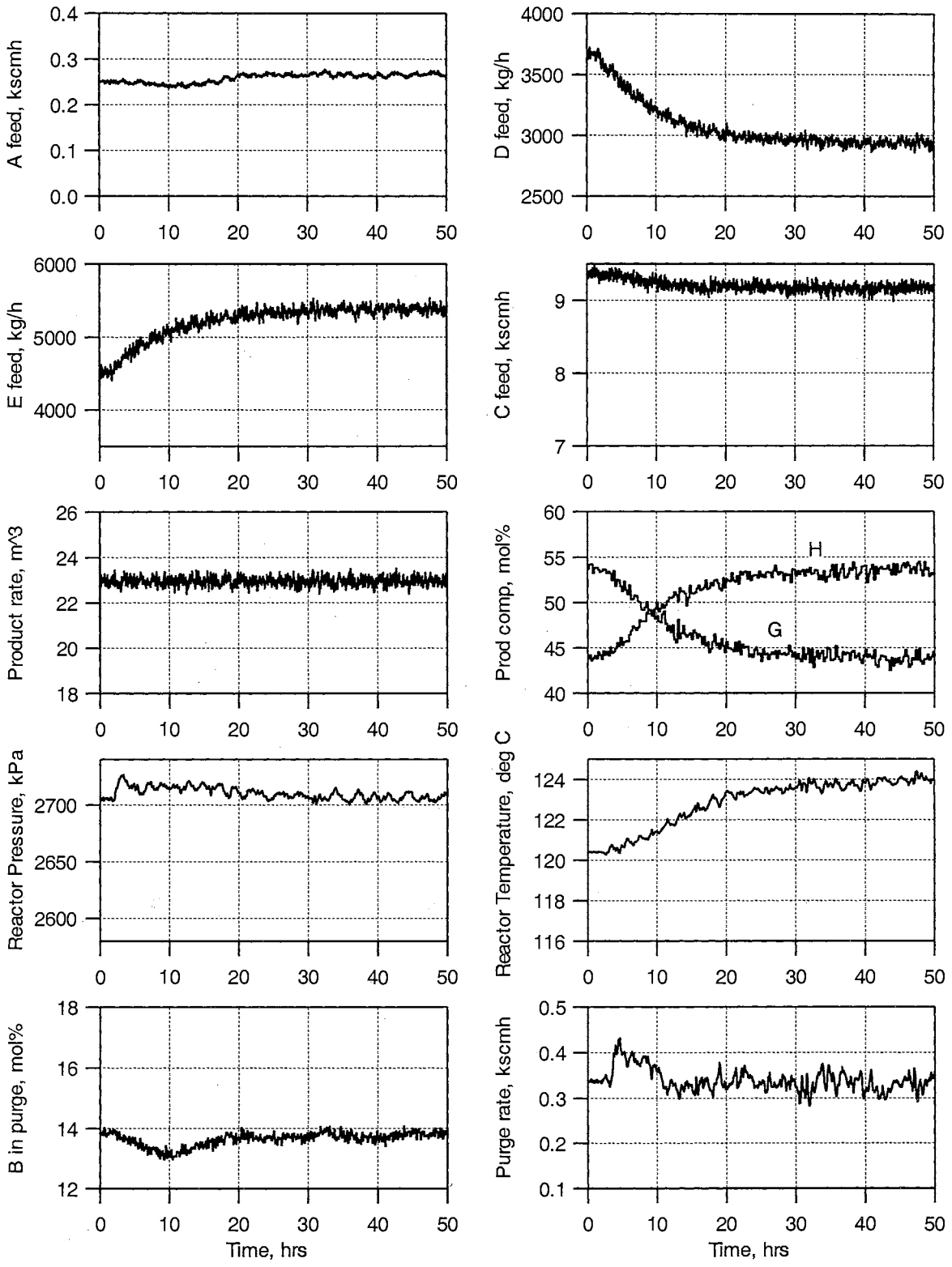


Figure 6.5: Product G/H ratio setpoint change from 50/50 to 40/60.

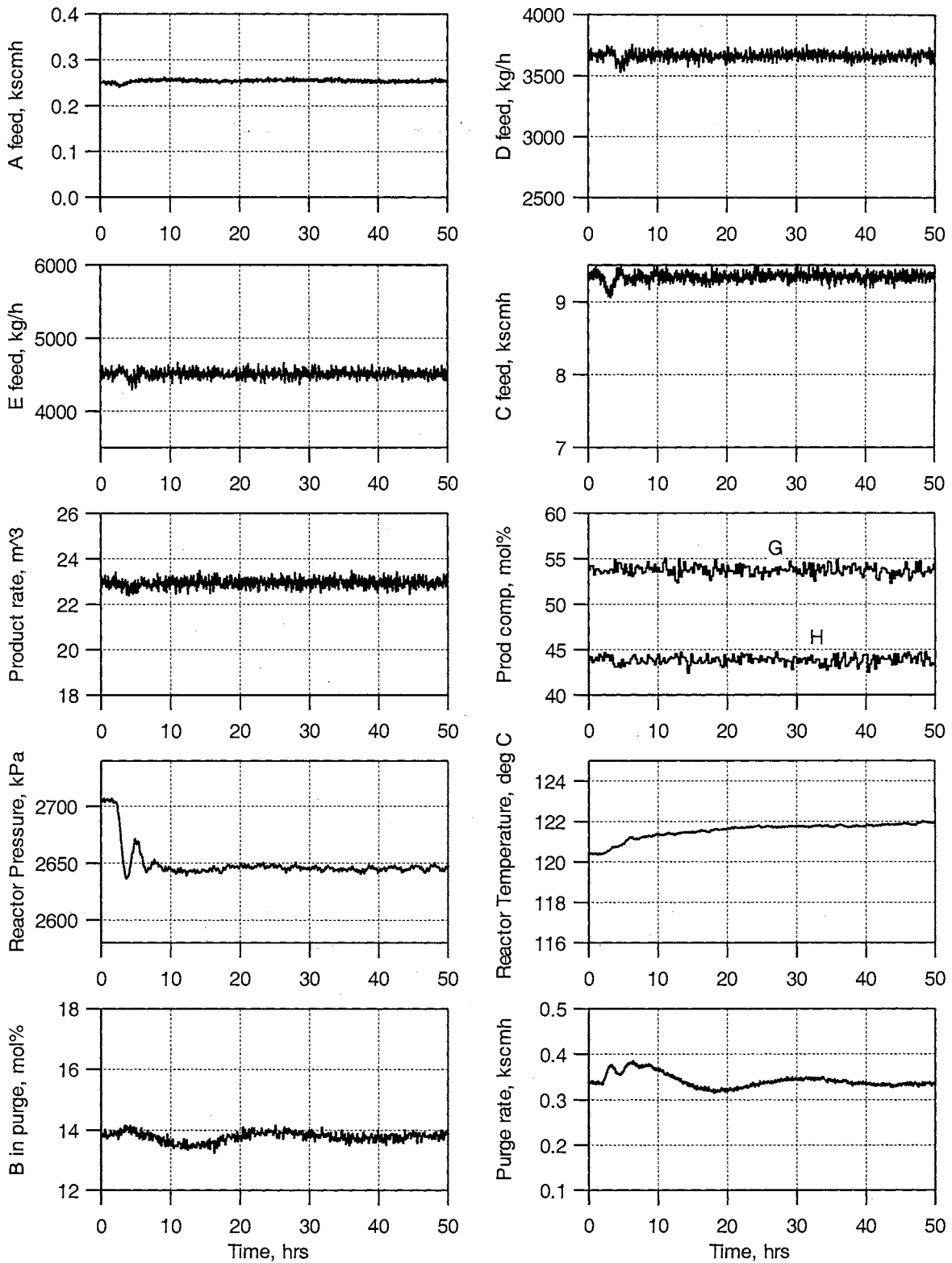


Figure 6.6: Reactor pressure setpoint change (-60 kPa).

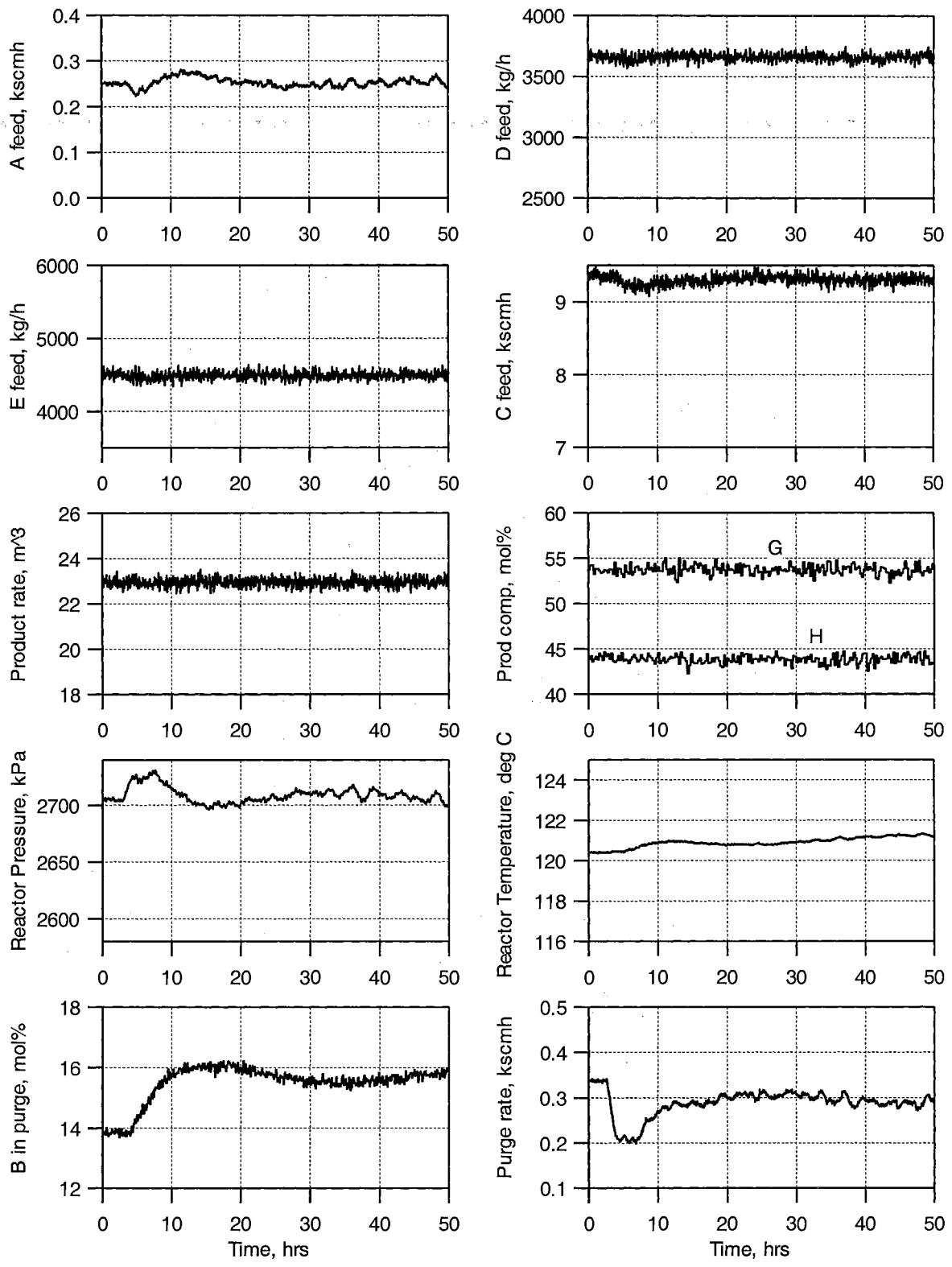


Figure 6.7: Purge B composition setpoint change (+2 mol%).

Figure 6.8 provides an illustration of the mismatch between the plant measurements and the RBF model predictions for the setpoint change in purge gas composition of component B. The result is typical for the other setpoint changes as well. The relatively small mismatches observed in the setpoint tracking confirm the accuracy of the RBF model predictions. These results were expected since the data used for training of the RBF network covered the necessary ranges represented in these simulations (in absence of any disturbance).

6.5.2 Regulatory Response

Disturbances IDV(4) and IDV(12+15) relate to cooling water of the reactor and condenser, respectively, and are rejected by the inner loops of the cascade structure. Since the NMPC subset is not affected, no appreciable transients were observed and the results have been omitted. Among the other disturbances, we found those that upset the A to C ratio in the reactor feed (i.e., IDV(1) and IDV(8)) were most difficult to control. In such instances, the unmeasured disturbances, which are not incorporated in the RBF model, lead to inaccurate model predictions. Results for IDV(1) (step change in A/C feed ratio in stream 4) and the corresponding process-model mismatch are shown in Figures 6.9 and 6.10, respectively. The unmeasured step disturbance is first detected by a drop in a controlled variable, the A/C molal ratio in the reactor feed (see Figure 6.11). Consequently, the A feed rate is increased to bring A/C ratio in reactor feed to its setpoint. Since IDV(1) was a step disturbance, the process-model mismatch ultimately reaches steady-state values (see Figure 6.10). The NMPC controller employing the traditional additive disturbance bias to account for model mismatch provides reasonably

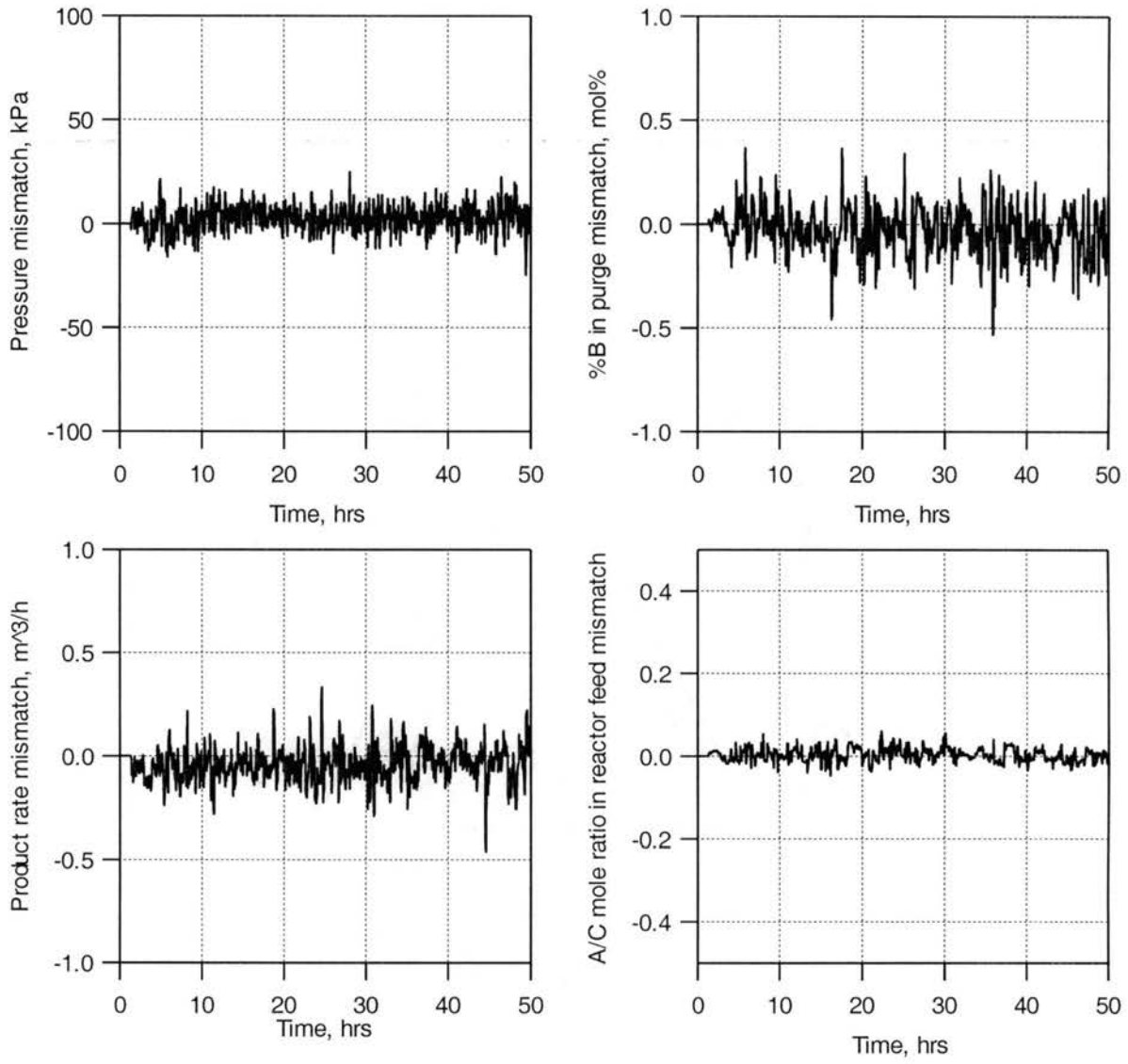


Figure 6.8: Process-model mismatch during implementation of +2% setpoint change in composition of component B in purge stream.

