UNIVERSITY OF ALBERTA

DUAL-MODEL PREDICTIVE CONTROL

BY

KENT ZHIHUA QI ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of DOCTOR OF PHILOSOPHY

IN

PROCESS CONTROL

DEPARTMENT OF CHEMICAL AND MATERIALS ENGINEERING

EDMONTON, ALBERTA

SPRING, 1997

0-612-21621-7

UNIVERSITY OF ALBERTA

LIBRARY RELEASE FORM

NAME OF AUTHOR: **KENT ZHIHUA QI**
TITLE OF THESIS: **DUAL-MODEL PREDICTIVE CONTROL**
DEGREE: **DOCTOR OF PHILOSOPHY**
YEAR THIS DEGREE GRANTED: **1997**

Kent. Z. Qi

7804 - 152 C Ave.
Edmonton, Alberta
Canada T5C 3L5

Dated Jan. 6. 1997

UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommended to the FACULTY OF GRADUATE STUDIES AND RESEARCH for acceptance, a thesis entitled DUAL-MODEL PREDICTIVE CONTROL submitted by KENT ZHIHUA QI in partial fulfillment of the requirements for the degree of DOCTOR OF PHILOSOPHY in PROCESS CONTROL

....
Dr. D. G. Fisher (Supervisor)

....
Dr. S. L. Shah

....
Dr. F. Forbes

....
Dr. R. K. Wood

....
Dr. M. Q-H. Meng

....
Dr. M. Perrier (External Examiner)

Dated December 25, 1996

ABSTRACT

Model Predictive Control(MPC) uses a mathematical model of the process to predict the future process output trajectory. A multi-step optimization problem is then formulated which gives an optimal control action even in the presence of hard constraints. MPC is "optimal" in the sense of minimization of a user-specified performance index but the traditional design does not include any way of insuring stability and/or robustness (to model error). Three important issues are covered in this thesis: (1) process modelling; (2) predictive control design and tuning; (3) closed loop analysis of stability and robustness. All three are important requirements in both academic and industrial applications.

A dual-model formulation is developed to represent the process. The dual-model, specified in state space form, combines the advantages of both the Finite Step Response(FSR) and the Deterministic Auto Regression and Moving Average(DARMA) model forms. Defining the output predictions directly as the states, the dual-model generates explicit output predictions for use in the control calculations and is also convenient for expansion to multivariable process modelling and identification. Two important related issues, state estimation and parameter estimation, are discussed in detail. For optimal state estimation, the standard observer theory can be simplified when applied to the specific structure of the dual-model. For parameter identification, the extended Kalman filter gives predictive-control-relevant model identification based on experimental data.

Predictive control design includes many user specified control tuning parameters.

Physically, the general effect of traditional tuning parameters is intuitive and easily understood, but the selection of specific values is quite 'ad hoc'. A new method which results in better dynamic matrix conditioning is developed for choosing numerical values of tuning parameters. Then, two simple new tuning parameters, α and β, are introduced to fine tune the servo and regulatory performance respectively. With α and β, the controller can be adjusted *on-line* to obtain the best trade-off between robustness and servo/regulatory performance.

Two important closed loop system issues, robustness and constrained stability, are also discussed. Using the modelling errors in parametric form and matrix perturbation theory, simpler and less conservative robust stability criteria are developed by using the special structure of the dual-model formulation. When MPC has active constraints, its closed loop control structure is changed. The traditional constrained stability analysis procedures are applied and some difficulties in practical applications are explored. Then, two new approaches to handle hard constraints are proposed.

The theoretical research results developed in this thesis are incorporated into a new predictive control scheme, Dual-Model Predictive Control(DMPC). DMPC provides enhanced functionality (*e.g.* control-relevant identification, constraint handling) and more flexibility (*e.g.* MIMO identification and control) for practical control applications.

ACKNOWLEDGEMENT

I wish to express my sincere appreciation to my supervisor, Dr. D. Grant Fisher for his guidance, patience, help and encouragement throughout my Ph.D. Program.

I also would like to thank Dr. Sirish. L. Shah for his comments and excellent graduate courses in my area of study.

Financial support from NSERC grant held by Dr. D. G. Fisher, from the University of Alberta in the form of Province of Alberta Graduate Fellowships, and from the Department of Chemical and Materials Engineering in the form of Teaching Assistantships are also gratefully acknowledged.

Finally, I also wish to express my deep gratitude to my wife, Dian, for her longtime support, understanding, and sacrifice, and to our little son, Andrew, who has brought lots of fun and happiness to our lives.

Contents

List of Figures

List of Tables

List of Symbols and Abbreviations

1. Symbols

A	Dynamic matrix
A_a	Dynamic matrix with full horizon
A_c	System matrix of the closed-loop system
$A(q^{-1})$	Denominator polynomial
$B(q^{-1})$	Numerator polynomial
C	Constant column vector
$d(k)$	Process disturbances
$E(\omega)$	Model uncertainty matrix
G	Constraint matrix
G_a	Active constraint matrix
h_i	Impulse response coefficients
H	Output matrix
$H_{\Delta u}, h_{\Delta u}$	High constraint limits on Δu
H_u, h_u	High constraint limits on u
H_y, h_y	High constraint limits on y
k	Time step
K	State observer gain vector
K^{db}	Deadbeat state observer gain vector
K_{mpc}	State feedback gain vector of unconstrained MPC
K^{c}_{mpc}	State feedback gain vector of constrained MPC
K^{LP}_{mpc}	State feedback gain vector of MPC-LP
K_{FF}	Feedforward gain vector
$L_{\Delta u}, l_{\Delta u}$	Low constraint limits on Δu
L_u, l_u	Low constraint limits on u
L_y, l_y	Low constraint limits on y
M	Input horizon
n	Dual model order
N	Total number of step response coefficients
n_a	Order of the denominate polynomial in DARMA model
n_b	Order of the numerator polynomial in DARMA
n_r	Order of the slow mode denominate polynomial in DARMA
N_{FB}	Output feedback horizon
N_{FF}	Feedforward horizon
P	Output prediction horizon
S_i	Step response coefficients
T_s	Sampling interval

$u(k)$	Process input at k-instant
$U(k)$	Process input profile at k-instant
$\Delta u(k)$	Incremental control move at k-instant
$\Delta U(k)$	Incremental control move profile at k-instant
ΔU_1	Incremental control move, the range component
ΔU_2	Incremental control move, the null-space component
$v(k), w(k)$	Zero-mean white noise
$X(k)$	State variable vector
$Y(k)$	Process output at k-instant
$Y^{sp}(k)$	Setpoint profile at k-instant
$Y_m(\cdot\|k)$	Open loop model prediction at k-instant
$Y_p(k)$	Closed loop prediction at k-instant, $Y_p = Y_m + A\Delta U$
$Z(k)$	Augmented state vector
α	Fractional control horizon
β	Tuning parameter for state observer
γ_j	Output weighting factor on the j-th prediction
γ_s	Output steady state weighting factor
Γ	Disturbance matrix
λ	Control weighting factor
Δ	Backward shift operator, $\Delta = 1 - q^{-1}$
Φ	System matrix
Φ_p	Constant mapping matrix
θ	Control matrix

2. Abbreviations

AIC	Akaike's Information Criterion
ARMAX	Auto Regressive and Moving Average with eXogenous Input
ARX	Auto Regressive with eXogenous Input
AUDI	Augmented Upper Decomposition Identification
BLS	Batch Least Squares
DARMA	Deterministic Auto Regressive and Moving Average
DCS	Distributed Control System
DMC	Dynamic Matrix Control
DMPC	Dual-Model Predictive Control
EKF	Extended Kalman Filter
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
FSR	Finite Step Response
GMV	Generalized Minimum Variance (Control)
GPC	Generalized Predictive Control
IDCOM	Identification and Command (control)
KF	Kalman Filter
LS	Least Squares
LQC	Linear Quadratic Control
LRPI	Long Range Predictive Identification
MA	Moving Average
MIMO	Multi-Input Multi-Output
MPC	Model Predictive Control
MPC-LP	Model Predictive Control with Linear Objective Function
MPC-QP	Model Predictive Control with Quadratic Objective Function
MPM	Model Plant Mismatch
MV	Minimum Variance (Control)
NMP	Non-Minimum Phase
PFC	Predictive Feedforward Controller
PID	Proportional Integral Derivative (Control)
PRBS	Pseudo Random Binary Sequence
$\mathcal{RB}$	Robust Bound
RBS	Random Binary Sequence
RLS	Recursive Least Squares
SISO	Single-Input Single-Output

Chapter 1

Introduction

Facing ever stronger global competition, today's chemical industry is re-organizing to achieve higher production efficiencies. In addition to the development of new process technology with high production efficiency and minimal environmental impact, plant wide automation becomes an important player to ensure optimized operation. With better controlled performance, process operations can be pushed closer to their limit to obtain more benefits. Conventional controllers such as PID can not meet the challenge simply because they are single loop based control techniques which, because of the lack of communication and coordination with other control loops, are not able to provide plant wide optimization.

Multivariable control technology has been the subject of extensive research studies in the past few years. Successful applications include LQC, a state space formulation based optimal control strategy. LQC is generally regarded as a good control method for areas where fast control responses are crucial and the processes themselves can be described accurately. Therefore, they have been successfully applied in cost intensive projects such as aerospace, aircraft control and robotics. LQC is very sound in theory and comprehensive in practice. Unfortunately, chemical processes usually have different characteristics. LQC based control can not used easily to handle chemical process problems due to the model uncertainty, the diversified dynamics, and the economic operation requirement.

1. Model uncertainty: Usually, there are many uncertain phenomenon inside the chemical process and external disturbances so that accurate mathematical descriptions are not available. Many methods have been proposed to improve the robustness of LQC, *e.g.* the robust observer (Friedland 1986).