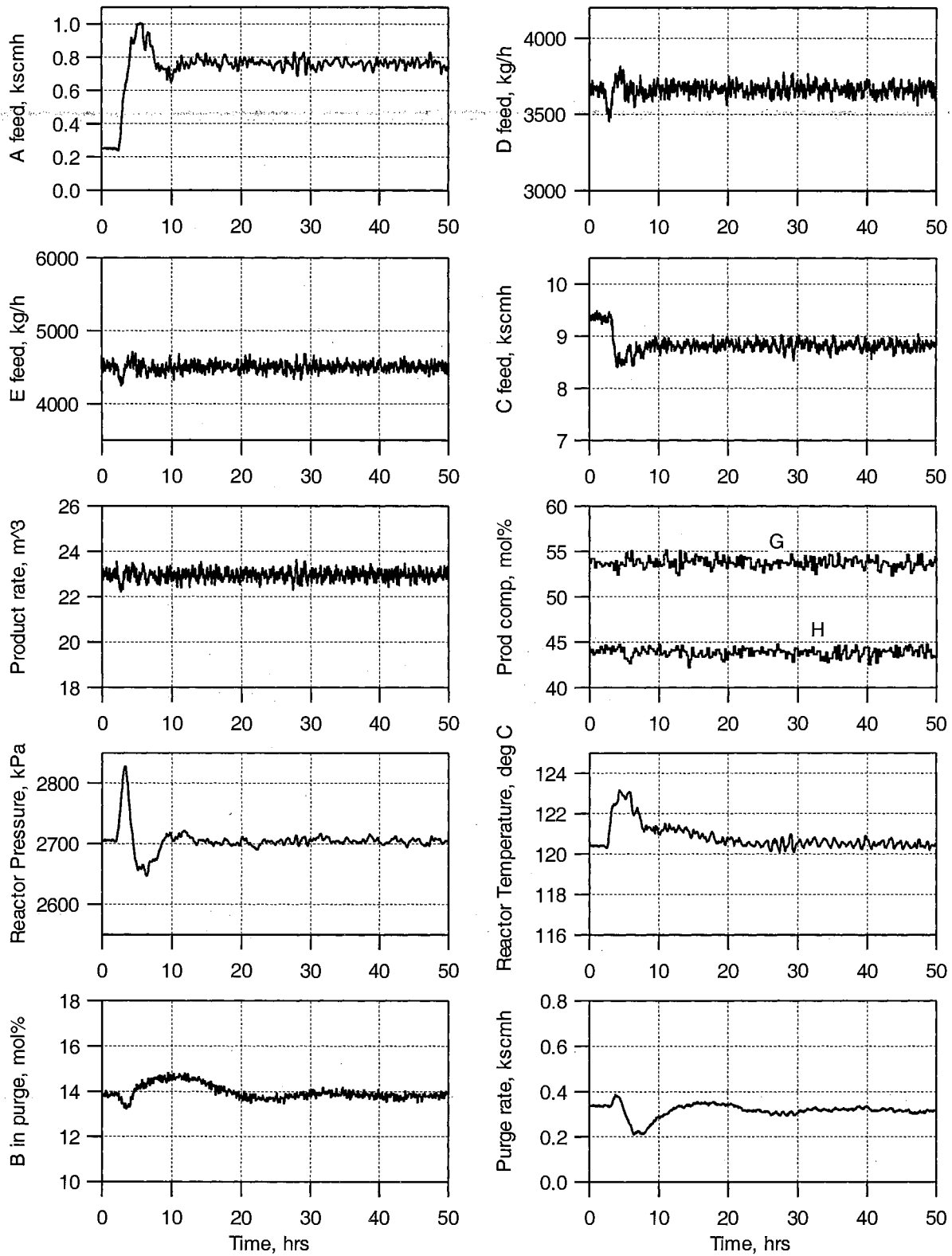


Figure 6.9: Disturbance IDV(1) (step change in A/C feed ratio in A & C stream)

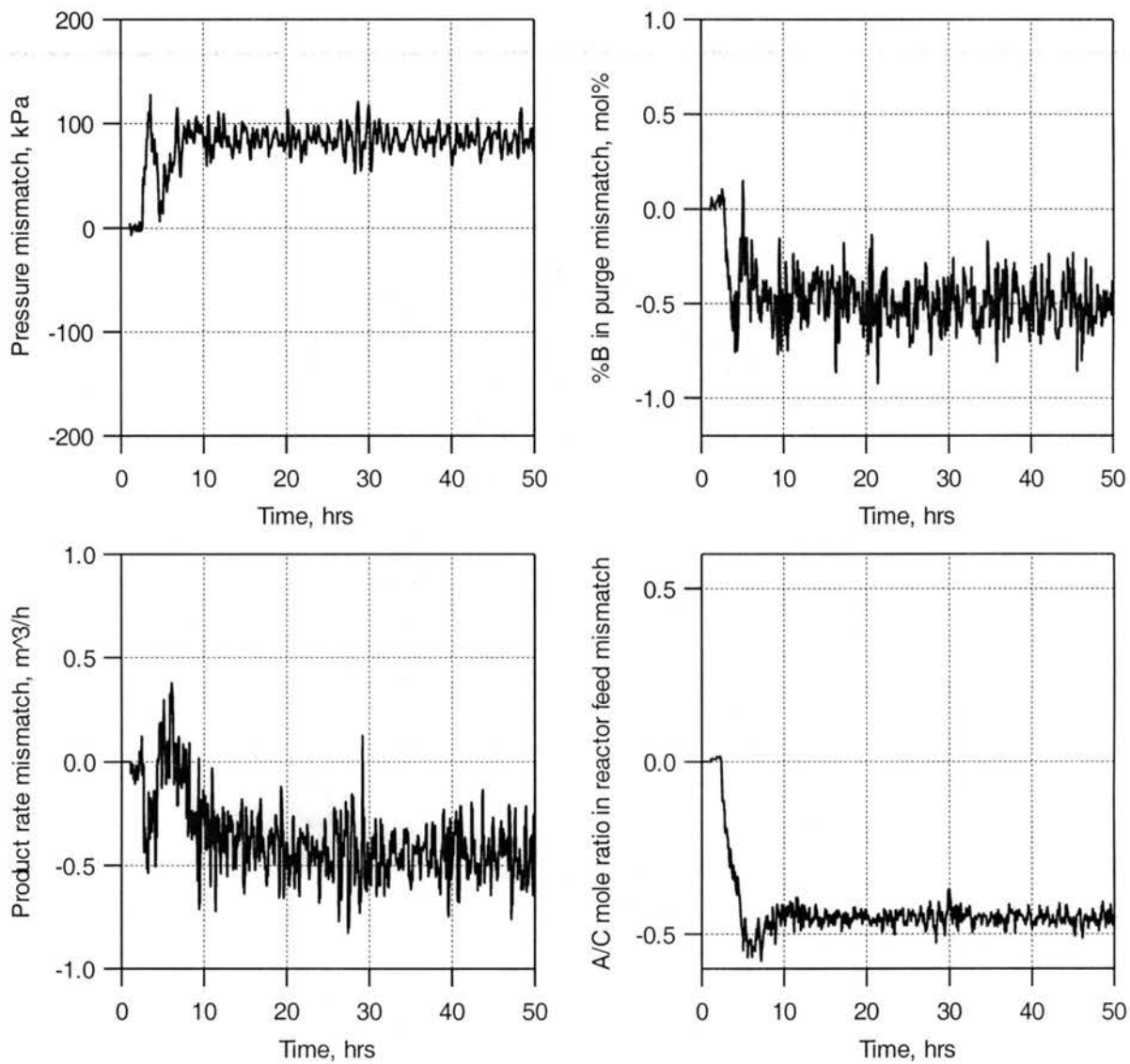


Figure 6.10: Process-model mismatch during occurrence of disturbance IDV(1). The unmeasured disturbance is not modeled by the RBF network, leading to poor predictions. However, due to the step nature of the disturbance, the process-model mismatches settle to near-steady values.

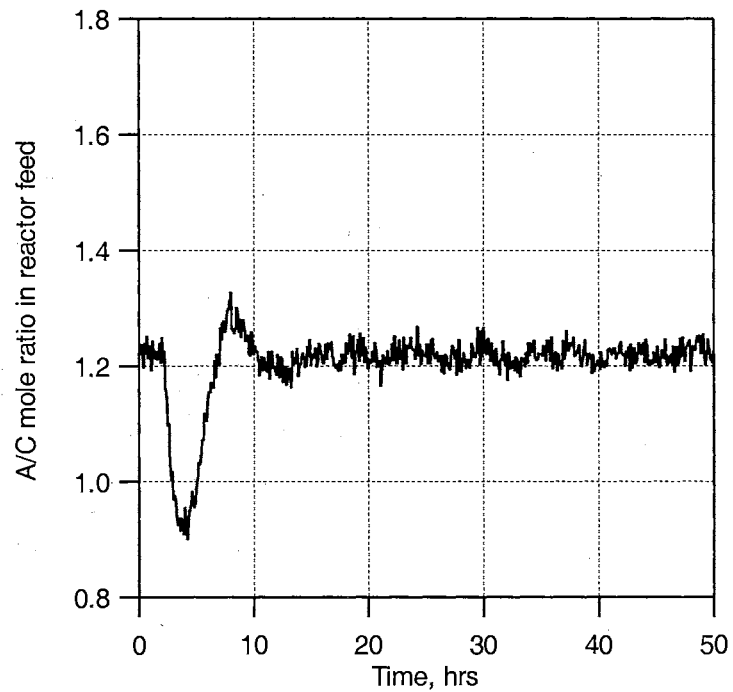


Figure 6.11: The step disturbance IDV(1) is detected by the drop in the controlled variable, A/C ratio in reactor feed, from its setpoint. The variable is then brought back to its setpoint by increasing A feed rate.

good control under this circumstance.

However, in presence of IDV(8) (random variations in A, B and C composition in stream 4), model mismatch no longer behaves as a step disturbance at the output. Consequently, poorer control was observed (Figure 6.12). The random mismatch between the model predictions and plant measurements is apparent from Figure 6.13. The observed degradation in control was expected and is common to all MPC algorithms that use a constant bias to account for model mismatch (Lundstrom *et al.*, 1995). Observer based MPC algorithms (Lee *et al.*, 1994; Ricker, 1990) represent one possible solution to this problem. Nonlinear MPC using closed-loop state estimation by an extended Kalman filter has been proposed by Lee and Ricker (1994). In their MPC strategy for control of the Eastman process, Ricker and Lee (1995a) use an extended Kalman filter to estimate unmeasured disturbance states. Their results also indicate excellent control in presence of disturbance IDV(8). While state estimation has not been pursued in the current work, it may be possible to linearize the RBF model at every control interval followed by a realization of a state-space model and use of state estimation techniques.

6.5.3 Computational Requirements

The MATLAB implementation of the Eastman process developed by Prof. N.L. Ricker was used for all simulations. During each control step, $\hat{\mathbf{y}}_{p,k+i}$ was calculated only once. The MATLAB function *constr* was used to perform the constrained nonlinear

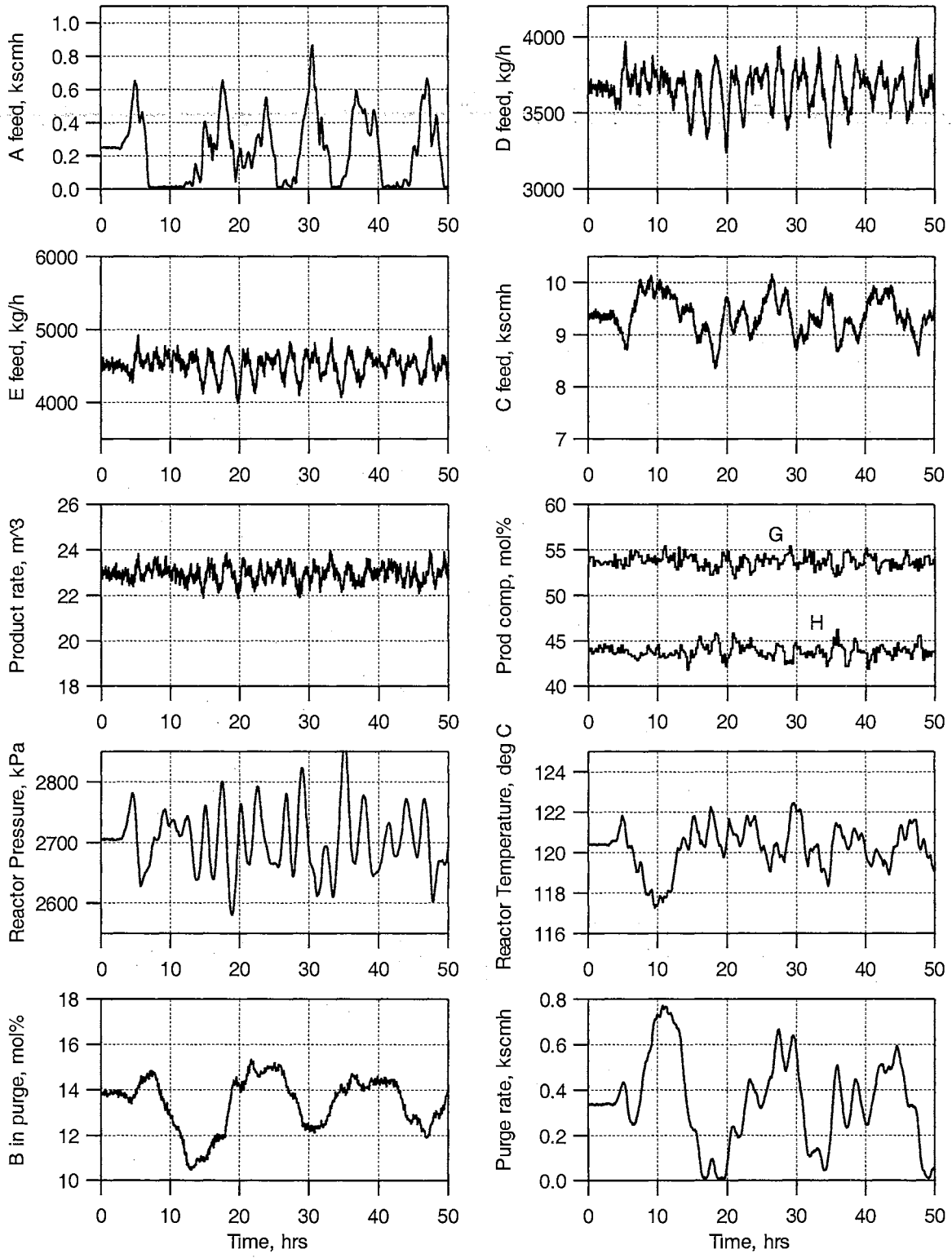


Figure 6.12: Disturbance IDV(8) (random variation in A, B, C compositions in A & C stream)

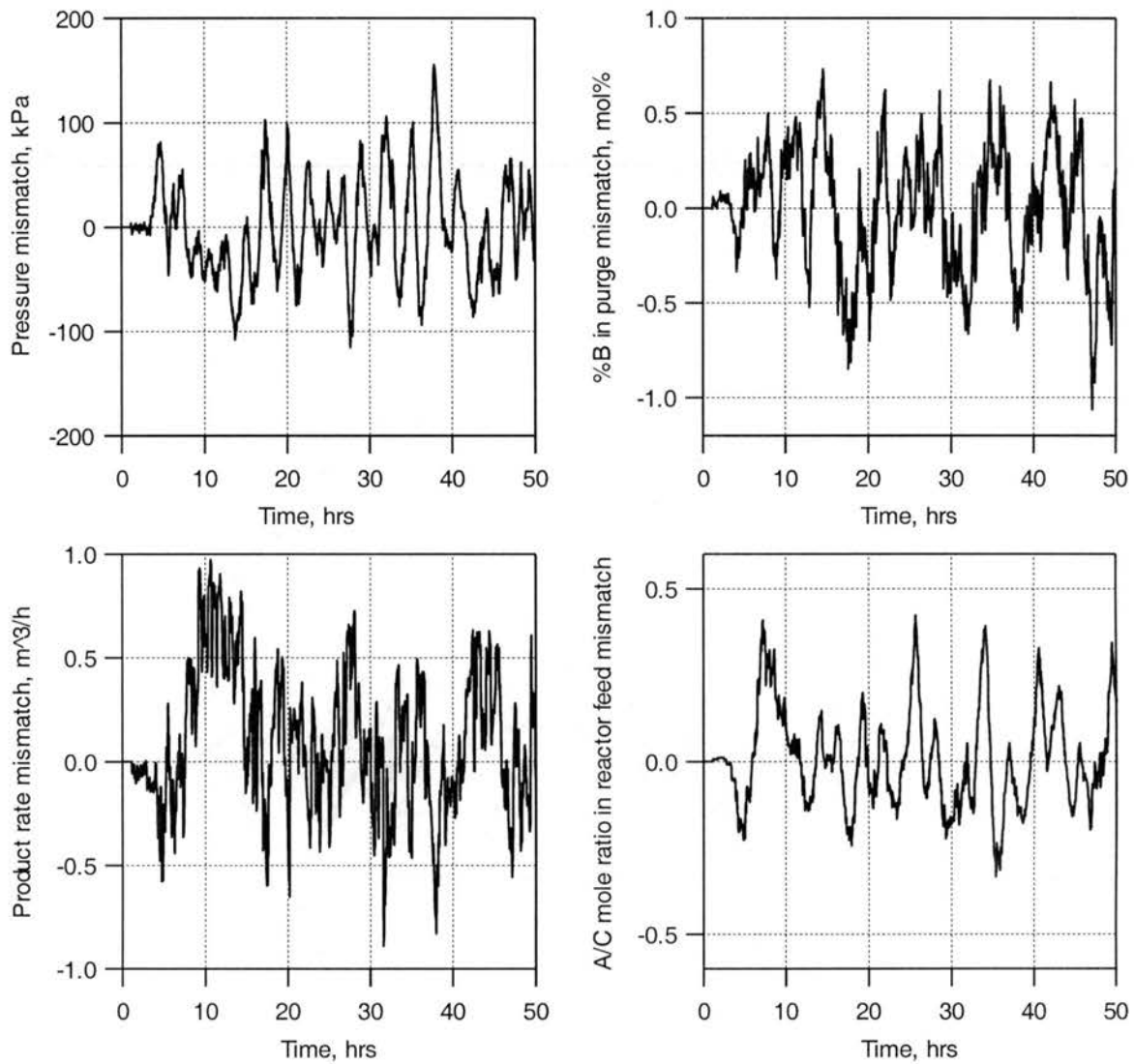


Figure 6.13: Process-model mismatch during occurrence of disturbance IDV(8). The unmeasured disturbance is not modeled by the RBF network, leading to poor predictions. The random nature of the disturbance leads to random variations in process-model mismatch.

minimization via sequential quadratic programming with active set method for constraint handling.

Table 6.7 documents the time needed to compute the control relevant instructions for the proposed factorized RBF based NMPC and a non-factorized approach for implementation of the pressure setpoint change (Figure 6.6). In the non-factorized approach, gradients were computed numerically and the future predictions involved computation of \hat{y}_{k+i} at each function call. The factorized approach used analytical gradient expressions and needed computation of only one factor, $\hat{y}_{t,k+i}$ (see equation (6.9)), at every function call during optimization since $\hat{y}_{p,k+i}$ contains past measurements and moves, and remains unchanged for a given control step. The computational results presented in Table 6.7 were calculated using the *tic-toc* commands in MATLAB and are typical of the 50 hr. simulation results presented in this paper using a 550 MHz Windows operating system.

Table 6.7

Comparison of computation time needed for implementation of a setpoint change of -60 kPa in reactor pressure (Figure 6.6).

(simulation for 1000 samples or 50 hours)

	Factorized RBF based NMPC	Non-factorized RBF based NMPC (gradients evaluated numerically)
Real-time needed for computation (hours)	1.82	38.31

As evident from Table 6.7, the factorized formulation of the RBF model NMPC scheme is significantly (2+ orders of magnitude) more efficient than a non-factorized form using numerical approximations for the required gradient and Hessian information. For the factorized formulation, the number of function and gradients evaluations that were needed for the optimization to converge at each control execution ranged from 3 to 15. For the non-factorized, non-analytical form, the number of iterations frequently reached the user specified upper bound of 600.

6.5.4 Discussion of Results

The results presented are consistent with expectations. RBF models can provide good approximation of any nonlinear system. Hence, an NMPC scheme that employs an RBF model would be expected to provide good control whenever the plant operated in regions used to train the model. Practical implementation of an RBF (or any empirical model) based MPC scheme would require a watchdog to verify that operation lies within model development bounds. The problem/concept is the same as required for the linear MPC systems currently employed in industry. However, the level of concern is greater for an RBF based system due to the less certain (compared to a linear impulse or step response model) extrapolation characteristics.

Another issue affecting practical implementation of an RBF based NMPC scheme is collection of the plant data necessary to develop the model. The amount of data required to develop a nonlinear model using an RBF or other type of neural network is orders of magnitude greater than required for a simple linear model. However, the time

and investment required to develop the linear models used with today's linear MPC systems is already an impediment in many cases. Significant economic benefits would be required for management to authorize plant tests to collect the amount of data used to produce the results presented in this paper. Nevertheless, the potential improvement in performance using nonlinear control methods provides the incentive to find new ways to develop robust RBF models from existing closed-loop plant data or other means.

6.6. Conclusions

Nonlinear model predictive control of the Eastman process using a factorized radial basis function network model is presented. The results demonstrate applicability of the technique on an industrial scale. The salient feature of the factorized approach lies in the ability to express model future predictions as an inner product of two vectors. One of the vectors contains the decision variables of the MPC optimization program while the other consists entirely of known past quantities. Thus, computational effort is minimized, since only the unknown parts of the objective function need to be re-evaluated during optimization at a given control execution step. The results presented confirm the computational efficiency of the proposed NMPC algorithm.

In the current work, the RBF network was trained using data from the Eastman plant simulator, corresponding to operation of plant in absence of disturbances. Consequently, tight control was obtained for the setpoint changes suggested by Downs and Vogel. Disturbances that upset the reaction stoichiometry were found to be more

difficult to control. From a practical standpoint, the most pressing issues involve development of the nonlinear RBF model.

7.1 Conclusions

The work described in previous chapters was developed along two lines of thought:

- 1) online update of models in linear MPC to reflect current operating conditions, and
- 2) use of neural network models for soft sensing and nonlinear MPC.

The potential benefits were demonstrated through use of simulation examples and real data. The salient conclusions that can be drawn from the documented work are presented below.

- Model adaptation is useful in improving control performance in cases where large errors in model parameters contribute to plant-model mismatch (Chapter 3).
- The adaptive QDMC algorithm depicted in Figure 3.3 has been designed such that it can be integrated with existing control software with minimal computational burden owing to efficiency of the recursive least squares algorithm.
- For a SISO process controlled by an adaptive DMC controller (section 3.4.1), the closed-loop system is identifiable, provided the data are sufficiently exciting.
- While neural networks (multi-layer perceptron) serve as a vehicle for constructing correlations, substantial effort must be invested in data preparation and variable selection for a successful application (Chapter 4).
- In the petroleum refinery example under study, the endpoint of kerosene could be predicted within an error standard deviation of 1.7 °F using the proposed methodology for inferential measurements.

- The novel nonlinear MPC (NMPC) algorithm using radial basis function (RBF) networks (Chapter 5) uses factorization of Gaussian functions to provide a computationally efficient method for control of nonlinear processes.
- The NMPC algorithm takes a generic form and relies only on the parameters and not on the mechanics of the particular process in question. This is similar to DMC/QDMC implementations where the processes are described in terms of the step response coefficients allowing development of generic software.
- Successful application of the RBF based NMPC algorithm to the Eastman challenge problem demonstrates the potential of the proposed technique to large problems.

The main theme in all of the above work consisted of incorporating process knowledge in control applications and follows the adage, "the more accurately we can predict the future process behavior, the better will be the chance of controlling it."

7.2 Recommendations

A number of opportunities exist for improving upon the present work. These are listed below.

7.2.1 Adaptive QDMC

The adaptive QDMC algorithm discussed in Chapter 3 attempts to provide an adaptive feature to linear MPC similar to self-tuning in commercially available PID-type controllers. The simulation examples described the potential of such a scheme for linear and nonlinear systems. However, certain points need to be further explored.

- a) The adapted model parameters may be biased in presence of non-white noise in measurements. One approach to this problem lies in development of suitable filters. Such filters may be based on noise models such as the Box and Jenkins transfer function model (Box and Jenkins, 1994).

- b) Further work also needs to be done on incorporating unmeasured process disturbance models (as opposed to noise). MPC handles all sources of plant-model mismatch as an additive step disturbance at the output. It has been shown by Lundstrom et al. (1995) by an example that such an approach leads to poor control by MPC in presence of ramp-like disturbances. To obviate the need for the assumption of step disturbance at the output, observer based MPC algorithms have been proposed in literature (Lee et al., 1994; Ricker, 1990). Benefits of observers with adaptive QDMC need to be further explored.

- c) Continuous parameter estimation can lead to parameter drift as the estimation algorithm tries to estimate parameters such that the error between the plant response and model prediction is minimized. One solution to avoid parameter drift is the dead zone approach. Here, the identification algorithm is stopped when the signals are not sufficiently excited to guarantee model improvement. In Chapter 3, the trigger for model adaptation was based on comparison of magnitudes of the variations in inputs and outputs with predetermined threshold values was employed. However, such a technique may be inappropriate if the

signal-to-noise ratio is small. In such situations, appropriate adaptation triggers need to be developed.

7.2 Neural network models for forecasting and control

- (a) Simplified theoretical models attempt to capture the essence of the underlying phenomena. Thus, such models provide insights into the behavior of the process. Further, since these models are constrained by the description of a physical phenomenon, less process data are required in their development. However, a concomitant disadvantage, is their inaccuracy when applied to real-world situations. This may be attributed to inappropriate model assumptions. Development of neural network models, on the other hand, needs a large amount of data. Moreover, there is no guarantee that these models satisfy material and energy balances, a fundamental consideration in process operation. However, if appropriate data are available, good predictions can be obtained. Thus, there exists an opportunity to integrate theoretical and neural network models to provide hybrid models. Work on these lines by Gurumoorthy and Kosanovich (1998) can be directly applied to the factorized RBF model based NMPC.

- (b) Data for neural network training in Chapter 5 and Chapter 6 was generated by applying random inputs to the process simulator and recording the response. Such an approach is based on excitation of the process and does not make use of any criteria that link prediction capabilities of networks to the data structure.

Development of such criteria will aid in designing appropriate test signal designs for system identification.

- (c) The RBF model used in Chapter 5 and Chapter 6 use input patterns that consist of process outputs, process inputs and past input moves. It is of interest to design tests that detect the distance between the input training patterns in the cluster space. This will also aid in obtaining data for training of the network which adequately spans the input pattern space in addition to the region of process operation.

- (d) A primary concern with neural network models is their validity when operating in regions not represented in the training data. This will often be the case if the no process data exist for certain regions or for a new process. Based on the work in Chapter 5 and Chapter 6, it is noted that the neural network model performance is generally poor in such regions. A possible improvement of the factorized RBF model based NMPC in such situations is use of weight adaptation. The network response is linear in the weight parameters and it may be possible to use the standard recursive least squares similar as in the adaptive QDMC work. Here, as new data become available (which are representative of the training set), the weights would be adapted to reflect the new operating region. A similar idea may be pursued with the number of nodes and their location. While algorithms exist in literature that adaptively increase or decrease the size of the network, (and hence

the number of centers and their widths), these tend to add a huge computational load and may not be of use for online applications.