2. Diversified dynamics: A chemical plant typically consists of several control elements in the field where time-delay, large process time constants are common dynamics. Some elements can respond within seconds while some loop need hours to settle.

3. Minimum cost: LQC does not explicitly consider costs and benefits which are the highest priority for a chemical plant to stay in business.

In practice, to control the distributed equipment (in the field), a centralized computer control structure, Distributed Control System(DCS), has been the dominant process control strategy. Supervisory control has been used to coordinate the processing of individual controllers but typically the strategy uses the steady state relationships among variables. Strictly speaking, it is a multi-loop control not multivariable control. For example, some control loops have to be detuned to prevent multi-controllers fighting situations.

After several years of practical development and testing in industry, a new control strategy, Model Predictive Control(MPC), started to show its powerful potential for chemical process control. One of the most important concepts is the long range predictive feature of the controller. With prediction, the controller can forecast ahead to overcome the limitations from process time-delay, nonminimum phase and slow process dynamics. The control calculation is an optimization procedure involving matrix manipulations which can include multi-channel dynamics easily. Therefore, it is a multivariable control strategy. The costs of manipulated variables and the benefits of controlled variables can appear directly in the optimization objective. The control performance is then directly linked to how to operate the plant economically. Obviously, the rapidly advanced computer technology makes it possible to do complicated on-line mathematical manipulations such as process modelling, model prediction and constrained optimization. As the first commercial MPC control package, Dynamic Matrix Control(DMC) has been applied widely by many large scale chemical companies to handle multivariable control problem and to achieve plant wide optimization. Currently, there are over 15 commercial competitors in this field.

The initial industrial successes of the model predictive control technology was very impressive even without sound theoretical support. In the optimization formulation, MPC emphasizes obtaining the optimization solution and handling constraints, rather than good control performance. Many theoretical control considerations, from fundamental problems such as closed loop stability to advanced issues such as robustness, constrained stability, were unresolved. The obtaining of a control solution with or without constraints does not mean a stable control system. In 1989, the first successful attempt was made to translate the step response model based DMC into a state space formulation (Li, Lim & Fisher 1989). Then, in the state space domain, state observer theory (including Kalman filter theory) was used to estimate the future effects of unmeasured disturbances, *i.e.* the feedback scheme (Navratil, Lim & Fisher 1988). The unconstrained MPC control action was further classified into a state feedback controller format such that the stability of the MPC closed loop control system can be evaluated (Morari & Lee 1991). Because of the similarity between MPC and LQC, another research direction is to analyze MPC control systems by means of LQC theory. It has been considered as a special form of (finite horizon) LQC (Bitmead, Gevers & Wertz 1990) and the stability can be improved by carefully selecting the prediction horizons (Muske & Rawlings 1993). Nowadays, major research efforts are concentrating on the following three general areas — process modelling, system analysis and controller design and tuning.

1.1 Process Modelling and Identification

Model Predictive Control uses a mathematical model of the process dynamics to predict the future output trajectory. Then, a multi-step optimization algorithm is executed which gives an optimal solution with or without hard constraints. Finally, feedback is necessary for regulatory control in the presence of unmeasured disturbances. Therefore, MPC systems usually include three major components — the process predictor, the control move calculation and output feedback. These components all require an accurate mathematical description of the process, *i.e.* a process model.

Over the last 20 years, many types of predictive control schemes have been proposed, where the major difference is the type of model used to represent the process. Since the dynamics of the process are represented by different model formats, the subsequent procedures for the controller design are also different. For example, an FSR type step response model is used in Dynamic Matrix Control(DMC) (Cutler & Ramaker 1980), a DARMA model in Generalized Predictive Control(GPC) (Clarke & Mohtadi 1987), a state space model in State Space Model Predictive Control(SSMPC) (Ricker 1991). Recently, a Laguerre polynomial model has been proposed in the design of the predictive controller (Finn, Wahlberg & Ydstie 1993). However, there has been proven that the *nominal performance* of all linear predictive control schemes is not dependent on the model form used to represent the process (Morari, Garcia & Prett 1989). On the other hand, *robust performance* does depend on the model used since different model forms, due to their inherent nature, tend to emphasize different ranges of process dynamics. For example, the DARMA model emphasizes the middle frequency dynamic part of the process with more model errors at the steady state and in the time delay. Once the model type is defined, the controllable dynamic range of the process is roughly fixed too. Therefore, the key step in the MPC design is to choose an appropriate mathematical form for the predictive controller design.

Two most popular model formats - finite step response(FSR) model and DARMA model, have been successfully used in the MPC design. For example, DMC uses the FSR model and GPC uses the DARMA model. There has been a long and continuing debate regarding which model form is most suitable for practical applications. Conceptually, the FSR model has the advantages of easy model testing, identification and understanding/interpretation. However, it has two major problems that more model coefficients are required and it is unable to handle open-loop unstable processes. The former results in slow parameter estimate convergence so that it is not suitable for adaptive control applications. The DARMA model uses fewer parameters which facilitates on-line model identification and adaptive control. However, as mentioned above, model uncertainties could degrade the control performance.

A dual-model approach for the implementation of predictive control is proposed in this thesis which combines the two most popular model formats - FSR and DARMA models. In a state space form, it has a nonparametric model (e.g. step response coefficients) to handle the fast dynamics and a simple, compact DARMA model to represent the unstable or slower dynamics. In other words, (expressed in terms of the unit step response), the initial transient and the final approach to steady state (or unstable) parts of the process response are described by different models. The two model forms are combined in a single state space format — a Dual-Model state space formulation. The proposed Dual-Model Predictive Control (DMPC) algorithm includes the two most common control algorithms, DMC and GPC as subsets.

Obvious advantages of this dual-model structure include

1. Like GPC, it can be used for open loop unstable processes and can reduce the high DMC dimension required for effective control.

2. Like DMC, it uses an unstructured (non-parametric) step response model to describe the initial (fast) dynamics.

3. It has a solid theoretical foundation for multivariable process modelling, identification and control, due to the state space format;

In Chapter 2, the state definition and corresponding dual-model state space formulation are given followed by open loop performance analysis including observability and controllability analysis. Especially, from the controllability analysis, important properties of the future output predictions can be explicitly evaluated.

Once the model structure for the predictive control is selected, the model parameters can then be determined by either transforming from model parameters in other different but equivalent descriptions of the process, or estimated directly from input/output data obtained from open loop tests. The latter approach to obtain the model parameters, *i.e.* model identification, is especially important for the chemical process control since in most processes, good mathematical models are almost impossible to derive from first principles. Therefore, model identification has played a central important role in the process control applications. The developments of model based, (advanced) control algorithms, including model based PID tuning (*e.g.* the direct synthesis method (Seborg, Edgar & Mellichamp 1989) and the IMC method for PID tuning (Rivera & Morari 1986)), rely heavily on the accuracy of the model.

Model identification algorithms are numerical procedures which have their own dynamic characteristics being dependent upon many issues. For example, even with the same least squares objective function, different model structures give model parameters emphasizing different dynamic response ranges of the process. Step response model identification puts more emphases on the low frequency part (steady state part) of the process. DARMA models, on the other hand, include more fast-dynamic components. To achieve the best control performance, the dynamic characteristics of the model identification algorithms should match those of the control algorithms, *i.e.* control relevant identification.

Dynamic Matrix Identification(DMI) (Cutler & Yocum 1991) gives the least squares estimates of the step response coefficients for DMC. DMI is an inverse process of DMC and makes similar assumptions about the process disturbances, *e.g.* step type disturbances. With the objective function extended to cover the prediction errors, LRPI (Shook, Mohtadi & Shah 1991) estimates the DARMA model parameters to match the control requirement of GPC. In the dual-model state space formulation, due to the particular definitions of the state variables, an extended Kalman filter algorithm is used to obtain a predictive control-relevant identification method for DMPC. Related issues will be discussed in Chapter 3.

1.2 Predictive Controller Design

After defining the model and obtaining the model parameters, a predictive controller can be designed to calculate future control moves based on the information about future process output trajectories. The future outputs are predicted from the deterministic models of the process or measurable disturbances, or estimated from stochastic disturbances/noise. Therefore, the design procedure for predictive controllers includes feedback, feedforward and control calculations.

1.2.1 Predictive Control Calculations

Using the process model, the future behaviour of the process can be predicted and subsequent future control moves can be found to bring the process output to the desired trajectory. The task of the predictive controller is to find optimal future control moves considering process dynamics, economic costs and benefits, and the constraints on process inputs and outputs. Generally, it includes the selection of certain tuning parameters.

Ideally, the future process output should track the desired trajectory exactly. However, there are various restrictions which limit the input energy and practical considerations such as safety and process capacity. Therefore, there have been many intuitive and practical control parameters used in predictive controller designs. In the predictive domain, the concept of prediction horizons is used for both manipulated variables (as the control horizon, M) and controlled variables (as the prediction horizon, P). A similar predictive horizon concept for better disturbance rejection can be used for even the disturbance variables (Saudagar 1995). A general interpretation is that the controlled variables are optimized over P-steps (instead of one step inherently required by other control algorithms) using M ($M \leq P$) future moves. Other parameters include weighting on various terms in the objective function such as the control weighting, steady state weighting and output weighting. Conceptually, from the point of view of controller design, the introduction of these parameters is very intuitive and easy to understand. However, the selection of numerical values is quite 'ad hoc' since they do not have *unique* effects on the control system performance and/or robustness. For example, the same control performance can be obtained by choosing different combinations of controller parameters, but other factors such as robustness may be very different. Therefore, the selection of a particular control parameter depends on not only the process model parameters but other controller tuning parameters. Generally speaking, there is another degree of freedom to choose the best combination of the tuning parameters for the control design.

The control move calculation for predictive control involves a pseudo-inverse of the dynamic matrix, which consists of step response coefficients. Therefore, the numerical condition of this dynamic matrix is a very important factor for control performance. For example, an ill-conditioned dynamic matrix could lead to not only aggressive control action but also poor robustness since even a small model error could result in large changes to the control calculations. Better matrix conditioning can be obtained by several methods including the addition of control weighting which is equivalent to adding positive numbers to the diagonal elements of the dynamic matrix (Wilkinson, Morris & Tham 1994). An interesting extension of this finding is that all tuning parameters in predictive control can be interpreted as different approaches to improve the matrix condition. Therefore, Chapter 4 extends the concept of matrix conditioning to include the selection of all control parameters. The integer type control horizons are chosen by a special matrix decomposition scheme and the weighting are determined by modifying the elements of the dynamic matrix. With such an integrated approach, the best combination of the (conventional) tuning parameters can be determined to implement the predictive control algorithm. Final performance tuning is done using "dynamic tuning" as discussed below.

1.2.2 State Observer

The future output trajectory calculated from the process model is not the true process future output because of model uncertainties and unmodelled disturbances/noise. Feedback techniques must be used to estimate those effects based on currently available output measurements.

The predictive controller, with and/or without hard constraints, can be classified as one type of state feedback controller (Morari & Lee 1991, Qi & Fisher 1994, Oliveira & Biegler 1994). The controller gain is uniquely determined by the tuning parameters. In the state space domain, two other issues, state variable estimation and the handling of disturbances, are required for the calculation of control moves.

State observer theory is the natural basis for the estimation of the state variables from input and output measurements. The dynamics of unmeasurable disturbances and noise can be incorporated in the state space formulation such that Kalman Filter type, optimal state estimation can be obtained (Navratil et al. 1988). While the estimation algorithm itself is very straightforward, the convergence and optimality properties are major concerns for these estimation algorithms. Since the original state space formulation of MPC has a high dimension, direct applications of the optimal based state observer design to predictive control leads to an algebraic Riccati equation with a large dimension and consequently a complicated solution (Navratil et al. 1988). However, if the disturbance is pre-defined as integrated type white noise, a much simplified solution can be obtained for the MPC observer design (Lee, Morari & Garcia 1993). Lee *et al* also proved that the simplified observer can be applied to other kinds of disturbance models by imposing an additional filter in the estimation algorithm. Note that similar disturbance models and feedback design have been extensively studied for GPC (McIntosh, Shah & Fisher 1991), without the proof of optimality.

The application of the dual model representation definitely reduces the computational requirements if a Riccati equation solution is used. However, a different approach is used in Chapter 5 to design the state observer. This pole-placement based method gives a simpler solution for the deadbeat observer. Then, an additional parameter, β, to fine tune the state observer and to compensate disturbance models is introduced. This approach focuses on the convergence properties of the estimate such that effects of the disturbance horizon (Saudagar 1995) can be evaluated in detail.

1.2.3 Feedforward Controller

Different approaches should be used to handle measurable disturbances. The feedforward control scheme is especially useful since it does not affect the stability of the feedback control system and in ideal situations, can reject the disturbance perfectly. Traditionally, the measurable disturbances are treated the same way as the manipulated variables except that there are no future move assumptions. With deterministic disturbance models, the future output profiles due to past and current disturbances can be readily predicted. (In some applications, it is even possible to predict the future values of the disturbance itself, *e.g.* ambient temperature, based on past data). Then, these effects can be added to the error trajectory. By minimization, the MPC controller considers them as equivalent to setpoint changes and can make control moves to compensate the disturbances. A feedback observer must also be used to eliminate the effects of disturbances since, in practice, the feedforward controller can not totally reject them, and to handle unmeasured disturbances and design approximations.

The analysis in Chapter 5 first shows that the traditional way of handling measurable disturbances in MPC can not eliminate the disturbance effects in a feedforward sense. Then, a new feedforward control scheme, predictive feedforward, is proposed. With the help of the explicit state definition in the dual model formulation, different options to design the predictive feedforward control are developed.

1.3 Control System Analysis

MPC was originally developed from an optimization formulation which minimized the difference between the desired trajectory and the process output trajectory. As long as the minimization functions properly, MPC is *assumed* to make the process track the desired trajectory. However, the control performance can not be accurately determined unless a full closed loop system analysis is performed.

1.3.1 Robust Stability Analysis

The dynamics of complicated, multivariable processes can not be totally captured by a simple mathematical model. Model uncertainty is therefore unavoidable. The effect of modelling errors on the control performances of any type of controller, especially model predictive controllers, is very important. Usually, the effect of the model uncertainties on the (nominal) closed loop stability, *i.e.* robust stability, is of greatest concern in control system design.

Robust stability can be evaluated in either the time domain or the frequency domain, depending mainly on the ways used to represent the model uncertainties. There are two common methods used for the estimation of model uncertainties. The first one is to place the nominal model in parallel with the process and subject it to the same input excitation. Then, a residual signal is obtained by comparing the process output and the model outputs. Power spectrum based signal analysis can be used to estimate the residual dynamics which are a measure of the model plant mismatch(MPM). The MPM dynamics are represented in a nonparametric format which lumps all sources (disturbances, noise, MPM, etc) together in the frequency spectrum domain. Therefore, in terms of MPM estimation, signal processing based algorithms are very accurate. They have been used to evaluate the robustness of GPC successfully (Banerjee & Shah 1995). Other useful information about the model uncertainty comes from the parameter estimation algorithm which normally gives the estimated parameters as well as their confidence intervals, *i.e.* parameter uncertainties. Because of their parametric form, the model parameter uncertainties have a very clear physical meaning which is helpful in identifying the sources of the plant and model mismatches, *e.g.* in fault detection.

In the state space formulation, model uncertainties appear directly as perturbations of matrices. Depending on the nature of the uncertainties, the robust stability problem has been treated differently for unstructured and structured perturbations. There are many research results available in the literature, especially for the case of structured perturbations (Yedavalli 1985, Juang, Kuo & Hsu 1986, Kolla, Yedavalli & Farison 1989). Most of these studies have applied the Lyapunov equation in the time domain, or norm conditions in the frequency domain. A recent result is reported for a particular perturbation structure, interval matrices, where the coefficients of the matrices can be described by their upper and lower limits (Keel & Bhattacharyya 1995). This leads to a specialized formulation (structure) but most state space representations of physical systems fall into this category. Efficient and less conservative robust results can be obtained using this formulation.

The robustness analysis results are applied to evaluate the robust stability property of MPC in Chapter 6. Instead of developing new robustness criteria, the main objectives are to develop simpler and less conservative rules for robust predictive control design by taking the advantage of the special structure of the dual model representation. Then, the effects of the tuning parameters in the predictive controller on the robustness are investigated in detail.

1.3.2 Constrained Stability Analysis

One of the major advantages of MPC is its ability to incorporate hard constraints into the control calculations. When formulated as an optimization problem, the future control moves may be found (numerically) inside the feasible region formed by constraints on input and output variables. Even though more computational effort is required to handle constrained MPC, the calculation of the predictive controller itself is not a practical problem due to the rapid developments in both powerful computer hardware technology and efficient numerical software.

However, from the design perspective, the nominal predictive controller is usually designed and analyzed using an unconstrained control method, say the procedure discussed in Chapter 4. Then, hard constraints are considered in the optimization stage to calculate the control moves. Obviously, with the control energy restricted, the controller structure is no longer a simple state feedback controller. The optimal solution does not guarantee a stable closed loop control system. The stability of the closed loop system is no longer guaranteed even if the system is designed to be unconstrained stable. Actually, since the optimization would always generate a constrained control solution on the constraint boundary, the control structure of the constrained MPC could switch from one structure to another, *i.e.* piece-wise linear control structures. Every combination of active constraint boundaries corresponds to one possible control structure. The constrained stability problem becomes one of closed loop system stability with multiple linear controller structures. Usually, to guarantee the constrained stability, it is sufficient to require that all possible control structures be stable. Sufficient conditions have been reported in recent research (Zafiriou 1990, Zafiriou 1991, Zafiriou & Marchal 1991). A general conclusion is that active input constraints make the closed loop system open-loop. Active output constraints introduce extra feedback affecting the the stability of the closed loop system. The stability conditions are usually conservative since some control structures may never become active or occur just transiently in the whole control period. The literature results are difficult to understand and generally lead to a conservative controller design which avoids all possible constraint violations. To overcome the difficulty of constrained stability, several efforts have been made to relax the output constraints (Zafiriou & Hung-Wen 1993), or modify the original MPC objective function (Campo & Morari 1987, Rawlings & Muske 1993, Oliveira & Biegler 1994).

The constrained stability problem is reinvestigated using the dual model state space formulation in Chapter 7. The concept and procedure for constrained stability analysis are clarified and illustrated by examples, using the original MPC objective function. It is shown that traditional constrained stability analysis methods have too many possible controller structures to be examined and hence are not practical for the design of stable constrained predictive controllers. An important stability result is developed for MPC with a linear objective function. Then, practical approaches to reduce constraint inconsistencies are discussed and a new predictive control scheme defining the output weighting as functions of the constraint violation is proposed which not only simplifies the control calculation but facilitates the stability analysis. The constrained stability issues are discussed again in Chapter 8 as part of the dynamic tuning.

1.4 Controller Tuning

After the tuning parameters are selected, both analysis and simulation should be applied to ensure that stable and good control performance of the nominal system is achieved. Some parameters may need re-tuning. The most commonly adjusted parameter in predictive control applications is the control weighting because of its easily understood meaning and continuous adjustability.