BIBLIOGRAPHY

- Agarwal, M., and D.E. Seborg, "Self-tuning controllers for nonlinear systems," *Automatica* **23**, 204 (1987).
- Akaike, H., "A new look at the statistical model identification," *IEEE Transactions on Automatic Control* **AC-19**, 716-723 (1974).
- Allen, D.M., "Mean square error of prediction as a criterion for selecting variables," *Technometrics* **13**, 469-475 (1971).
- Astrom, K.J., and B. Wittenmark, *Adaptive control*, Addison Wesley, (1995).
- Badgwell, T.A., "Robust model predictive control of stable linear systems," *International Journal of Control* **68**, 797-818 (1997).
- Banerjee, A., and Y. Arkun, "Control configuration design applied to the Tennessee Eastman plant-wide control problem," *Computers chem. Engng.* **19**, 453-480 (1995).
- Banerjee, A., Y. Arkun, B. Ogunnaike, and R. Pearson, "Estimation of nonlinear systems using linear multiple models," *AIChE Journal* **43**, 1204-1226 (1997).
- Barnett, V., *Outliers in Statistical Data*, Wiley & Sons, NY, (1994).
- Benallou, A., D. E. Seborg, and D.A. Mellichamp, "Dynamic compartmental models for separation processes," *AIChE J.* **32**, 1067-1078 (1986).
- Bequette, B.W., "Nonlinear control of chemical processes: A review," *Industrial and Engineering Chemistry Research* **30**, 1391 (1991).
- Bhartiya, S., and J.R. Whiteley, "Nonlinear Model Predictive Control Using a Radial Basis Function Model," *AIChE Journal* (**submitted**),
- Bhat, N.V., and T.J. McAvoy, "Use of neural nets for dynamic modeling and control of chemical process systems," *Comput. Chem. Engng.* **14**, 573 (1990).
- Bhide, V. M., M.J. Piovoso, and K.A. Kosanovich, "Statistics on the reliability of neural network estimates," *American Control Conference*, Vol. 1877 (1995).
- Box, G.E.P., and G.M. Jenkins, *Time series analysis: forecasting and control*, Prentice-Hall, Englewood Cliffs, NJ (1994).

Bozin, A.S., and P.C. Austin, "Dynamic matrix control of a paper machine benchmark problem," *Conference on control systems in the pulp and paper industry*, 242-248 (1995).

Brogan, W., *Modern control theory*, Prentice-Hall, Englewood Cliffs, NJ (1991).

Brusch, R.G., "A nonlinear programming approach to space shuttle trajectory optimization," *Journal of Optimization Theory and Applications* **13**, 94-118 (1974).

Chen, S., S. Billings, C. Cowen, and P. Gren, "Practical identification of NARMAX models using radial basis functions," *Int. J. Control* **52**, 1327-1350 (1990).

Chu, X., and D.E. Seborg, "Nonlinear model predictive control based on Hammerstein models," *International Symposium on Process Systems Engineering*, Seoul, Korea, Vol. 995 (1994).

Clarke, D.W., "Application of generalized predictive control," *IFAC Adaptive Control of Chemical Processes*, Copenhagen, Denmark, Vol. (1988).

Clarke, D.W., and C. Mohtadi, "Properties of generalized predictive control," *Automatica* **25**, 859-875 (1989).

Clarke, D.W., C. Mohtadi, and P.S. Tuffs, "Generalized predictive control - Part I. The basic algorithm," *Automatica* **23**, 137-148 (1987).

Clarke, D.W., D.E. Mosca, and R. Scattaloni, "Robustness of an adaptive predictive controller," *30th IEEE Conference on Decision and Control*, Brighton, England, Vol. 979-984 (1991).

Cook, P.A., *Nonlinear dynamical systems*, Prentice hall, NJ, (1986).

Cook, R.D., and S. Weisberg, "Characterizations of an empirical influence function for detecting influential cases in regression," *Technometrics* **22**, 495-508 (1980).

Cutler, C.R., and R.B. Hawkins, "Constrained multivariable control of a hydrocracker reactor," *American Control Conference*, Minneapolis, Minnesota, Vol. 1014-1020 (1987).

Cutler, C.R., and B.L. Ramaker, "Dynamic matrix control - a computer control algorithm," *AIChE National Meeting*, Houston, TX, Vol. (1979).

Cybenko, G., "Approximation by superposition of a sigmoidal function," *Math. Control Signal Syst.* **2**, 303 (1987).

- De Keyser, R.M.C., P G A. De Van De Velde, and F.A.G. Dumortier, "A comparative study of self-adaptive long-range predictive control methods," *Automatica* **24**, 149-163 (1988).
- Desai, A.P., and D. Rivera, "Controller structure selection and system identification for the Tennessee Eastman challenge problem via intelligent process control," *1993 Annual AIChE Meeting*, St. Louis, Vol. Paper 148a (1993).
- Dion, J.M., L. Dugard, A. Franco, N.M. Tri, and D. Rey, "MIMO adaptive constrained predictive control case study: ' An environment test chamber'," *Automatica* **27**, 611-626 (1991).
- Dollar, R., "Consider adaptive multivariable predictive controllers," *Hydrocarbon Processing* March 1993, 109-112 (1993).
- Downs, J.J., and E.F. Vogel, "A plant-wide industrial process control problem," *Computers chem. Engng.* **17**, 245-255 (1993).
- Duchene, P., and P. Rouchon, "Kinetic scheme reduction via geometric singular perturbation techniques," *Chemical Engineering Science* **51**, 4661-4672 (1996).
- Eaton, J.W., and J.B. Rawlings, "Model-predictive control of chemical processes," *Chemical Engineering Science* **47**, 705-720 (1992).
- Emmanouilides, C., and L. Petrou, "Identification and control of anaerobic digesters using adaptive, online trained neural networks," *Computers & Chemical Engineering* **21**, 113-143 (1997).
- Fletcher, R., *Practical Methods of Optimization*, John Wiley & Sons, NY (1987).
- Froisy, B.J., "Model predictive control: past, present and future," *ISA Transactions* **33**, 235-243 (1994).
- Garcia, C.E., "Quadratic/dynamic matrix control of nonlinear processes: An application to batch reaction process," *AIChE annual meeting*, San Francisco, CA, Vol. (1984).
- Garcia, C.E., and M. Morari, "Internal model control. 1. A unifying review and some new results," *Ind. Eng. Chem. Process Des. Dev.* **21**, 308-323 (1982).
- Garcia, C.E., and M. Morari, "Internal model control. Multivariable control law computation and tuning guidelines," *Ind. Eng. Chem. Process Des. Dev.* **24**, 484-494 (1985).
- Garcia, C.E., and A.M. Morshedi, "Quadratic dynamic matrix control," *Chemical Engineering Commun.* **46**, 73-87 (1986).

- Garcia, C.E., D.M. Prett, and M. Morari, "Model predictive control: Theory and practice - a survey," *Automatica* **25**, 335-348 (1989).
- Gattu, G., and E. Zafiriou, "Observer based nonlinear quadratic dynamic matrix control for state space and input/output models," *The Canadian Journal of Chemical Engineering* **73**, 883-895 (1995).
- German, S., E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural Computation* **4**, 1-58 (1992).
- Gokhale, V., S. Horowitz, and J.B. Riggs, "A comparison of advanced distillation control techniques for a propylene-propane splitter," *Industrial Engineering Chemistry Research* **34**, 4413-4419 (1995).
- Gurumoorthy, A., and K.A. Kosanovich, "Improving the prediction capability of radial basis function networks," *Ind. Eng. Chem. Res.* **37**, 3956-3970 (1998).
- Gustavsson, I., T. Ljung, and T. Soderstrom, "Survey paper - identification of processes in closed loop - Identifiability and accuracy aspects," *Automatica* **13**, 59 (1977).
- Hair, J.F., R.E. Anderson, R.L. Tatham, and W.C. Black, *Multivariate Data Analysis*, Prentice Hall, NJ, (1995).
- Hartman, E.J., J.D. Keeler, and J.M. Kowalski, "Layered neural networks with Gaussian hidden units as universal approximations," *Neural Computation* **2**, 210-215 (1990).
- Haykin, S., *Neural networks: A comprehensive introduction*, Macmillan Publishing, Englewood Cliffs, NJ (1994).
- Henson, M.A., "Nonlinear model predictive control: current status and future directions," *Computers and Chemical Engineering* **23**, 187-202 (1998).
- Hernandez, E., and Y. Arkun, "Neural network modeling and an extended DMC algorithm to control nonlinear processes," *American Control Conference*, Vol. 2454-2459 (1990).
- Hocking, R.R., "Developments in linear regression methodology: 1959-1982," *Technometrics* **25**, 219-229 (1983).
- Hocking, R.R., and R.N. Leslie, "Selection of the best subset in regression analysis," *Technometrics* **9**, 531-540 (1967).
- Hornik, K., M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks* **2**, 359-366 (1989).
- Hunt, K.J., and D. Sbarbaro, *Studies in neural network based control*, London (1992).

Hush, D.R., and B.G. Horne, "Progress in supervised neural networks; What's new since Lippmann," (1993).

Hussain, M.A., "Review of the applications of neural networks in chemical process control - simulation and online implementation," *Artificial Intelligence in Engineering* **13**, 55-68 (1999).

Hwang, Y.L., "Nonlinear wave theory for dynamics of binary distillation column," *AIChE J.* **37**, 705-723 (1991).

Johnson, I.L., "Optimization of the solid-rocket assisted space shuttle ascent trajectory," *Journal of Spacecraft* **12**, 765-769 (1975).

Johnson, R.A., and D.W. Wichern, *Applied Multivariate Statistical Analysis*, Prentice Hall, NJ, (1988).

Jolliffe, I.T., *Principal Component Analysis*, Springer-Verlag, NY, (1986).

Kalman, R.E., "Contributions to the theory of optimal control," *Bull. Soc. Math. Mex.* **5**, 102-119 (1960a).

Kalman, R.E., "A new approach to linear filtering and prediction problems," *Trans. ASME, J. Basic Engineering* 35-45 (1960b).

Kanadibhotla, R.S., and J.B. Riggs, "Nonlinear model based control of a recycle reactor process," *Computers chem. Engng.* **19**, 933-948 (1995).

Karjala, T.W., and Himmelblau, "Dynamic data rectification by recurrent neural networks vs. traditional methods," *AIChE Journal* **40**, 1865-1875 (1994).

Kelly, S., M. Rogers, and D. Huffman, "Quadratic dynamic matrix control of hydrocarbon reactors," *American Control Conference*, Vol. 295-300 (1988).

Kinnaert, M., "Adaptive generalized predictive controller for MIMO systems," *International Journal of Control* **50**, 162-172 (1989).

Kleinman, D.L., "An easy way to stabilize a linear constant system," *IEEE Transactions of Automatic Control* **15**, 692 (1970).

Kramer, M.A., "Autoassociative neural networks," *Computers & Chemical Engineering* **16**, 313-328 (1992).

Kresta, J.V., T.E. Marlin, and J.F. MacGregor, "Development of inferential process models using PLS," *Computers & Chemical Engineering* **18**, 597-611 (1996).

- Krishnan, A., and K. Kosanovich, "Batch reactor control using a multiple model-based controller design," *The Canadian Journal of Chemical Engineering* **76**, 806-815 (1998).
- Kwon, W.H., and A.E. Pearson, "A modified quadratic cost problem and feedback stabilization of a linear system," *IEEE Transactions on Automatic Control* **AC-22**, 838-842 (1977).
- LaMotte, L.R., and R.R. Hocking, "Computational efficiency in the selection of regression variables," *Technometrics* **12**, 1, (1970).
- Lee, E.B., and L. Markus, *Foundations of optimal control theory*, Wiley, NY, (1967).
- Lee, J.H., and B. Cooley, "Robust Model Predictive Control of Multi-Variable Systems Using Input / Output Models with Stochastic Parameters," *American Control Conference*, Seattle, WA, Vol. pp 3694--3698 (1995).
- Lee, J.H., and B. Cooley, "Recent advances in model predictive control and other related areas," *Fifth international conference on chemical process control*, Tahoe City, California, Vol. 93, 201-216 (1996).
- Lee, J.H., M. Morari, and C.E. Garcia, "State space interpretation of model predictive control," *Automatica* **30**, 707-717 (1994).
- Lee, J.H., and N.L. Ricker, "Extended Kalman filter based nonlinear model predictive control," *Ind. Eng. Chem. Res.* **33**, 1530-1541 (1994).
- Lelic, M.A., and M.B. Zarrop, "Generalized pole-placement self-tuning control - Part I and II," *International Journal of Control* **46**, 548-568 (1987).
- Levine, J., and P. Rouchon, "Quality control of binary distillation columns via nonlinear aggregated models," *Automatica* **27**, 463-480 (1991).
- Ljung, L., *System identification - theory for the user*, Prentice-Hall, Englewood Cliff, NJ (1987).
- Lundstrom, P., J.H. Lee, M. Morari, and S. Skogestad, "Limitations of dynamic matrix control," *Computers chem. Engng.* **19**, 409-421 (1995).
- Luyben, W.L., "Simple regulatory control of the Eastman Process," *Ind. Eng. Chem. Res.* **35**, 3280-3289 (1996).
- Lyman, P.R., and C. Georgakis, "Plant-wide control of the Tennessee Eastman problem," *Computer chem. Engng.* **19**, 321-331 (1995).

- MacGregor, J.F., T.E. Marlin, and J.V. Kresta, "Some comments on neural networks and other empirical modelling methods," *AIChE Symp., CPC-IV*, South Padre Island, TX, Vol. 665-672 (1991).
- Maiti, S.N., and D.N. Saraf, "Adaptive dynamic matrix control of a distillation column with close-loop online identification," *J. Proc. Cont.* **5**, 315-327 (1995).
- Mallows, C.L., "Some comments on Cp," *Technometrics* **15**, 661-675 (1973).
- Maner, R.B., F.J. Doyle, B.A. Ogunnaike, and R.K. Pearson, "Nonlinear model predictive control of a simulated multivariable polymerization reactor using second-order Volterra models," *Automatica* **32**, 1285-1301 (1996).
- Marlin, T.E., *Process Control - Designing processes and control systems for dynamic performance*, McGraw Hill, NY, (1995).
- Mayne, D.Q., "Optimization in model predictive control," *NATO Advanced Study Institute*, Antalya, Turkey, (1995).
- Mayne, D.Q., "Nonlinear model predictive control: an assessment," *Fifth international conference on chemical process control*, Tahoe City, California, Vol. 93, 232-256 (1996).
- McAvoy, T.J., and N. Ye, "Base control for the Tennessee Eastman problem," *Computers chem. Engng.* **18**, 383-413 (1994).
- Meadows, E.S., and J.B. Rawlings, "Model predictive control," *Nonlinear Process Control*, (M.A. Henson and D.E. Seborg, Eds). Prentice Hall, NJ, (1997).
- Mehra, R.K., R. Rouhani, J. Eterno, J. Richalet, and A. Rault, "Model algorithmic control: review and recent development," *Engineering Foundation Conference on Chemical Process Control II*, Vol. 287-310 (1982).
- Mendel, J.M., *Lessons in estimation theory for signal, processing, communications, and control*, Prentice-Hall, Englewood Cliffs, NJ (1995).
- Meziou, A.M., P.B. Deshpande, C. Cozewith, N.I. Silverman, and W.G. Morrison, "Dynamic matrix control of an ethylene-propylene-diene polymerization reactor," *Ind. Eng. Chem. Res.* **35**, 164-168 (1996).
- Moody, J., and C.J. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Computation* **1**, 281-294 (1989).
- Muske, K.R., and J.B. Rawlings, "Model predictive control with linear models," *AIChE journal* **39**, 262-287 (1993).

- Narendra, K.S., and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on Neural Networks* **1**, 4-27 (1990).
- Nelson, W.L., *Petroleum Refinery Engineering*, McGraw Hill, NY, (1941).
- Ogunnaike, B.A., and W.H. Ray, *Process Dynamics, Modeling and Control*, Oxford University Press, NY, (1994).
- Ogunnaike, B.A., and R.A. Wright, "Industrial applications of nonlinear control," *Fifth international conference on chemical process control*, Tahoe City, California, Vol. 93, 46-59 (1997).
- Palavajhala, S., R. Motard, and B. Joseph, "Plantwide control of the Tennessee Eastman problem," *1993 Annual AIChE Meeting*, St. Louis, Vol. Paper 148g (1993).
- Pearson, R.K., and B.A. Ogunnaike, "Nonlinear process identification," *Nonlinear Process Control*, (M.A. Henson and D.E. Seborg, Eds). Prentice Hall, NJ, (1997).
- Piovosio, M., and A. Owens, "Sensor data analysis using artificial neural networks," *Chemical Process Control IV*, Vol. 101 (1991).
- Poggio, T., and F. Girosi, "Networks for approximation and learning," *Proceedings of the IEEE* **78**, 1481-1497 (1990).
- Pottman, M., and D.E. Seborg, "Identification of nonlinear processes using reciprocal muliquadric functions," *Journal of Process Control* **2**, 189-203 (1992).
- Pottmann, M., and D.E. Seborg, "A nonlinear predictive control strategy based on radial basis function models," *Computers chem. Engng.* **21**, 965-980 (1997).
- Prett, D.M., and R.D. Gillette, "Optimization and constrained multivariable control of a catalytic cracking unit," *AIChE National Meeting*, Houston, TX, Vol. (1979).
- Propoi, A.I., "Use of LP methods for synthesizing sampled-data automatic systems," *Autumn Remote Control* **24**, 837-844 (1963).
- Psichogios, D.M., and L.H. Ungar, "Direct and indirect model based control using artificial neural networks," *Industrial Engineering Chemistry Research* **30**, 2564-2573 (1991).
- Qin, J.S., and T.A. Badgwell, "An overview of industrial model predictive control technology," *Fifth international conference on chemical process control*, Tahoe City, California, Vol. 93, 232-256 (1997).
- Qin, S.J., H. Yue, and R. Dunia, "Self-validating inferential sensors with application to air emission monitoring," *Ind. Eng. Chem. Res.* **36**, 1675-1685 (1997).

Ramaker, B.L., H.K. Lau, and E. Hernandez, "Control technology challenges for the future," *Fifth international conference on chemical process control*, Tahoe City, California, Vol. 93, 1-7 (1997).

Rawlings, J.B., and K.R. Muske, "Stability of constrained receding horizon control," *IEEE Transactions on Automatic Control* **38**, 1512-1516 (1993).

Rhinehart, R.R., "A CUSUM type on-line filter," *Process control and quality* **2**, 169-176 (1992).

Rhinehart, R. R., "A hot/cold mixing process simulator," *Personal Communication* (1998).

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

Ricker, N.L., "Use of quadratic programming for constrained internal model control," *Ind. Eng. Chem. Process Des. Dev.* **24**, 925-936 (1985).

Ricker, N.L., "Model predictive control with state estimation," *Ind. Eng. Chem. Res.* **29**, 374-382 (1990).

Ricker, N.L., and J.H. Lee, "Nonlinear model predictive control of the Tennessee Eastman challenge process," *Computers chem. Engng* **19**, 961-981 (1995a).

Ricker, N.L., and J.H. Lee, "Nonlinear modeling and state estimation for the Tennessee Eastman challenge process," *Computers chem. Engng* **19**, 983-1005 (1995b).

Ricker, N.L., "Decentralized control of the Tennessee Eastman challenge process," *J. Proc. Cont.* **6**, 205-221 (1996).

Rumelhardt, D., G. Hinton, and R. Williams, *Parallel distributed processing: explorations in the microstructures of cognition*, MIT Press, (1986).

Samdani, G., "Neural nets. They learn from examples," (1990).

San, Y.P., K.C. Landells, and D.C. Mackay, "Inferential control - Conclusion; Crude unit controls reduce quality giveaway, increase profits," (1994a).

San, Y.P., K.C. Landells, and D.C. Mackay, "Inferential control - Part 1; Crude unit advanced controls pass accuracy and repeatability tests," (1994b).

Santos, L., N. deOliveria, and L.T. Biegler, "Reliable and efficient optimization strategies for nonlinear model predictive control," *Fourth IFAC Symposium on*

Dynamics and Control of Chemical Reactor, Distillation Columns and Batch Processes (DYCORD '95), Helsingor, Denmark, Vol. 33-38 (1995).

Schnelle, D., and J. Fletcher, "Using neural based process modeling in process control," *ISA-90 International Conference*, (1990).

Schwarz, G., "Estimating the dimension of a model," *Annals of Statistics* **6**, 461-464 (1978).

Scokaert, P.O.M., and J.B. Rawlings, "Constrained linear quadratic regulation," *IEEE Transactions on Automatic Control* **43**, 1163-1169 (1998).

Soderstrom, T., and P. Stoica, *System Identification*, Prentice Hall, NY (1989).

Srinivas, G.R., and Y. Arkun, "Optimization and convergence issues for MPC algorithms using polynomial ARX models," *AIChE Meeting*, Miami, Florida, Vol. Paper 185c (1995).

Srinivas, G.R., and Y. Arkun, "Control of the Tennessee Eastman process using input-output models," *J. Proc. Cont.* **7**, 387-400 (1997).

Staus, G.H., L.T. Biegler, and B.E. Ydstie, "Adaptive control via non-convex optimization," *Nonconvex optimization and its applications*, (C. Floudas and P. Pardolas, Eds). Kluwer, (1996).

Steel, R. G. D., J.H. Torrie, *Principles and Procedures of Statistics: A Biometric Approach*, McGraw Hill, NY, (1980).

Su, T.H., and T.J. McAvoy, "Artificial neural networks for nonlinear process identification and control," *Nonlinear process control*, (M. A. Henson and D. E. Seborg, Eds). Prentice Hall, NJ, (1997).

Sznaier, M., and M.J. Damborg, "Heuristically enhanced feedback control of constrained discrete-time linear systems," *Automatica* **26**, 521-532 (1990).

Thomas, Y.A., "Linear quadratic optimal estimation and control with receding horizon," *Electronic Letter* **11**, 19-21 (1975).

Turner, P., G.A. Montague, and A.J. Morris, "Neural networks in dynamic process state estimation and nonlinear predictive control," *International Conference on Neural Networks*, Cambridge, Vol. 284-289 (1995).

Tyreus, B.D., "Dominant variables for partial control. 2. Application to the Tennessee Eastman Challenge Process," *Ind. Eng. Chem. Res.* **38**, 1444-1455 (1999).

Uppal, A., W.H. Ray, and A.B. Poore, "On the dynamic behavior of continuous stirred tank reactor," *Chem. Eng. Sci.* **29**, 967-985 (1974).

Van Hoof, A., C. Cutler, and S. Finlayson, "Application of a constrained multi-variable controller to a hydrogen plant," *American Control Conference*, Vol. (1989).

Venkatasubramanian, V., R. Vaidyanathan, and Y. Yamamoto, "Process fault detection and diagnosis using neural networks - I. Steady state processes," *Comput. Chem. Eng.* **14**, 583 (1990).

Vuthandam, P., H. Genceli, and M. Nikolaou, "Performance bounds for robust quadratic dynamic matrix control with end condition," *AIChE Journal* **41**, 2083-2097 (1995).

Walls, R.C., and D.L. Weeks, "A note on the variance of a predicted response in regression," *American Statistician* **23**, 24-26 (1969).

Watkins, R.N., *Petroleum Refinery Distillation*, Gulf Publishing Co., Houston, Texas, (1979).

Willis, M. J., G.A. Montague, and A.J. Morris, "Artificial neural networks in process engineering," *IEEE Proceedings. Part D, Control Theory and Applications*, Vol. 138, 256-266 (1991).

Wood, R.K., and M.W. Berry, "Terminal composition control of a binary distillation column," *Chem. Eng. Sci.* **29**, 1808 (1973).

Yang, X., J. Zhang, and A.J. Morris, "An artificial neural network approach for inferential measurement," *International Conference - Neural Network Processing (ICONIP)*, Beijing, Vol. 1, 485-488 (1995).

Ydstie, B.E., "Certainty equivalence adaptive control: what's new in the gap," *Fifth international conference on chemical process control*, Tahoe City, California, Vol. 93, 9-23 (1996).

Zafiriou, E., "Robust model predictive control of processes with hard constraints," *Computers chem. Engng.* **14**, 359-371 (1990).

Zafiriou, E., and A.L. Marchal, "Stability of SISO quadratic dynamic matrix control with hard output constraints," *AIChE Journal* **37**, 1550-1560 (1991).

APPENDIX A -- QUADRATIC DYNAMIC MATRIX CONTROL

A-1 Introduction

The Quadratic Dynamic Matrix Control algorithm makes use of the a step response model in which the output prediction is a function of the input changes,

$$\Delta u_k = u_k - u_{k-1} \quad (\text{A-1})$$

where k is the sample instant and u_k is the input to the process at k . At sample instant 0, a unit step input is applied to the system. A typical discrete-response of a stable system is shown in Figure A-1.

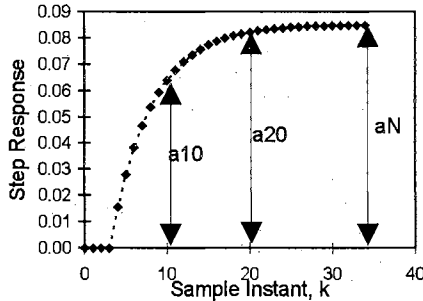


Figure A-1: Step response model

The step response gives no information at times between the sampled points. The values of the response at the sample instants are often referred to as the step coefficients. Assuming that the plant is linear, the overall effect of all step inputs is evaluated as the sum of each individual effect. The step coefficients can thus be used to predict the output from the input values at each sample instant as follows:

$$\begin{aligned}
y_1 &= a_1 \Delta u_0 \\
y_2 &= a_2 \Delta u_0 + a_1 \Delta u_1 \\
&\dots \\
y_k &= \sum_{i=1}^{N-1} a_i \Delta u_{k-i} + a_n u_{k-N}
\end{aligned}
\tag{A-2}$$

where N is the number of sample periods required for the step response to reach steady-state conditions.

QDMC belongs to the general class of Model Predictive Control (MPC) algorithms. Often, due to slow process dynamics, it takes a substantial amount of time for the effect of each control action taken to be fully reflected in the process measurements. Thus, it is not possible to interpret the results of the current output using only the input applied at the previous instant. Like other model predictive controllers, QDMC determines the future behavior of the plant if no further control action is taken. The future period, p (sample period), called the prediction horizon, must be long enough to ensure that the past inputs have completely manifested in the process outputs. The task of the controller is to determine a set of plant inputs over an input horizon, c (sample period), which will minimize the future errors in presence of process constraints.

A-2 QDMC Algorithm

The discrete-time step-response model is used to predict the future output values. The following is an algorithmic description of the QDMC controller. Let the current sample instant be k .

Step 1: Input History Shift

The value of the past inputs is shifted backwards in time, thus retaining only the past N values of the inputs.

$$\Delta u_{k-N+i-1} = \Delta u_{k-N+i} \quad \forall i = 1, N \quad (\text{A-3})$$

where N is the number of sample instants for the step response of the process to reach steady-state.

Step 2: Output Feedback

To relate the prediction of the model with the current measurement, \hat{y}_k , a model bias term b_k is computed. The use of the bias is to set the model prediction at the current time to the current output. It is computed as follows:

$$b_k = y_k - \left(\sum_{j=1}^{N-1} a_j \Delta u_{k-j} + a_N u_{k-N} \right) \quad (\text{A-4})$$

where a_i , $i = 1, N$ are the step response coefficients as discussed previously.

This simple form of feedback is often regarded as estimation of an output disturbance that is assumed to be constant for all future time.

Step 3: Reference Trajectory

At every instant k a smooth path from the current measured output, \hat{y}_k , to the setpoint y_{sp} , called the reference trajectory, is computed. The path can be parameterized in terms of a desired closed loop time constant τ_r :

$$r_{k+i} = \begin{cases} y_{k+i}, & i = 1, \alpha_r \\ y_{k+\alpha_r} + (y_{sp} - y_{k+\alpha_r})(1 - e^{-(i-\alpha_r)\Delta t / \tau_r}), & i = \alpha_r + 1, p \end{cases} \quad (\text{A-5})$$

where α_r is the time delay of the process.