Since complicated matrix manipulations of the dynamic matrix (which is a function of all the tuning parameters) and its pseudo inverse, are involved in the control calculation, it is not recommended to tune the control parameters on-line. The major reason is whenever a parameter is changed, the dynamic matrix and its inverse must be reconstructed or calculated *on-line*. This is not a problem for small applications (*e.g.* SISO), but causes big problems for large scale applications with many input and output variables. Actually, commercial control packages such as DMC calculates the required matrices and results off-line to save on-line computation effort and time. Obviously, the enhanced power of current computer technology is very helpful in terms of on-line tuning. For example, the latest version of DMC (*e.g.* Version 5.02) does allow on-line dynamic matrix calculation by introducing new concepts like dynamic weighting (Ishikawa, Baba, Miki, Ochi & Minter 1995).

Predictive controllers use both continuous and integer type parameters. The (integer) predictive horizon concept applied to both inputs and outputs is the fundamental advantage of MPC. Their effects on the control performance of MPC are very significant. However, in integer form, these crucial parameters can not be used to adjust the control performance smoothly. Robustness analysis also shows that their effects are not unique which increases the difficulty of on-line tuning. The earlier chapter of this thesis (Chapter 4) put forward a systematic method for selecting conventional tuning parameters. In Chapter 8, a new parameter, the fractional horizon, is introduced to improve the smoothness of the tuning. The resulting new control structure, the α-controller, is able to adjust the predictive control on-line easily and efficiently. It also leads to excellent stability and robust stability results. In the later part of Chapter 8, it is extended to handle hard constraints. *An important theoretical breakthrough is that constrained control stability can be absolutely guaranteed.*

1.5 Thesis Structure

The structure of this thesis is organized as follows. A new model structure, the dual-model state space formulation, is developed in Chapter 2. It is a general process description with better properties and more flexibility than previous model forms. Then, a Kalman filter based parameter estimation method is developed in Chapter 3. This model identification algorithm is essentially a predictive-control-relevant identification scheme. After obtaining the process model parameters, the nominal MPC controller is designed as outlined in Chapter 4 where it is shown that all MPC tuning parameters can be determined following one general approach. To handle disturbances, measurable or unmeasurable, predictive feedforward and state observer feedback are used as discussed in Chapter 5. Chapter 6 covers the robustness analysis issue by using matrix perturbation theory and the dual model formulation. The effects of hard input and output constraints on the closed loop stability are described in Chapter 7 for MPC with either a quadratic objective function (MPC-QP) or a linear objective function (MPC-LP). This eventually leads to an important stability result for MPC-LP. A new constraint handling scheme, dynamic weighting, to avoid the uncertain stability difficulty is also discussed. Chapter 8 introduces a new tuning parameter, α-controller, which can be used not only to continuously improve the control performance, off-line or on-line, but also to obtain an analytical solution for the constrained control problem. The final chapter, Chapter 9, summarizes the contributions of this thesis and points out potential extensions.

Simulation results using MATLAB software (Matlab 1989) are used to illustrate the development and performance of dual-model predictive control.

Chapter 2

Control-Relevant Process Modelling: A Dual-Model Formulation

2.1 Introduction

Several MPC schemes have been developed in the past fifteen years including the industrially popular Dynamic Matrix Control(DMC)(Cutler & Ramaker 1980, Garcia & Morshedi 1986) algorithm, the academic favourite adaptive Generalized Predictive Control(GPC) (Clarke & Mohtadi 1987) and the State Space Model Predictive Control(SSMPC)(Ricker 1991). They are all based on the same general concepts, *i.e.* finite output horizon for prediction, receding horizon control, and calculation of the control action to optimize a (constrained) quadratic or linear objective function. But they differ in the mathematical structures of the process model used to predict the future output trajectory.

For a given process, a number of alternative, but equivalent, input-output representations are possible. For example, in DMC, the process is characterized by its finite step response(FSR) expressed as a series of coefficients $\{s_i, i = 1, 2, \ldots, N\}$. For stable processes, N is normally selected such that s_N (or in some formulations, s_{N+1}) is equal to the steady state process response to a unit step input. GPC uses a parametric DARMA model which facilitates on-line identification for adaptive control applications. Even though the mathematical descriptions for a given linear process can be in different forms, it has been proven that the *nominal performance* of all predictive control schemes is not dependent on the model form used to represent the process (Morari et al. 1989). However, *robust performance* does depend on the model used since different model forms, due to their inherent natures, tend to emphasize different frequency ranges of the process dynamics. The step response model requires little prior knowledge about the plant. It can describe dynamic characteristics such as time delay, non-minimum zeros very well. But it is of high dimension and can only handle open-loop stable plants. On the other hand, the parametric DARMA model has low order and can describe more general processes but the structure and order of the process must be specified a priori which may introduce large modelling errors.

A new state space, "dual-model" approach has been proposed (Qi & Fisher 1993) which explicitly defines future output values as state variables. In fact, this "dual-model" structure uses two different model formats. It uses step response data $\{S_i, i = 1, 2, \ldots, n < N\}$ to characterize the first n points of the process step response and a low-order DARMA model to characterize the remaining $(N - n)$ points of the step response. The result is a non-minimal order, state space model that can be used as a basis for output prediction and/or controller design. The advantages of the dual-model approach are:

- it is general enough to handle most open-loop stable and/or unstable processes.
- the utilization of a low order DARMA form reduces the overall dimension of the step response formulation significantly.
- the high frequency dynamics, including any time-delay, can be represented effectively.
- it includes most published techniques as special cases and provides a link between GPC and MPC.
- the state variables have direct physical meaning which allows explicit evaluation of the predictions.
- since it is a state space format, it can be expanded to multivariable systems without any difficulty. Classical state space control techniques are also available for feedback observer design, stability analysis, *etc.*

Conceptually, the dual-model approach is similar to the orthogonal function based predictive control technique of Finn *et al* (1993) but it is more general.

This chapter is organized as follows: the model predictions using the step response model and DARMA model are given in Section 1. The recursive relation is built for predictions at the same time instant in Section 2. Comparison results show their properties and relations. Section 3 discusses several ways to select the order and combine the step response and DARMA models. The corresponding state space formulation using dual models is also presented in this section. Properties of the state variables derived from controllability analysis are presented in Section 4. Three typical examples in Section 5 show how to apply this new formulation.

2.2 Model Predictions

If a mathematical model of the process is available, the future behaviour of the process can be predicted.

2.2.1 Step Response Model Prediction

Output prediction using discrete step response model is based on *the superposition principle of linear systems* and can be expressed in the following convolution form:

$$\begin{aligned} Y(k+i \mid k+i) &= \sum_{j=1}^{\infty} S_j \Delta u(k+i-j) \\ &= \sum_{j=1}^{i} S_j \Delta u(k+i-j) + \sum_{j=i+1}^{\infty} S_j \Delta u(k+i-j) \\ &= \sum_{j=1}^{i} S_j \Delta u(k+i-j) + Y_m(k+i \mid k) \qquad (2.1) \\ i &= 0, 1, 2, \ldots \end{aligned}$$

where $\{S_j, j = 1, 2, \ldots\}$ are the discrete step response coefficients. k represents the current time instant and the contribution to the future output trajectory due to all past control actions is defined as

$$Y_m(k+i \mid k) = \sum_{j=1}^{\infty} S_{i+j} \Delta u(k-j) \qquad (2.2)$$

The notation $Z(\cdot|k)$ represents the prediction of $Z(\cdot)$ made at time k (and by implication including the effects of all inputs up to and including time $(k-1)$).

Then the recursive relationship between two consecutive time instants, k and $k-1$, is

$$\begin{aligned} Y_m(k+i \mid k) &= Y_m(k+i \mid k-1) + S_{i+1} \Delta u(k-1) \qquad (2.3) \\ i &= 0, 1, 2, \ldots, n \end{aligned}$$

where $n+1$ is the dimension of the prediction vector $Y_m(\cdot \mid k-1)$.

Equation (2.3) is a forward recursive relation so that the output predictions at instant k can be obtained using the predictions at $k-1$ plus control move $\Delta u(k-1)$. When $i = n$, the final element of the prediction vector, *i.e.* $Y_m(n \mid k)$, becomes

$$Y_m(k+n \mid k) = Y_m[(k-1) + (n+1) \mid k-1] + S_{n+1} \Delta u(k-1) \qquad (2.4)$$

where $Y_m[(k-1) + (n+1) \mid k-1]$ is the invalid, $(n+1)$-th element of the prediction vector. A new recursive relation is required to generate this term. In the standard step response formulation where $n = N$, this is accomplished by assuming that (Li et al. 1989) for a stable process

$$\begin{aligned} S_{N+1} &= S_N \\ Y_m[(k-1) + (N+1) \mid k-1] &= Y_m[(k-1) + N) \mid k-1] \end{aligned}$$

For an integrating process, Morari and Lee (1991) use

$$\begin{aligned} S_{N+1} &= 2S_N - S_{N-1} \\ Y_m[(k-1)+(N+1)|k-1] &= 2Y_m[(k-1)+N|k-1] - Y_m[(k-1)+(N-1)|k-1] \end{aligned}$$

The existence of a practical recursive relationship for general processes with $n << N$ is developed in the next section by using a DARMA model of the same process.

2.2.2 DARMA Prediction

The open loop process model in parametric DARMA form is

$$A(q^{-1})\Delta\, y(k) = B(q^{-1})\,\Delta u(k-1)$$

where

$$\begin{aligned} A(q^{-1}) &= 1 + a_1q^{-1} + a_2q^{-2} + \cdots + a_{n_a}q^{-n_a} \\ B(q^{-1}) &= b_0 + b_1q^{-1} + b_2q^{-2} + \cdots + b_{n_b}q^{-n_b} \end{aligned} \tag{2.5}$$

and the time-delay is included in the $B(q^{-1})$ polynomial.