Step 4: Move Calculation

To compute the future moves, the model prediction at the $(k+i)^{\text{th}}$ instant is separated into a past input contribution, y^p_{k+i} and a future input contribution, y^f_{k+i} .

$$y_{k+i} = y^f_{k+i} + y^p_{k+i} + b_k \quad (\text{A-6})$$

where the future input contribution is evaluated using the step-response model as:

$$y^f_{k+i} = \sum_{j=1}^i a_j \Delta u_{k+i-j} \quad \forall i = 1, p \quad (\text{A-7})$$

and the past input contribution is evaluated as

$$y^p_{k+i} = \sum_{j=i+1}^{N-1} a_j \Delta u_{k+i-j} + a_N u_{k+i-N} \quad \forall i = 1, N-2 \quad (\text{A-8})$$

$$y^p_{k+i} = a_N u_{k-1} \quad \forall i = N-1, p \quad (\text{A-9})$$

The future errors are the differences between the setpoints and the predicted output at each future instant $k+i$, $\forall i = 1, p$:

$$e_{k+i} = r_{k+i} - y_{k+i} \quad (\text{A-10})$$

Using equation (A-6), the above equation may be rewritten as

$$e_{k+i} = (r_{k+i} - y^p_{k+i} - b_k) - y^f_{k+i} \quad (\text{A-11})$$

Let,

$$\hat{e}_{k+i} = r_{k+i} - y^p_{k+i} - b_k \quad (\text{A-12})$$

The term \hat{e}_{k+i} represents the future deviations of the output from the reference trajectory that would occur if no future control adjustments were made. Thus,

$$e_{k+i} = \hat{e}_{k+i} - y^f_{k+i} \quad (\text{A-13})$$

The future input contribution to the projected outputs may be written in a matrix form as:

$$\begin{bmatrix} y^f_{k+1} \\ y^f_{k+2} \\ y^f_{k+3} \\ \vdots \\ y^f_{k+c} \\ \vdots \\ y^f_{k+N} \\ \vdots \\ y^f_{k+p} \end{bmatrix} = \begin{bmatrix} a_1 & 0 & 0 & \cdots & 0 \\ a_2 & a_1 & 0 & \cdots & 0 \\ a_3 & a_2 & a_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ a_c & a_{c-1} & a_{c-2} & \cdots & a_1 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ a_N & a_{N-1} & a_{N-2} & \cdots & a_{N-c+1} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ a_N & a_N & a_N & \cdots & a_{p-c+1} \end{bmatrix} \begin{bmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \Delta u_{k+2} \\ \vdots \\ \Delta u_{k+c-1} \end{bmatrix} \quad (\text{A-14})$$

or equivalently

$$\mathbf{y}^f = \mathbf{A}\Delta\mathbf{u} \quad (\text{A-15})$$

Here, the matrix \mathbf{A} is called the dynamic matrix. Combining this equation with (A-11) we obtain:

$$\mathbf{e} = \hat{\mathbf{e}} - \mathbf{A}\Delta\mathbf{u} \quad (\text{A-16})$$

Finally, the control input to the process is calculated as the solution to the constrained optimization problem:

$$\min_{\Delta\mathbf{u}} \phi = (\hat{\mathbf{e}} - \mathbf{A}\Delta\mathbf{u})^T \Gamma^T \Gamma (\hat{\mathbf{e}} - \mathbf{A}\Delta\mathbf{u}) + \Delta\mathbf{u}^T \Lambda^T \Lambda \Delta\mathbf{u} \quad (\text{A-17})$$

subject to

$$\begin{aligned} \Delta\mathbf{u}_{\min} &\leq \Delta\mathbf{u} \leq \Delta\mathbf{u}_{\max} \\ \mathbf{u}_{\min} &\leq \mathbf{u} \leq \mathbf{u}_{\max} \\ \mathbf{y}_{\min} &\leq \mathbf{y} \leq \mathbf{y}_{\max} \end{aligned} \quad (\text{A-18})$$

A-3 Formulation of the Quadratic Program(QP)

The objective function can be recast as:

$$\phi = \Delta \mathbf{u}^T \mathbf{H} \Delta \mathbf{u} - 2 \mathbf{g}^T \Delta \mathbf{u} + \hat{\mathbf{e}}^T \Gamma^T \Gamma \hat{\mathbf{e}} \quad (\text{A-19})$$

where the Hessian matrix, \mathbf{H} , and the gradient vector, \mathbf{g} , are given by

$$\begin{aligned} \mathbf{H} &= \mathbf{A}^T \Gamma^T \Gamma \mathbf{A} + \mathbf{A}^T \mathbf{A} \\ \mathbf{g} &= \mathbf{A}^T \Gamma^T \Gamma \hat{\mathbf{e}} \end{aligned} \quad (\text{A-20})$$

The constraints are converted in terms of the design variable, $\Delta \mathbf{u}$ as follows:

$$\begin{bmatrix} u_k \\ u_{k+1} \\ \vdots \\ u_{k+c-1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \vdots \\ \Delta u_{k+c-1} \end{bmatrix} \quad (\text{A-21})$$

Or in vector notation,

$$\mathbf{u} = \mathbf{L} \Delta \mathbf{u} + \mathbf{u}_0 \quad (\text{A-22})$$

Thus the inequality constraints (A-18) are reformulated as

$$(\mathbf{u}_{\min} - \mathbf{u}_0) \leq \mathbf{L} \Delta \mathbf{u} \leq (\mathbf{u}_{\max} - \mathbf{u}_0) \quad (\text{A-23})$$

Similarly, the constraints on the process output can be written in term of the design variable, $\Delta \mathbf{u}$ as

$$(\mathbf{y}_{\min} - \mathbf{y}^p - \mathbf{b}) \leq \mathbf{A} \Delta \mathbf{u} \leq (\mathbf{y}_{\max} - \mathbf{y}^p - \mathbf{b}) \quad (\text{A-24})$$

Thus, the optimization problem represented by equation (A-17) and equation (A-18) in the QDMC algorithm can be stated in the form of a standard QP:

$$\min_{\Delta \mathbf{u}} \phi = \Delta \mathbf{u}^T \mathbf{H} \Delta \mathbf{u} - 2 \mathbf{g}^T \Delta \mathbf{u} \quad (\text{A-25})$$

subject to:

$$\begin{bmatrix} \Delta \mathbf{u}_{\min} \\ \mathbf{u}_{\min} - \mathbf{u}_0 \\ y_{\min} - y^p - b \end{bmatrix} \leq \begin{bmatrix} \mathbf{I} \\ \mathbf{L} \\ \mathbf{A} \end{bmatrix} \Delta \mathbf{u} \leq \begin{bmatrix} \Delta \mathbf{u}_{\max} \\ \mathbf{u}_{\max} - \mathbf{u}_0 \\ y_{\max} - y^p - b \end{bmatrix} \quad (\text{A-26})$$

The constrained problem is solved at each control interval using a standard QP code. For reasonable tuning parameters, the Hessian matrix will be positive definite, which makes the optimization problem relatively simple.

Performance of the control system improves as the control horizon c increases, since the optimization problem will then have additional degrees of freedom with which to minimize future prediction errors. The prediction horizon p should be long enough to capture the steady-state effects of moves. Garcia and Morshedi (1986) recommend the setting:

$$p = N + c \quad (\text{A-27})$$

Figure A-2 illustrates the QDMC algorithm

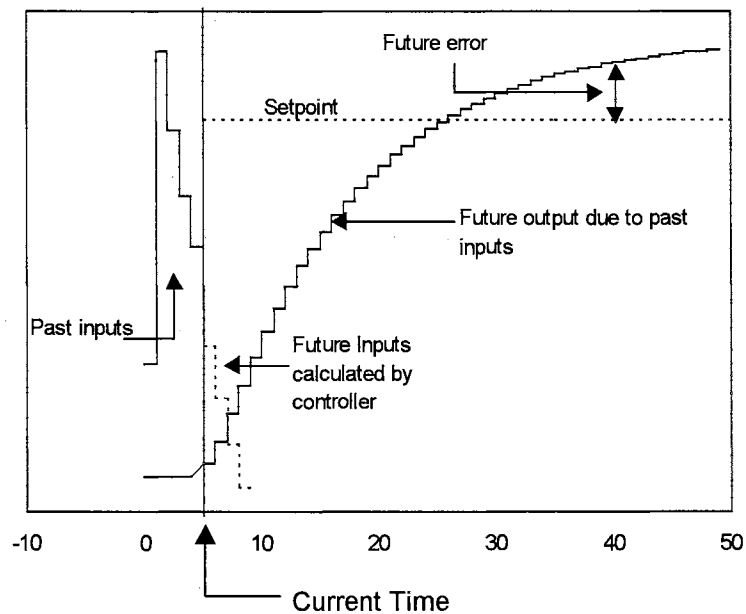


Figure A-2: Illustration of QDMC control

APPENDIX B -- RECURSIVE LEAST SQUARES ALGORITHM

B-1 Introduction

The least squares method is a basic technique in parameter estimation and is particularly simple if the model is linear in the parameters. Let $y(i)$, the observation at the i^{th} instant, be related to known variables, $\varphi_1(i), \varphi_2(i), \dots, \varphi_n(i)$ as follows:

$$y(i) = \varphi_1(i)\theta_1 + \varphi_2(i)\theta_2 + \dots + \varphi_n(i)\theta_n \quad (\text{B-1})$$

or in vector notation

$$y(i) = \boldsymbol{\phi}^T \boldsymbol{\theta} \quad (\text{B-2})$$

where the regressor vector $\boldsymbol{\phi}(i)$ is given by

$$\boldsymbol{\phi}^T(i) = [\varphi_1(i) \quad \varphi_2(i) \quad \dots \quad \varphi_n(i)] \quad (\text{B-3})$$

and the parameter vector, $\boldsymbol{\theta}(i)$ is:

$$\boldsymbol{\theta} = [\theta_1 \quad \theta_2 \quad \dots \quad \theta_n]^T \quad (\text{B-4})$$

The model index i often denotes time while n is the number of parameters. Pairs of observation and the regressors $\{(y(i), \boldsymbol{\phi}(i)), i=1,2,\dots,t\}$ are obtained from an experiment. Then, the least squares problem is to determine the model parameters in such a way that the model outputs agree as closely as possible with the observations in the least square sense. The parameter vector $\boldsymbol{\theta}$ should be chosen to minimize the least squares objective function:

$$V(\boldsymbol{\theta}, t) = \frac{1}{2} \sum_{i=1}^t (y(i) - \boldsymbol{\phi}^T(i)\boldsymbol{\theta})^2 \quad (\text{B-5})$$

where t is the total number of measurements available. Let

$$\mathbf{\Phi}(t) = \begin{bmatrix} \varphi^T(1) \\ \varphi^T(2) \\ \vdots \\ \varphi^T(3) \end{bmatrix} \quad (\text{B-6})$$

and

$$\mathbf{P}(t) = \left(\mathbf{\Phi}^T(t) \mathbf{\Phi}(t) \right)^{-1} \quad (\text{B-7})$$

Using equation (B-6) the above equation can be rewritten as

$$\mathbf{P}(t) = \left(\sum_{i=1}^t \varphi(i) \varphi^T(i) \right)^{-1} \quad (\text{B-8})$$

The least -square computations are arranged in such a way that the estimates obtained at time $t-1$ are used to evaluate the estimates at time t . An important assumption made is that the matrix $\mathbf{\Phi}^T \mathbf{\Phi}$ is non-singular for all t . Then given initial conditions $\theta(t_0)$ and $\mathbf{P}(t_0)$, the least-square estimate satisfies the recursive equations:

$$\begin{aligned} \theta(k) &= \theta(k-1) + \mathbf{K}(k) \left(y(k) - \varphi^T(k) \theta(k-1) \right) \\ \mathbf{K}(k) &= \mathbf{P}(k-1) \varphi(k) \left(\mathbf{I} + \varphi^T(k) \mathbf{P}(k-1) \varphi(k) \right)^{-1} \\ \mathbf{P}(k) &= \left(\mathbf{I} - \mathbf{K}(k) \varphi^T(k) \right) \mathbf{P}(k-1) \end{aligned} \quad (\text{B-9})$$

Thus, the estimate at time t is obtained by adding a correction to the previous estimate. The correction is proportional to the error in prediction and is also called the innovation.

with the assumption that the autocorrelation matrix $\sum_{i=1}^t \varphi(i) \varphi^T(i)$ is nonsingular for all available t measurements. This error reflects a part of the new measurement at time k which could not be predicted by the previous model and is therefore called the innovation process. Matrix \mathbf{P} provides an estimate of the measure of parameter vector covariance (Mendel, 1995). Matrix $\mathbf{K}(k)$ is often referred to as the adaptation gain. A detailed

treatment of least square estimators can be found in standard texts on estimation theory (Ljung, 1987; Mendel, 1995).

The matrix $\mathbf{P}(t)$ is defined only when $\Phi^T \Phi$ is non-singular. It follows from equation (B-8) that $\Phi^T \Phi$ is always singular if $t < n$. However, it is convenient to use the recursive equations in all steps. If the recursive equations use the initial condition

$$\mathbf{P}(0) = \mathbf{P}_0 \quad (\text{B-10})$$

where \mathbf{P}_0 is positive definite, then

$$\mathbf{P}(t) = \left(\mathbf{P}_0^{-1} + \Phi^T(t) \Phi(t) \right)^{-1} \quad (\text{B-11})$$

Thus, $\mathbf{P}(t)$ can be made arbitrarily close to $(\Phi^T \Phi)^{-1}$ by choosing a sufficiently large \mathbf{P}_0 .

B-2 Properties of Least Squares

The recursive least squares algorithm is formulated from the well known batch solution. It is, therefore, of interest to evaluate the relationship between the two algorithms.

Batch v/s Recursive: Let the parameter vector consist of d_1 elements. Then assuming availability of N observations, the well known batch least squares solution takes the form,

$$\hat{\boldsymbol{\theta}}_{\text{Batch}}(N) = \left(\sum_{k=1}^N \phi(k) \phi^T(k) \right)^{-1} \sum_{k=1}^N \phi(k) \mathbf{y}(k) \quad (\text{B-12})$$

If the estimate is instead calculated by recursive least squares, the following estimate is obtained,

$$\hat{\boldsymbol{\theta}}_{RLS}(N) = \left(\mathbf{P}^{-1}(0) + \sum_{k=1}^N \boldsymbol{\phi}(k) \boldsymbol{\phi}^T(k) \right)^{-1} \left(\sum_{k=1}^N \boldsymbol{\phi}(k) y(k) + \mathbf{P}^{-1}(0) \hat{\boldsymbol{\theta}}(0) \right) \quad (\text{B-13})$$

where $\boldsymbol{\theta}(0)$ is the initial estimate and $\mathbf{P}(0)$ is the initial covariance estimator. Thus, by making $\mathbf{P}(0)$ positive definite but arbitrarily large, the recursive estimation can be made arbitrarily close to the batch solution. Choosing a large positive definite value of $\mathbf{P}(0)$ also reduces the influence of the initial estimate $\boldsymbol{\theta}(0)$ on the subsequent iterates. Thus, the RLS estimator enjoys the same properties as the batch estimate, provided appropriate initial conditions for \mathbf{P} are chosen.

Conditions for Unbiasedness of Least Square Estimates: Stochastic estimators assume that the model parameters are random variables. Thus, a primary concern is to evaluate the conditions under which the estimates will be unbiased. A discussion of these conditions for the least square estimate is provided below.

Assume that the data are generated by the equation,

$$y(i) = \boldsymbol{\phi}(i)^T \boldsymbol{\theta}^0 + e(i) \quad (\text{B-14})$$

where $e(i)$ is white noise of zero mean and known variance. Then, the parameter estimate $\hat{\boldsymbol{\theta}}$ is an unbiased estimate of $\boldsymbol{\theta}^0$, provided the regressor vector $\boldsymbol{\phi}$ is deterministic (Soderstrom and Stoica, 1989). However, this is a serious limitation since the disturbance $e(i)$ invariably imparts a stochastic character to the process output $y(i)$, thereby, making $\boldsymbol{\phi}$ non-deterministic. Further, since the plant input is generated as a feedback reaction, plant inputs, $u(i)$, is also generally non-deterministic.

A less stringent condition allows ϕ to be a stochastic variable, but deems that it be independent of the disturbance, $e(i)$. However, this condition too is violated in practice due to the interaction between the plant outputs and inputs with the disturbance. Results for unbiasedness of least square estimates have been proved for large samples under relatively weaker conditions. One limitation in the large sample estimate is that the input signal must be persistently exciting of sufficient degree (Soderstrom and Stoica, 1989). Further, the nature of feedback must be complex enough to avoid development of linear dependencies of regressor vectors. In section 3.4, it is shown that a SISO DMC controller satisfies the later condition.

Despite of the restrictive conditions for unbiasedness, least square estimates have the attractive property of computational simplicity. The recursive algorithm is computationally efficient and does add any significant burden to the online control calculations of MPC. Further, the presence of small bias may be tolerable since MPC methods incorporate feedback to overcome model uncertainty. In contrast, modifications of least squares such as instrument variable methods provide unbiased and consistent estimates under less restrictive condition. However, the implementation is more involved.

The properties of least square estimate, indeed, any estimator are crucially linked to model selection. For instance, if the process data is governed by an autoregressive, moving average model with external inputs (ARMAX) and the identification model is of the ARX type, the parameter estimates are likely to be biased.

APPENDIX C -- QUADRATIC PROGRAMMING USING ACTIVE SET METHOD

C-1 Introduction

Many questions dealing with "what is the 'best' approach" employ optimization techniques. Such applications arise in various fields of science and engineering, which are often models of reality. The index of 'goodness' is measured by an objective function, $f(\mathbf{x})$, where the elements of \mathbf{x} represent the independent or decision variables. The optimal solution refers to the variable \mathbf{x}^* at which the objective function has a minimum or a maximum value.

For example, consider the optimal operation of a distillation column. The column is used to separate mixtures of components of differing volatility. We may formulate an objective function that measures the revenues from this column. Thus, based on market conditions (that is, demand, cost, etc.), it might be desirable to maximize yield of some product. The decision variables could be feed rate, separated component purity among others. Furthermore, constraints may be associated with the objective function. For instance, the flow rates cannot be negative, the component purity must lie between 0 and 100% and other system constraints.

C-1.1 Problem Statement

A problem, such as above may be mathematically formulated as,

$$\begin{aligned}
& \underset{\mathbf{x}}{\text{minimize}} \quad f(\mathbf{x}), \quad \mathbf{x} \in \mathfrak{R}^n \\
& \text{subject to} \quad c_i(\mathbf{x}) = 0 \quad i \in E \\
& \quad \quad \quad c_i(\mathbf{x}) \geq 0 \quad i \in I
\end{aligned} \tag{C-1}$$

where E and I refer to the set of indices of equality and inequality constraints, $c_i(\mathbf{x})$, respectively. The solution search space is n -dimensional. Note that a maximization problem could be cast as a minimization one by choosing the objective function to be $-f(\mathbf{x})$. Equation (C-1) defines the general constrained optimization problem

In many applications, the objective function is quadratic (such as minimize sum of squared errors of a linear system or a quadratic approximation of a nonlinear function) and the constraint functions linear (or affine). Such a formulation defines the quadratic programming problem. It is expressed as,

$$\begin{aligned}
& \underset{\mathbf{x}}{\text{minimize}} \quad f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{G} \mathbf{x} + \mathbf{g}^T \mathbf{x} \\
& \text{subject to:} \quad \mathbf{a}_i^T \mathbf{x} = \mathbf{b}_i \quad i \in E \\
& \quad \quad \quad \mathbf{a}_i^T \mathbf{x} \geq \mathbf{b}_i \quad i \in I
\end{aligned} \tag{C-2}$$

Here, \mathbf{G} represents the symmetric Hessian matrix or the curvature of the objective surface. Vectors \mathbf{a}_i denote the gradients of constraints, $c_i(\mathbf{x})$, with respect to \mathbf{x} .

Thus, the goal of an optimization algorithm is solution of the quadratic program presented in equation (C-2), that is, search for a point, \mathbf{x}^* , in \mathfrak{R}^n such that the objective function $f(\mathbf{x})$ is minimized in addition to satisfying the set of constraints.

C-1.2 Terminology

Before proceeding further, a few terms used in solving the quadratic program in equation (C-2) are reviewed.

Feasibility:

Any point \mathbf{x}' in \mathfrak{R}^n that satisfies all constraints in equation (C-2) is said to be a feasible point. The set of feasible points is referred to as the feasible region. If the constraints are inconsistent, then the problem will be infeasible (ex: minimize $f(\mathbf{x})$ subject to $x_1 > 2$ and $x_1 < 1$). During a search, an incremental step δ , such that the resulting point \mathbf{x} exists in the feasible region, is called a feasible step.

Active Constraint:

Constraints, $c_i(\mathbf{x})$, $i \in A$, are said to be active at \mathbf{x}' , if \mathbf{x}' lies on the boundary of the feasible region and this boundary is formed by the constraints whose indices are members of set A . Set A is referred to as the active set. Note that all equality constraints are necessarily active, i.e. $E \subseteq A$. During the search process, some inequality constraints may become active and their indices will also be included in A .

C-1.3 An Example

To illustrate these concepts further, the following simple example is presented.

$$\begin{aligned}
& \underset{\mathbf{x}}{\text{minimize}} \quad f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \mathbf{x} + \begin{bmatrix} -2 \\ -2 \end{bmatrix}^T \mathbf{x} + 2, \quad \mathbf{x} \in \mathfrak{R}^2 \\
& \text{subject to:} \quad c_1(\mathbf{x}): \begin{bmatrix} 1 \\ 1 \end{bmatrix}^T \mathbf{x} \geq 2 + \sqrt{2} \quad 1 \in I \\
& \quad \quad \quad c_2(\mathbf{x}): \begin{bmatrix} -1 \\ 1 \end{bmatrix}^T \mathbf{x} \geq -2 \quad 2 \in I
\end{aligned} \tag{C-3}$$

Only, two decision variables are considered to allow graphical display of various properties. The objective function represents as family of circles centered at $[1 \ 1]^T$. The linear constraints represent straight lines. These are shown in Figure C- 1. The dashed lines represent inequality constraints $c_1(\mathbf{x})$ and $c_2(\mathbf{x})$. The solid curves refer to the contours of the objective function. From the figure, it is clear that the minimum is achieved at $\left[1 + \frac{1}{\sqrt{2}} \quad 1 + \frac{1}{\sqrt{2}}\right]^T$, where the active constraint $c_1(\mathbf{x})$ is tangent to the circle of unit radius. In the problem statement (equation (C-3)), the set of equality constraint indices, E , is the null set, while the set of inequality constraint indices, I , is the set $\{1,2\}$. At the minimum point, \mathbf{x}^* , only the first constraint is active. Thus, $A=\{1\}$. Ignoring the second constraint does not affect the solution in any way.

The remainder of this appendix is arranged as follows. Section C-2 describes method of Lagrange multipliers to solve constrained optimization problems. Active set methods are used to handle inequality constraints and will be the topic of discussion in section C-3. The example problem stated in equation (C-3) will be used to demonstrate various features of constrained optimization. Finally, a brief description of additional considerations will be provided in section C-4.

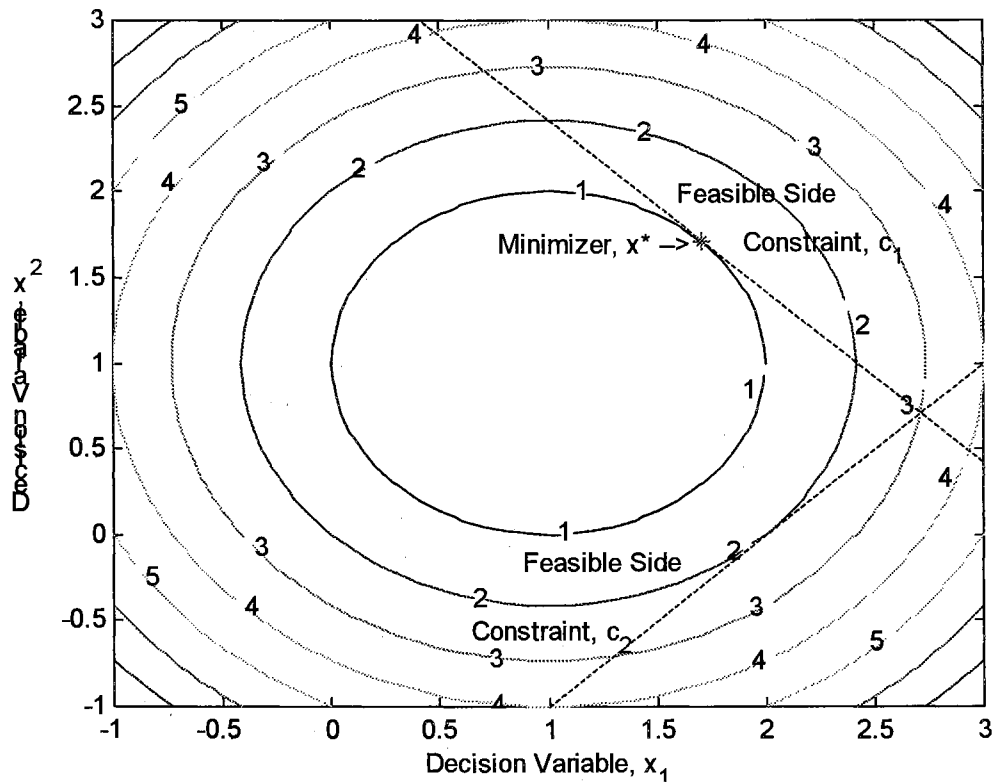


Figure C-1: Illustration of some concepts using quadratic program in equation (C-3) as an example. See section 1.2 for details.

C-2 Method of Lagrange Multipliers

In unconstrained optimization, necessary and sufficient conditions for a minimum \mathbf{x}^* are based on first and second order conditions.

$$\begin{aligned} \nabla_{\mathbf{x}} f(\mathbf{x})|_{\mathbf{x}^*} &= \mathbf{0} \\ \nabla^2_{\mathbf{x}} f(\mathbf{x})|_{\mathbf{x}^*} &\geq \mathbf{0} \end{aligned} \tag{C-4}$$

To generalize this concept to constrained optimization, the notion of Lagrange multipliers is introduced. Here, an additional complication of feasible region is introduced. For \mathbf{x}^* to be a minimum, no feasible descent direction must exist.

Let us momentarily assume that we have a set of equality constraints alone. The feasible minimum point, \mathbf{x}^* , must necessarily lie on the intersection of these constraints. Let δ represent an incremental feasible step from the minimum point. Then the new position, $\mathbf{x}^* + \delta$, must also lie on the linear equality constraints. Thus,

$$\mathbf{a}_i^T (\mathbf{x}^* + \delta) - \mathbf{b}_i = 0, \quad \forall i \in E \quad (\text{C-5})$$

However, since \mathbf{x}^* is a feasible point, it satisfies the equality constraint in equation (C-2). It, therefore follows that,

$$\mathbf{a}_i^T \delta = 0 \quad (\text{C-6})$$

Equation (C-6) provides a means of identifying feasible directions. If in addition, $f(\mathbf{x})|_{\mathbf{x}^*}$ has a negative slope along δ , that is

$$\delta^T \mathbf{g}^* < 0 \quad (\text{C-7})$$

then the feasible directions along δ will reduce $f(\mathbf{x})$. However, since \mathbf{x}^* is a local minimum this cannot occur, that is, no further feasible descent directions are possible at the minimum point. Thus, equations (6) and (7) cannot be satisfied simultaneously at the minimum point \mathbf{x}^* . The preceding statement will not be violated if \mathbf{g}^* is a linear combination of the vectors \mathbf{a}_i , $i \in A$, that is,

$$\mathbf{g}^* = \sum_{i \in E} \mathbf{a}_i^* \lambda_i^* = \mathbf{A}^* \boldsymbol{\lambda}^* \quad (\text{C-8})$$

where, \mathbf{A}^* denotes the matrix with \mathbf{a}_i^* arranged in columns. Equation (C-8) forms the necessary condition for a local minimizer. The coefficients λ_i^* are referred to as the Lagrange multipliers. The superscript $*$ indicates that the multipliers are associated with the minimum solution \mathbf{x}^* .

These ideas are illustrated in Figure C- 2. To be consistent with the discussion above, we temporarily assume that constraint c_1 in equation (C-3) is an equality constraint and let us ignore constraint c_2 completely. Thus, $E=\{1\}$ and $I=\{ \}$. Consider the point \mathbf{x}'' whose coordinates are $(1,1+\sqrt{2})$, in Figure C- 2, which lies on the c_1 constraint. At point \mathbf{x}'' , which is not a local minimum point, $\mathbf{g}'' \neq \mathbf{a}_1'' \lambda_1$, since \mathbf{g}'' and \mathbf{a}_1'' are non-collinear. Thus, there exists an incremental feasible step, δ , as shown that will satisfy both equations (6) and (7). Thus, δ represents a feasible step in the descent direction. Taking this step will reduce the value of the objective function as is evident from the figure. On the other hand, at the minimum point, \mathbf{x}^* , $(1+1/\sqrt{2},1+1/\sqrt{2})$, the gradient, $\mathbf{g}^* = \mathbf{a}_1^* \lambda_1$ satisfies equation (C-8) and hence no feasible descent direction exists.

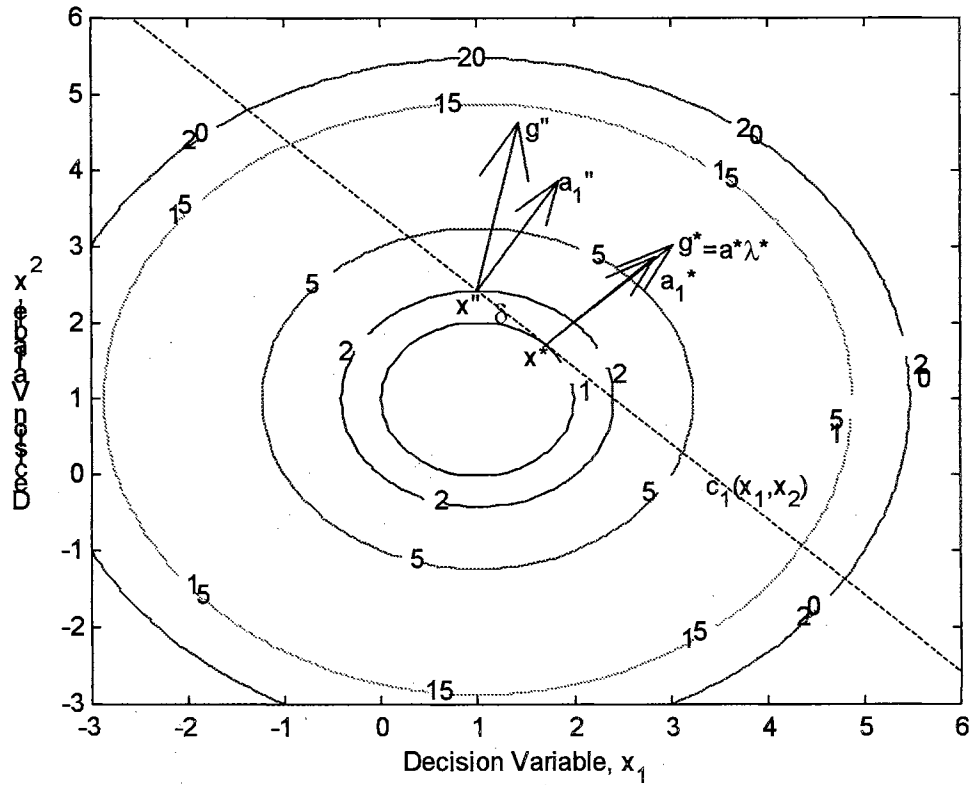


Figure C-2: The gradient of the objective function and constraints are collinear at the local minimum \mathbf{x}^* . At any other non-stationary point, equation (C-8) is not satisfied.