Obviously, by definition, the step response coefficients $S(k)$ also satisfies the DARMA process model when the control action $u(k)$ is a unit step. (Note: both S_i and $S(i)$ refer to the step response coefficients here but S_i represents the original while $S(i)$ refers to the calculated values using the DARMA model).

Let the step input be:

$$u(k) = \{ \begin{matrix} 0 & k < 0 \\ 1 & k \geq 0 \end{matrix}$$

and

$$\Delta u(k-1) = \delta(k-1) = \{ \begin{matrix} 1 & k = 0 \\ 0 & k \neq 0 \end{matrix}$$

Then, the step response data $S(k)$ can be obtained by

$$\begin{aligned} A(q^{-1})\Delta S(k) &= B(q^{-1})\delta(k-1) \\ &= b_{k-1} \qquad (k \geq 1) \end{aligned} \tag{2.6}$$

Multiplying both sides of Equation (2.2) by $A\Delta$ and using (2.6) give

$$A\Delta Y_m(k+i \mid k) = \sum_{j=1}^{\infty} A\Delta S_{i+j}\Delta u(k-j) \tag{2.7}$$

$$= \sum_{j=1}^{\infty} b_{i+j-1}\Delta u(k-j) \tag{2.8}$$

Define the polynomial $B(q^{-1})$ as

$$B(q^{-1}) = \tilde{B}_i + q^{-i}\bar{B}_i$$

Then, Equation (2.7) can be written as

$$A\Delta Y_m(k+i \mid k) = \bar{B}_i\Delta u(k-1) \tag{2.9}$$

REMARKS:

1. The output prediction vector (the free response in GPC) assuming no further control action $\{\ \Delta u(k+i) = 0, i = 0, 1, 2, \dots\ \}$ can be calculated using the DARMA model in (2.9).

2. The prediction equation (2.9) is in an implicit form instead of the explicit form used by GPC.

   Actually, using a Diophantine identity, the implicit prediction equation can be easily transformed into the explicit form of GPC where

$$\frac{1}{A\Delta} = E_i + q^{-i}\frac{F_i}{A\Delta}$$
$$A\Delta = \frac{1 - q^{-i}F_i}{E_i}$$

   therefore

$$(1 - q^{-i}F_i)Y_m(k+i \mid k) = E_i\bar{B}_i\Delta u(k-1)$$
$$Y_m(k+i \mid k) = \bar{G}_i\Delta u(k-1) + F_iY_m(k \mid k) \tag{2.10}$$

   If the prediction for the current time $Y_m(k \mid k)$ is replaced by the current output measurement $Y(k)$, the prediction equation is identical to that used for GPC (McIntosh et al. 1991).

3. For most applications, n is much larger than n_b which leads to $\bar{B}_{n+1} = 0$. Therefore, the resulting recursive equation generated from (2.9) can be directly applied to replace the $n+1$ term in Equation (2.4). However, the full model structure of the $A(q^{-1})$ polynomial, *i.e.* the order and coefficients, must be known. Low order approximations of this polynomial should be considered.

2.2.3 Model Truncation and Integration

The prediction in Equation (2.9) is in a DARMA form and therefore requires knowledge of the structure of the parametric model. Compared with GPC methodology, it has no advantage. In order to keep the advantages of non-parametric model predictions, approximation methods should be considered.

Let n be the number of step response data used for prediction,

1. For any $n > n_b$, we have

$$\bar{B}_n = 0$$

   and Equation (2.9) becomes

$$A\Delta Y_m(k+n \mid k) = 0$$

   Considering the order of the state space formulation in Equation (2.3), this recursive relation can be applied when $n \geq max(n_a, n_b)$.

   Under this condition, only poles of the process (*i.e.* the poles of $A(q^{-1})$) must be identified. The zeros and time delay are defined by the non-parametric step response model.

2. Assume the polynomial $A(q^{-1})$ can be factored into two parts:

$$A(q^{-1}) = A_1(q^{-1})A_2(q^{-1})$$

where $A_1(q^{-1})$ consists of stable, fast mode(s) and $A_2(q^{-1})$ represents the slow or unstable mode(s).

Then, Equation (2.9) can be rewritten as

$$\begin{aligned} A_2 \Delta Y_m(k+n \mid k) &= (\frac{B}{A_1})_n \Delta u(k-1) \\ &= \bar{H}_n \Delta u(k-1) \end{aligned} \tag{2.11}$$

where $\bar{H}(q^{-1})$ is a polynomial which has coefficients equal to the impulse response of $\frac{B}{A_1}$ and

$$\begin{aligned} H(q^{-1}) &= h_0 + h_1 q^{-1} + h_2 q^{-2} + \cdots \\ &= \bar{H}_i + q^{-i}\bar{H}_i \end{aligned}$$

Since, by assumption, all roots of A_1 are within the unit circle and the fast modes decay quickly to zero, after n_h steps,

$$\lim_{n_h \to \infty} h_{n_h} = 0$$

Therefore, if n is large enough so that $h_n \approx 0$, Equation (2.11) reduces to

$$A_2 \Delta Y_m(k+n \mid k) \approx 0 \tag{2.12}$$

REMARKS:

(a) Only the slow mode(s) dynamics in polynomial $A_2(q^{-1})$ need to be estimated.

(b) Usually, the slow modes are relatively easy to identify using the final portion of the step response $\{S_i, i > n\}$ or by using various low pass filters with parameter estimation algorithms such as Least Squares.

(c) Most chemical processes are over-damped with time-delay. A first order ARX model is thus frequently sufficient to represent the dominant slow mode. The n step response coefficients can cover time-delay, non-minimum phase zeros effectively.

(d) For MIMO processes with fast loop(s) and slow loop(s), a similar procedure is also applicable.

The dual-model strategy is shown in Figure 2.1 where the initial time-delay, non-minimum phase, and fast dynamics are represented by the discrete step response coefficients. The number of step coefficients typically describes approximately a quarter of the time to steady state. The continuous part refers to the slow dynamics represented by a simple DARMA model and covers 3 quarters of the time span. This arrangement significantly reduces the number of step response coefficients required without sacrificing model accuracy.

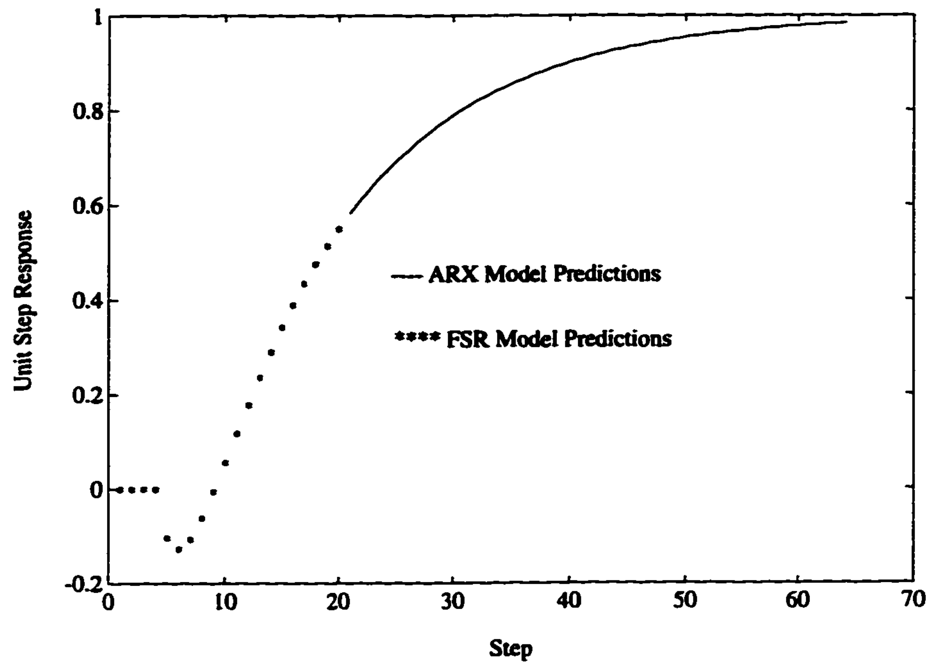

Figure 2.1:Dual Models for Dynamic Processes

2.3 The Dual-Model, State Space Formulation

The two model formats, step response and DARMA, are combined to describe the process dynamics. Future output predictions are obtained first from the step response model in the short range, Equation (2.3), and then extrapolated by the ARX model in the far range , Equation (2.12). The control move calculations are performed based on those predictions. However, the two input/output model descriptions have to be integrated into a compact model formulation for the purposes of further analysis and feedback design.

2.3.1 Definition of The State Variables

The future output trajectory $Y_m(\cdot \mid k)$ plays a crucial role in MPC especially when the process is subjected to disturbances and noise. Explicit expression of the trajectory facilitates the estimation algorithms as well as the evaluation of the prediction properties. Therefore, the state variables are defined here as

$$\begin{aligned} X(k) &= [Y_m(k \mid k)\ Y_m(k+1 \mid k)\ \cdots\ Y_m(k+n \mid k)]^T_{(n+1)\times 1} \\ X(k-1) &= [Y_m(k-1 \mid k-1)\ Y_m(k \mid k-1)\ \cdots\ Y_m(k+n-1 \mid k-1)]^T_{(n+1)\times 1} \end{aligned}$$

Note that the dimension of the state variable $X(k)$ is equal to the user-chosen value n rather than N which is determined by the number of points required to define the process step response to final steady state.