Equation (C-8) can be written more conveniently by introducing the Lagrange function,

$$L(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) - \sum_{i \in E} \lambda_i c_i(\mathbf{x}) \quad (\text{C-9})$$

Then, the conditions for a local minimizer translate to

$$\nabla L(\mathbf{x}^*, \boldsymbol{\lambda}^*) = 0, \quad \text{where } \nabla = \begin{bmatrix} \nabla_{\mathbf{x}} \\ \nabla_{\boldsymbol{\lambda}} \end{bmatrix} \quad (\text{C-10})$$

For the quadratic objective function and linear constraints in equation (C-2), the condition represented by equation (C-10) becomes,

$$\begin{bmatrix} \mathbf{G} & -\mathbf{A} \\ -\mathbf{A}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \lambda \end{bmatrix} = \begin{bmatrix} -\mathbf{g} \\ -\mathbf{b} \end{bmatrix} \quad (\text{C-11})$$

Solution of equation (C-11) gives the desired solution, \mathbf{x}^* , and the Lagrange variables associated with the active constraints, c_i .

It can also be shown (Fletcher, 1987) that the Lagrange multiplier, λ_i , of the i^{th} constraint measures the rate of change in the objective function value relative to changes in that constraint function. Let us assume that λ_i is a negative number for $i \in A \cap I$, the set of active inequality constraints. This implies that if we move away from the i^{th} active inequality constraint in the direction $c_i > 0$ direction (which is feasible for $i \in A \cap I$), then the objective function decreases. But since at the minimum point \mathbf{x}^* , no further decrease of $f(\mathbf{x})$ is possible in the feasible region, the multiplier must have a non-negative value at the minimizer. Thus,

$$\text{at } \mathbf{x} = \mathbf{x}^*, \lambda_i^* \geq 0, \quad \forall i \in A \cap I \quad (\text{C-12})$$

This condition is very useful in evaluating the current set of active inequality constraints. Thus, Lagrange multipliers aid in identification of constraints, which are not binding at a given feasible point. This idea is further discussed in the section 3.

The above ideas embodied in equations (C-5) through (C-12) are summarized in form of a theorem. This is reproduced from Fletcher (1987).

Theorem for first order necessary conditions

If \mathbf{x}^* is a local minimizer of problem (1) and if a regularity condition holds (briefly, the constraints, c_i , must be independent; for details, see [1]) at \mathbf{x}^* , then there exist Lagrange multipliers λ^* such that \mathbf{x}^* , λ^* satisfy the following system:

$$\begin{aligned}\nabla_{\mathbf{x}}L(\mathbf{x}, \lambda) &= \mathbf{0} \\ c_i(\mathbf{x}) &= 0, \quad i \in E \\ c_i(\mathbf{x}) &= 0, \quad i \in I \\ \lambda_i &\geq 0, \quad i \in I \\ \lambda_i c_i(\mathbf{x}) &= 0, \quad \forall i\end{aligned}\tag{C-13}$$

These conditions are also referred to as Kuhn-Tucker conditions.

C-3 Active Set Method

Equality constraints force the minimum solution to lie on the intersection of the hypersurfaces of those constraints, since \mathbf{x}^* must satisfy $c_i(\mathbf{x}) = 0, \forall i \in E$. However, inequality constraints do not necessarily require the solution to exist on the hypersurface of those constraints, since $c_i(\mathbf{x}) \geq 0, \forall i \in I$. The Lagrange method discussed so far, involves active constraints (equality and active inequality constraints) only and searches along the intersection of these hypersurfaces.

The primal active set method describes a method for identifying a correct set of active inequality constraints and temporarily disregards the remaining inequality constraints. The active constraints are treated as equality constraints. With this information, one can then use the Kuhn-Tucker condition (equation (C-13)) to solve for

the desired solution. Checks are made to ensure that the obtained solution is feasible with respect to the constraints not in the active set. If the solution is infeasible, then a new set of active constraints is formed based on certain criteria. An algorithmic description of the procedure is described below.

The following algorithm is documented by Fletcher (1987). It begins by choosing certain inequality constraints as active and thus forming the active set $A^{(1)}$.

- (a) An initial feasible point, $\mathbf{x}^{(1)}$ is found which satisfies the active constraints in $A^{(1)}$.
Set $k=1$.
- (b) Let δ be defined by a shift of origin to $\mathbf{x}^{(1)}$. Now solve the quadratic program,

$$\begin{aligned} \underset{\delta}{\text{minimize}} \quad & f(\delta) = \frac{1}{2} \delta^T \mathbf{G} \delta + \mathbf{g}^{(k)T} \delta \\ \text{subject to:} \quad & \mathbf{a}_i^T \delta = \mathbf{b}_i \quad i \in A^{(k)} \end{aligned} \tag{C-14}$$

Note that the term $\mathbf{g}^{(k)}$ in the transformed coordinate system is $\mathbf{G}\mathbf{x}^{(k)} + \mathbf{g}$. If $\delta = \mathbf{0}$ does not solve equation (C-14) go to (d).

- (c) Compute Lagrange multipliers $\lambda^{(k)}$ and solve for the minimum value of active inequality constraint multiplier using,

$$\min_{i \in A \cup I} \lambda_i^{(k)} \tag{C-15}$$

Let λ_q represent the minimum value. If the minimum value is non-negative, then the solution $\mathbf{x}^{(k)}$ is a minimum and satisfies all constraints (see equation (C-12)).

Thus, set $\mathbf{x}^* = \mathbf{x}^{(k)}$ and terminate program. On the other hand, if the minimum value is negative, the q^{th} constraint is not binding and is removed from the current active set $A^{(k)}$.

- (d) Take a step in the direction of δ , by setting,

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)} \mathbf{s}^{(k)} \quad (\text{C-16})$$

where $\mathbf{s}^{(k)}$ represents the current search vector, δ . $\alpha^{(k)}$ is chosen such that the new location $\mathbf{x}^{(k+1)}$ lies on the set of active constraint as shown,

$$\alpha^{(k)} = \min \left(1, \min_{\substack{i, i \in A \\ \mathbf{a}_i^T \mathbf{s}^{(k)} < 0}} \frac{b_i - \mathbf{a}_i^T \mathbf{x}^{(k)}}{\mathbf{a}_i^T \mathbf{s}^{(k)}} \right) \quad (\text{C-17})$$

Let the minimum in the curly bracket be satisfied by the p^{th} constraint.

- (e) If $\alpha^{(k)} < 1$, add the p^{th} constraint to the active set A .
- (f) Set $k=k+1$, and go to (b).

An illustration of this algorithm is provided by applying it to the example problem presented in section C-1.3. We begin by choosing an initial feasible point at $\mathbf{x}^{(1)} = [3 \ 1]^T$. As shown in Figure C- 3, this starting point lies on constraint c_2 (see equation (C-3)) and is shown by the symbol 'o' on the plot. Thus, the index set of active constraints, $A^{(1)} = \{2\}$. Constraint c_1 is inactive at this time. Solution of equation (C-14) yielded the solution, $\delta^{(1)} = [-1 \ -1]^T$ in the transformed coordinate system or $[2 \ 0]^T$ in the x_1 - x_2

coordinate system (recall that we have shifted the origin to $\mathbf{x}^{(1)}$ before solving equation (C-14)). The symbol '□' represents this point in Figure C- 3.

The obtained solution indeed minimizes the objective function with respect to the active constraint c_2 . However, in doing so, constraint c_1 gets violated and hence the solution is infeasible. Thus, a line search is made in the direction of $\delta^{(1)}$ giving the best feasible point, $\mathbf{x}^{(2)} = [2+1/\sqrt{2} \quad 1/\sqrt{2}]^T$ depicted by the symbol '◇' in Figure C- 3. The best feasible point also corresponds to the point of intersection of the two linear constraints. Thus, at this point both the constraints become active, that is, $A^{(2)} = \{2,1\}$.

In the next step, we discover that λ_2 associated with constraint c_2 is negative. This implies that if we move away from this constraint in the feasible direction, we could further minimize the function, f . Thus, constraint c_2 is removed from the active set of constraints, resulting in $A^{(3)} = \{1\}$. In the final step, equation (C-14) is solved once again and we obtain the desired solution depicted by '*' in Figure C- 3. A summary of the iterative parameters is given in Table C- 1 below.

Table C-1

Description of parameters during stages of iteration. These can also be followed from Figure C- 3

Iteration, k	Iterate, $\mathbf{x}^{T(k)}$	Active set, $A^{(k)}$	$\alpha^{(k)}$	Multipliers, $\lambda_i^{T(k)}, i \in A$
1	[3 1]	{2}	0.2929	[-2]
2	[2.707 0.707]	{2, 1}	1.0000	[-2, 1.4142]
3	[2.707 0.707]	{1}	1.0000	[1.4142]
4	[1.7071 1.7071]	Converged	—	—

Thus, only four iterations were required to successfully terminate the algorithm. In fact, quadratic programs enjoy the finite termination property. This is briefly discussed by Fletcher (1987).

C-4 Qualifications and Conclusions

In this appendix, the use of active set method in solving quadratic programs is discussed. A simple example is used to illustrate the algorithm graphically. Most modern codes for quadratic programming are far more sophisticated. We solved equation (C-11) using standard inversion subroutine in MATLAB. In real applications, the problems may not

be numerically well conditioned and superior algorithms must be employed. Fletcher (1987) discusses a few. Further, if the Hessian matrix is indefinite, local minima may exist. This introduces extra complications in the algorithm. Also, most real applications involve a large number of decision variables and thus storage

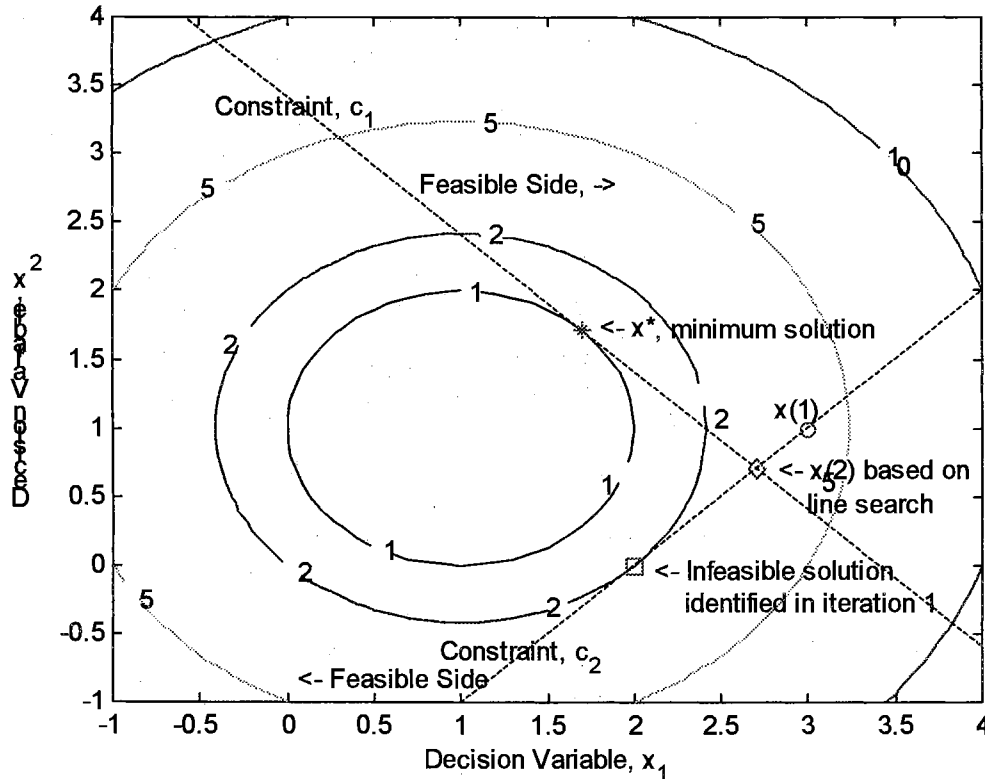


Figure C-3: Illustration of algorithm using Active Set method to solve a quadratic program. $x(k)$ represents iterates at different stages, k . Details of the calculations are provided in Table C-1.

concerns arise. Moreover, many control algorithms employ quadratic programming in real-time (for e.g. quadratic dynamic matrix controller), requiring efficient algorithms.

Quadratic programs enjoy a wide range of applications. In most instances, users buy well-written codes to solve the problem. In this work, the basic features of a quadratic program solver are discussed.

2
VITA

Sharad Bhartiya

Candidate for the Degree of

Doctor of Philosophy

Thesis: ENHANCEMENTS FOR MODEL PREDICTIVE CONTROL AND
INFERENCE MEASUREMENT

Major Field: Chemical Engineering

Biographical:

Personal Data: Born in Ribandar, Goa, India, On May 16, 1970, the son of Krishna and M.N Bhartiya.

Education: Graduated from Don Bosco High School, Panjim, Goa in May 1985; received Bachelor of Engineering degree in Chemical Engineering from Regional Engineering College, Durgapur, India in June 1991; received Master of Technology degree in Chemical Engineering from Indian Institute of Technology, Madras, India in January, 1993. Completed the requirements for the Doctor of Philosophy degree with a major in Chemical Engineering at Oklahoma State University in July, 2000.

Experience: Raised in a coastal town in Goa; employed by Galaxy Organics Pvt. Ltd., Tarapore as a production supervisor from 01/93 to 04/93; by Bhabha Atomic Research Centre, Bombay as a Scientific Officer from 05/93 to 07/95 and by Oklahoma State University, School of Chemical Engineering as a graduate research and teaching assistant from 08/95 to present.

Professional Memberships: AIChE