2.3.2 The State Space Formulation

By combining Equation (2.3), (2.4), (2.12) and the state vector definition, the state space model format of the process can be written as:

$$\begin{aligned} X(k) &= \Phi X(k-1) + \theta \Delta u(k-1) \\ Y(k) &= H X(k) \end{aligned} \qquad (2.13)$$

where

$$\Phi = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & 0 & \cdots & 0 & r_{n_r} & \cdots & r_2 & r_1 \end{bmatrix}$$

$$\theta = \begin{bmatrix} S_1 & S_2 & S_3 & \cdots & S_{n+1} \end{bmatrix}^T$$

$$H = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

where $\{ r_i, \ i = 1, 2, \ldots, n_r \}$ are the coefficients of $A_2\Delta$ defined by

$$A_2\Delta = 1 - r_1 q^{-1} - r_2 q^{-2} - \cdots - r_{n_r} q^{-n_r}$$

REMARKS:

1. The structure of the dual-model formulation of MPC in (2.13) is a standard state space realization in canonical form, and is recommended by many researchers for multivariable model identification (Guidorzi 1981).
2. The dual-model structure includes both an n-th order FSR model and a low order ARX model.
3. The FSR model defines the time-delay and nonminimum phase dynamics.
4. The dual-model structure only uses the AR part of the parametric model so that time delay, zeros, steady state gain, etc are not strict concerns of the DARMA model estimation algorithms.
5. The AR part of the parametric model can describe open loop unstable process dynamics and dramatically reduces the number of FSR coefficients required for effective control, *e.g.* from 40 to 10.
6. Since it is in state space form, the prediction equation can be easily extended for multi-input, multi-output processes by augmenting the state vector $X(k)$. The system matrix Φ and input matrix θ can be extended correspondingly and have a similar dual-model structure.

7. The state variables can be directly used in the calculation of control moves. For simplicity of model expression, the output prediction horizon P is assumed to be less than the model order n of Equation (2.13). This assumption is quite reasonable in practice though it can be removed without affecting the structure of the open loop model predictor.

   Assuming the prediction horizon P is greater than the model order n, the model predictions, $Y_m(k+i|k)$, $(i = 0, 1, \ldots, n, n+1, \ldots, P)$, can be obtained using the standard state space prediction given by Ricker (1991) as:

$$\begin{bmatrix} Y_m(k+1|k) \\ \vdots \\ Y_m(k+n|k) \\ Y_m(k+n+1|k) \\ \vdots \\ Y_m(k+P|k) \end{bmatrix} = \begin{bmatrix} H\Phi \\ H(\Phi^2+\Phi) \\ \vdots \\ H\sum_{i=1}^{n}\Phi^i \\ H\sum_{i=1}^{n+1}\Phi^i \\ \vdots \\ H\sum_{i=1}^{P}\Phi^i \end{bmatrix} \Delta X(k) = \begin{bmatrix} H\Phi \\ H\Phi^2 \\ \vdots \\ H\Phi^n \\ H\Phi^{n+1} \\ \vdots \\ H\Phi^P \end{bmatrix} X(k)$$

   With the addition of the current output prediction $Y_m(k|k) = HX(k)$, the above prediction vector becomes

$$\begin{bmatrix} Y_m(k|k) \\ Y_m(k+1|k) \\ \vdots \\ Y_m(k+n|k) \\ Y_m(k+n+1|k) \\ \vdots \\ Y_m(k+P|k) \end{bmatrix} = \begin{bmatrix} H \\ H\Phi \\ H\Phi^2 \\ \vdots \\ H\Phi^n \\ H\Phi^{n+1} \\ \vdots \\ H\Phi^P \end{bmatrix} X(k)$$

   Using the special structures of Φ and H in the dual-model equation (2.13), it is easy to verify that

$$\begin{bmatrix} H \\ H\Phi \\ H\Phi^2 \\ \vdots \\ H\Phi^n \end{bmatrix} = I_{(n+1)}$$

   Therefore, the prediction vector $Y_m(\cdot|k)$ can be partitioned as two parts, $Y_{m,1}(k)$ and $Y_{m,2}(k)$, where

$$
\begin{aligned}
Y_{m,1}(k) &= \begin{bmatrix} Y_m(k|k) \\ Y_m(k+1|k) \\ \vdots \\ Y_m(k+n|k) \end{bmatrix} = X(k) \\
Y_{m,2}(k) &= \begin{bmatrix} Y_m(k+n+1|k) \\ Y_m(k+n+2|k) \\ \vdots \\ Y_m(k+P|k) \end{bmatrix} = \begin{bmatrix} H\Phi^{n+1} \\ H\Phi^{n+2} \\ \vdots \\ H\Phi^{P} \end{bmatrix} X(k)
\end{aligned}
$$

Obviously, after obtaining the values of state variables from the dual-model or the state observer discussed later, future predictions can be calculated as above. Note that in this situation, the prediction vector $Y_{m,2}(k)$ is not directly estimated but calculated which strongly depends on the process model parameters (*e.g.* r_i). Model uncertainties may cause prediction errors. The MPC based on the minimal order state space formulation, *e.g.* SSMPC (Ricker 1991), has a similar problem. Direct estimation of the prediction vector is one of advantages of the dual-model state space formulation.

2.4 Special Cases of The Dual-Model Formulation

A large number of model predictive control techniques have been presented in which different process models have been used. The following sections show that the dual-model formulation includes several of them as special cases.

2.4.1 Full Step Response Model(DMC)

The dual-model formulation approaches the full order MPC step response formulation as $n \to N$, since

$$A_2\Delta = \Delta = 1 - q^{-1}$$

i.e.

$$r_1 = 1, n_r = 1$$

which is exactly the full-order MPC(DMC) state space formulation (Li et al. 1989).

2.4.2 Dual-Model in a Block-Companion Canonical Structure

Taking $n = n_a$ and assuming $n_a \geq n_b$, the prediction formulation can be written as

$$\Phi = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ r_{n_a+1} & r_{n_a} & r_{n_a-1} & \cdots & r_1 \end{bmatrix}$$

$$\theta = \begin{bmatrix} S_1 & S_2 & S_3 & \cdots & S_{n_a+1} \end{bmatrix}^T$$

$$H = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

where $\{ r_i, i = 1, 2, \ldots, n_a + 1\}$ are coefficients of $A\Delta$ and have a very simple relationship to the parameters in the DARMA model since $\{r_i = \Delta a_i = a_i - a_{i-1}, i = 1, 2, \ldots, n_a + 1\}$ where $a_0 = 1, a_{n_a+1} = 0$.

The state space formulation, in block-companion canonical form, gives a canonical realization of the DARMA model used by GPC.

2.4.3 State Space Model for MPC(SSMPC)

Directly employing a state space model to represent processes in MPC where the open loop model is minimum-order has been recommended by Ricker (1991). Recall that as discussed in Section (2.4.2), the dual-model formulation can provide a block-companion canonical structure which is also in minimum-order state space form. Although the model parameters may differ due to different state variable definitions, these two models are similarity equivalent.

2.4.4 Predictive Control with Steady State Weighting

It is often important to include a steady state prediction in predictive control *e.g.* (Kwok & Shah 1994, Saudagar 1995). The prediction equation (2.13) can be modified to include a steady state prediction as an extra variable. The new state vector can be defined as:

$$X(k) = [Y_m(k \mid k) \; Y_m(k+1 \mid k) \; \cdots \; Y_m(k+n \mid k) \; Y_m(ss \mid k)]^T_{(n+2)\times 1} \tag{2.14}$$

where $Y_m(ss \mid k)$ represents the steady state output prediction based on all past control actions.

The state space form of the MPC predictor with steady state prediction has the same structure as (2.13) with:

$$\Phi = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & r_{n_r} & \cdots & r_2 & r_1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & 1 \end{bmatrix}_{(n+2)\times(n+2)}$$

$$\theta = \begin{bmatrix} S_1 & S_2 & S_3 & \cdots & S_{n+1} & S_{ss} \end{bmatrix}^T$$

2.4.5 Dual-Model for Integrating Processes

A first-order integrating process has a pole at $q = 1$ and the unstable mode is described by $A_2 = 1 - q^{-1}$. Thus

$$A_2\Delta = 1 - 2q^{-1} + q^{-2}$$

i.e.

$$r_1 = 2, r_2 = -1, n_r = 2$$

This is exactly the structure suggested by Morari and Lee (1991).

2.4.6 Dual-Model for Exponentially Unstable Process

When a process has pole(s) located on the real axis outside the unit circle, it is exponentially unstable. The unstable mode(assuming only one such pole for simplicity) can be represented by $A_2 = 1 - pq^{-1}$, where $p > 1$. Then

$$A_2\Delta = 1 - (p+1)q^{-1} + pq^{-2}$$

i.e.

$$r_1 = p + 1, r_2 = -p, n_r = 2$$

2.4.7 Dual-Model for Slow, Lightly-Damped Process

The parametric model in this dual-model approach can also be a stable mode. For example, if a process consists of fast mode(s) but includes a dominate pair of slow, lightly-damped modes such as $A_2 = (1 - (\sigma + j\omega)q^{-1})(1 - (\sigma - j\omega)q^{-1})$, Then

$$A_2\Delta = 1 - 3\sigma q^{-1} + (\sigma^2 + \omega^2 + 2\sigma)q^{-2} - (\sigma^2 + \omega^2)q^{-3}$$

i.e.

$$r_1 = 3\sigma, r_2 = -(\sigma^2 + \omega^2 + 2\sigma), r_3 = \sigma^2 + \omega^2, n_r = 3$$

2.4.8 Dual-Model with Known Time-delay

Time delays appear as d zero terms in the control vector θ. This not only increases the dimension of the state space formulation but also makes the first d elements of the state vector uncontrollable. Usually, if the time delay is known, predictive control algorithms consider only the output predictions after the initial dead-zone. To accommodate a known time delay, the state variables can be redefined as:

$$X(k) = [Y_m(k+d \mid k) \; Y_m(k+d+1 \mid k) \; \cdots \; Y_m(k+d+n \mid k)]^T_{(n+1)\times 1} \tag{2.15}$$

and the state space formulation has the same structure as before.

2.4.9 Dual-Model with Absolute Input

By replacing the $\Delta u(k-1)$ by $u(k-1)$, the dual-model formulation becomes

$$\begin{aligned} X(k) &= \Phi X(k-1) + \theta u(k-1) \\ Y(k) &= H X(k) \end{aligned} \tag{2.16}$$

Note that it is slightly different than the original dual model formulation in which the absolute input $u(k)$ is used instead of the incremental input $\Delta u(k)$. This arrangement does not change the dual model structure but can improve the parameter estimation performance as discussed later in Chapter 3. The state definition remains the same but matrices Φ, θ include different coefficients. θ, is the Finite Impulse Response(FIR) coefficients h_i (instead of FSR). The last row elements of Φ is determined by the coefficients of the polynomial $A_2(q^{-1})$ (instead of ΔA_2).

2.5 Properties of The State Variables

The state variables in the dual model state space formulation are not directly measurable but can be easily estimated from the output measurements since they are completely observable. It is easily proven that the observability matrix is an identity matrix, *i.e.*

$$\begin{bmatrix} H \\ H\Phi \\ H\Phi^2 \\ \vdots \\ H\Phi^n \end{bmatrix} = I_{(n+1)}$$

This result even holds for MIMO processes. Therefore, each individual state variable can be estimated with equal weighting by a state observer (Qi & Fisher 1993). The equally weighted state formulation is also very helpful for the estimation of model parameters, which will be discussed in the next chapter.

On the other hand, it is also important how the state variables (*i.e.* the future output values) are affected by changing the input sequence, *i.e.* the controllability of the state variables. This is a fundamental property for several control design methods, *e.g.* the pole-placement technique.

2.5.1 Complete Controllability Analysis

The controllability of future outputs in Equation (2.13) can be analyzed by evaluating the condition of the controllability matrix:

$$\mathcal{C} = [\theta,\ \Phi\theta,\ \cdots,\ \Phi^n\theta]$$

1. Non-invertibility with Time Delay:

   Obviously, if there is any time delay in the control vector θ, the first several elements of the output predictions can not be controlled by the current input action. Therefore, in formulating the MPC objective function, it is not desirable to start the optimal minimization from time one. An initial prediction horizon (called N_1 in GPC) is commonly used in the design of predictive controller, where N_1 is greater than, or equal to, the time delay plus one.

2. Deficient Rank or Ill-Conditioned:

   Excluding the time delay, it is easy to show that the controllability matrix is full-rank. However, it may be a matrix with very poor conditioning. For example, consider a dual-model formulation for a third order process as follows.

$$\text{Process} \qquad 1/(s+1)(3s+1)(5s+1) \tag{2.17}$$

Dual-Model $\qquad T_s = 1, n = 11$

$$\Phi = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 0.2158 & -1.1514 & 1.9031 \end{bmatrix}_{12\times 12}$$

$$\theta = \begin{bmatrix} 0.0077 & 0.0358 & 0.0629 & \cdots & 0.0539 & 0.0465 \end{bmatrix}^T$$

Controllability Matrix $\mathcal{C}$

$$\det(\mathcal{C}) = 1.18 \times 10^{-65}$$
$$\text{cond}(\mathcal{C}) = 3.42 \times 10^{10}$$
$$\text{rank}(\mathcal{C}) = 12$$

If only the rank is considered, the controllability matrix is full rank ($= 12$). However, the condition of this matrix is very bad ($= 3.42 \times 10^{10}$). Obviously, with the same control effort, the states (*i.e. points in the output trajectory*) can not be adjusted equally. Some linear combinations of the state variables are almost totally out of control. In terms of open loop poles, some poles are easy to shift arbitrarily while others are very hard to shift by state feedback control. There is no serious problem if all the uncontrollable poles are stable since they will eventually decay even without external inputs. However, the dominant poles and/or steady state must be controllable for effective closed loop control, *i.e.* to solve the stabilizability problem.

2.5.2 Controllability of Dominant Poles and Steady State

As defined in the dual-model formulation, the open-loop dominant poles are captured by the slow ARX model in the system matrix Φ. The non-minimum, dual-model state space formulation can then be transformed into two sub-systems, controllable and uncontrollable, by similarity transformations.

Define

$$\bar{X}(k) = TX(k) = \begin{bmatrix} \bar{X}_{\bar{c}}(k) \\ \bar{X}_c(k) \end{bmatrix}$$

$$\bar{\Phi} = T\Phi T^{-1} = \begin{bmatrix} \bar{\Phi}_{\bar{c}} & 0 \\ \bar{\Phi}_{12} & \bar{\Phi}_c \end{bmatrix}$$

$$\bar{\theta} = T\theta = \begin{bmatrix} 0 \\ \bar{\theta}_c \end{bmatrix}$$

$$\bar{H} = HT^{-1}$$

where $\bar{X}_c(k)$ refers to the controllable sub-system with system matrix $\bar{\Phi}_c$ and control vector $\bar{\theta}_c$. The $\bar{X}_{\bar{c}}(k)$ is the uncontrollable sub-system with system matrix $\bar{\Phi}_{\bar{c}}$ and no control vector ($= 0$).

For example, applying a similarity transformation to the process in (2.17), yields

$$\bar{\Phi}_c = \begin{bmatrix} 0.1397 & 0.1617 & 0.0000 \\ -0.6184 & 0.7701 & 0.2028 \\ 0.1227 & -0.1979 & 0.9934 \end{bmatrix}$$

$$\bar{\theta}_c = \begin{bmatrix} 0.00 & 0.00 & 0.2330 \end{bmatrix}^T$$

The eigenvalues of $\bar{\Phi}_c$ are $0.3679, 0.7165, 0.8187$. Therefore, the three dominant poles of the open loop process are controllable.

The steady state equation can be easily separated from other state equations as:

$$x_{ss}(k+1) = x_{ss}(k) + S_{ss}\Delta u(k)$$

It is obviously controllable with respect to $\Delta u(k)$. However, note that it is not controllable by $u(k)$ since $\Delta S_{ss} = 0$. In other words, an integral term must be used for steady state control which is well known in the context of PID controller applications.

2.6 Comments on State and Parameter Estimation

The discussion in the last section showed that it is possible to reduce the order of the MPC state space formulation by using a dual-model. This raises the problem of how to get the parameters of the ARX model that is used to represent the slow modes of the plant. Three methods can be applied for this purpose:

1. **Input/Output Model:** The model parameters in a input/output process model can be directly converted to the model coefficients in the dual model. For example, the $A(q^{-1})$ polynomial of a DARMA model can be used as the last row elements of Φ, with or without fast and slow mode decomposition. In this way, the predictive controller based on the dual-model is totally equivalent to GPC;

2. **Extended Kalman Filter:** This approach uses a pre-defined dual-model structure first. Then, the state space model parameters are estimated by plant testing;

3. **Step Response Test:** While the first few points can be used as the elements in θ, the last part of the step response coefficients can be 'curve-fit' to yield a recursive AR model.

Here, the discussion is restricted to the third method starting from the given step response data which, for example, is obtained by either direct step excitation of the plant or other techniques such as DMI (Cutler & Yocum 1991). The main purpose is to enhance the understanding of the dual-model structure by means of the explicit step response coefficients. A more general model parameter identification scheme using an Extended Kalman Filter will be discussed in the next chapter.

2.6.1 Division Point n

The division point for the dual-model formulation (*i.e.* n) is determined by the user. This point will separate the full step response data into an initial stage and a final stage. Further, this point also assumes that the effect of the fast modes (*i.e.* B/A_1 in Equation (2.11)) have decayed to zero and the rest of the step response points (*i.e.* the final stage) can be completely determined by a simple DARMA model. Obviously, the larger the value of n, the higher the order of DMPC state formulation. At the other extreme, if $n = max(n_a, n_b)$, all the step response data would be fitted by the DARMA model which could result in a large error if the model structure is not chosen appropriately.

2.6.2 Estimation Algorithm

The parameters for the DARMA model that fits the final stage of the step response transient can be easily obtained by any identification algorithm. The batch least squares method or AUDI (Niu, Fisher & Xiao 1992*a*) is ideal for this problem because of its simplicity and good properties. By the step type excitation, all control inputs are equal to one unit for this final part of the system response. The estimation algorithm does not, in general, estimate the time delay of the process. As shown before, the last row of system matrix needs only the coefficients of an AR model rather than those of the MA part. Therefore, the problems associated with LS estimation method under non-rich excitation do not affect the application of this algorithm in the dual-model state space method.

2.6.3 Smoothness Constraint

Another important feature of the parameter estimation method should be mentioned here. In addition to estimating the coefficients of a DARMA model, the connection of these two models should be considered. In fact, the structure of the dual-model does allow a overlap of those two models such that the model predictions at time *n+j, j=1,2,...* have a smooth transition for practical applications. This means that at the division point n, in addition to the continuity condition $S_n = S(n)$, the derivative should also be kept constant. This requires that the division point n should be chosen in a region where the slope changes smoothly and that a constraint on the slope should be added to the parameter estimation algorithm.

2.6.4 Effect of Noise

Under ideal conditions, the estimated parameters are very close to the true values if the division point, *i.e.* n, is chosen properly. For example, the process

$$G(s) = \frac{1}{(s+0.2)(s^2+s+1)}$$

consists of a slow overdamped mode and a fast oscillatory mode. Assuming a first order model for estimation, only two parameters are required and Figure 2.2 shows the estimated normalized parameters as a function of the truncation point. Note that the parameter estimation in the denominator polynomial $A(q^{-1})$ converges very quickly while that of the numerator polynomial $B(q^{-1})$ is slower. When $n > 25$, the contribution of the fast, underdamped mode to the overall system step response can be neglected, and the coefficients in the AR part of the model can be easily estimated correctly.

When the step response data are contaminated by noise, the signal-to-noise ratio should also be considered. For uniformly distributed, zero-mean noise added to the step response data, the selection of large n can reduce the effects of the fast mode. However, as the process approaches steady state, the changes in the output signal become smaller. Therefore, the signal-to-noise ratio(SNR) of the final stage of the step response data (*i.e.* $N - n$ points) tends to be smaller. This results in more estimation error. For example, Figure 2.3 shows the parameter estimation trajectory of the first order DARMA model of the above process when uniformly distributed noise with variance 0.1 is added to the step response data. The parameters in the MA part change significantly (*i.e.* are sensitive to noise) while those in the AR part show only small changes. Fortunately, the most commonly occurring noise in industrial applications is at high frequencies which affects the estimation of high-frequency (fast) dynamic modes much more than it does the estimation of the slower modes since noise can be filtered or averaged out. An advantage of the dual-model representation is that only the less noise sensitive AR part of the model is used for the dual-model coefficients.

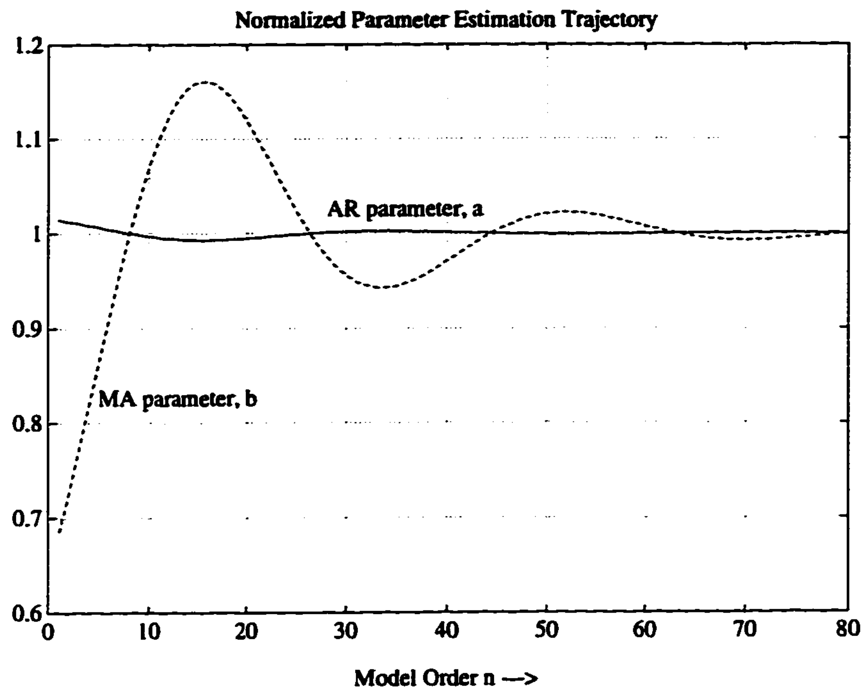

Figure 2.2: DARMA Parameter Estimates from Step Response Coefficients

2.7 Case Studies

This section presents two examples based on ideal situations, *i.e.* there is no noise or error in the step response data. A constrained BLS algorithm is applied to the final stage step response data to obtain the slow mode(s) DARMA model. The effects of the model parameters on the open-loop predictions are evaluated by the setpoint tracking performance of an MPC control system. Tuning parameters in the MPC algorithm are specified as $M = 1, P = 10$ and without any constraints. Initially, external output feedback is not used. Then, the effects of feedback options show that for some cases, it is necessary to include feedback to compensate for modelling errors which cause instability problems - *internal model instability*.

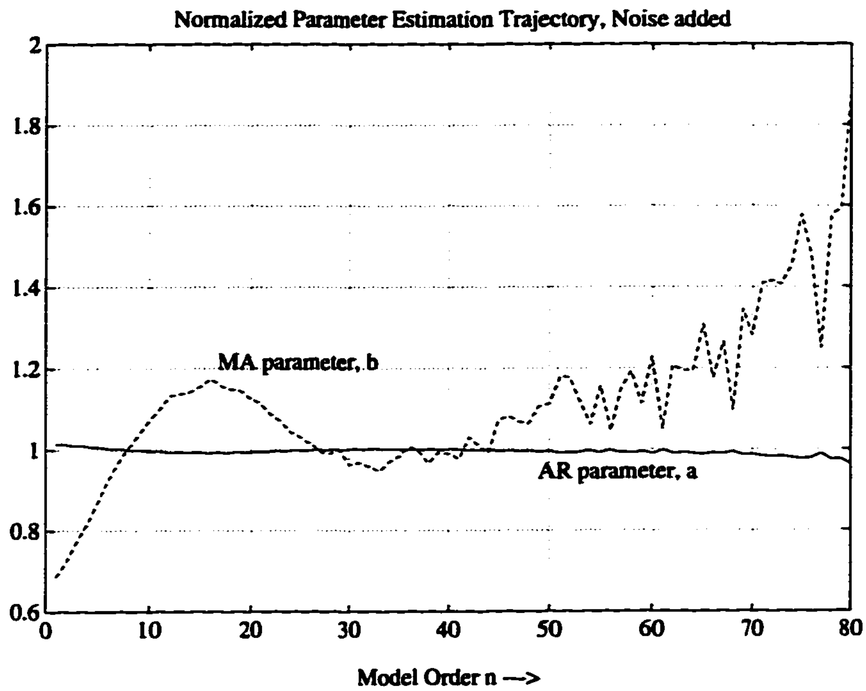

Figure 2.3:The Effect of Noise on Parameter Estimation

2.7.1 Case 1: Open Loop Stable Underdamped Process

The first example is an underdamped 3rd order process

$$G(s) = \frac{1}{(s+0.2)(s^2+s+1)}$$

which has fast under-damped mode(s) $s = -0.5 \pm 0.86i$ and a slow mode $s = -0.2$.

With the sampling interval $T_s = 0.2$, the process response reaches steady state within 120 steps. This means the dimension of a full state space formulation (Li et al. 1989) would be at least 121. However, since the fast mode decays to zero in 30 steps, the dimension of the reduced dual-model formulation can be only one quarter of the full sized MPC.

With $n = 30$, the remainder of the 90 points of the step response data can be fit by a first order model where

$$A(q^{-1}) = 1 - 0.9638q^{-1}$$

$$B(q^{-1}) = 0.1823q^{-1}$$

The true discrete model corresponding to the slow mode $1/(s+0.2)$ is $A(q^{-1}) = 1 - 0.9608q^{-1}$ and $B(q^{-1}) = 0.1961q^{-1}$ so the fitted response is very close. In practical applications where the "slow response" is higher-order, then the fitting of a first-order model implies an approximation. (Note that the delay is included in the initial step response data and hence does not occur in the DARMA model).

The open loop control results are shown in Figure 2.4. Note that since there is no output feedback, the small gain mismatch in the DARMA model results in small steady state error. Closed loop MPC would eliminate this error completely.

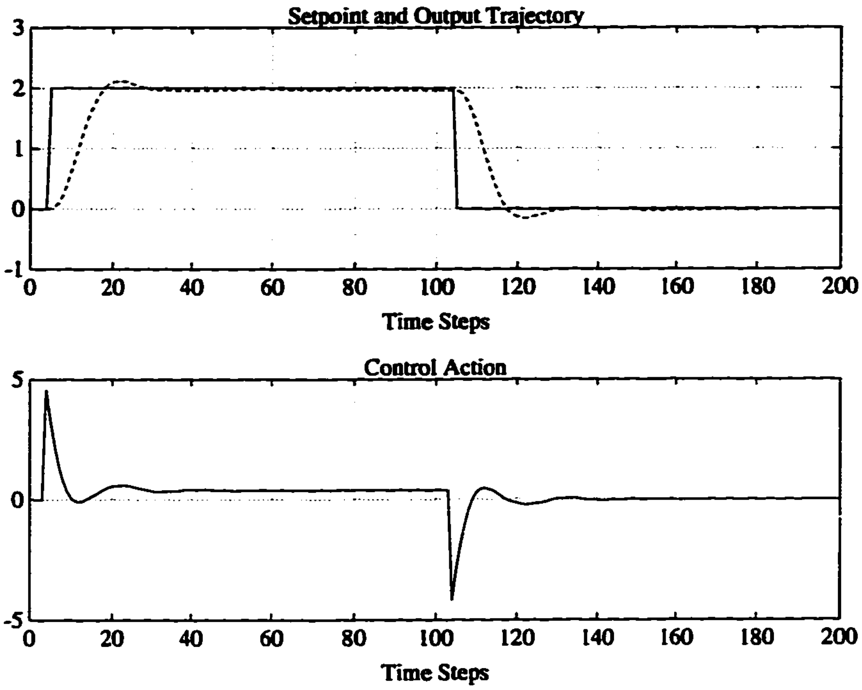

Figure 2.4:DMPC Control without Feedback

2.7.2 Case 2: Exponential Unstable Process

The open-loop process in the following example is exponentially unstable,

$$G_p(s) = \frac{1}{(s-0.2)(s^2+s+1)}$$

Since the unstable mode dominates the process output, the oscillatory modes have very little effect on the parameter estimation. Again choosing $n = 30$, the DARMA model is

$$A(q^{-1}) = 1 - 1.0408q^{-1}$$
$$B(q^{-1}) = 0.2019q^{-1}$$

Due to truncation error accumulation, the open loop controller tends to be divergent (Figure 2.5). Stable performance can be obtained as shown in Figure 2.6 by introducing a deadbeat feedback option (Qi & Fisher 1993).

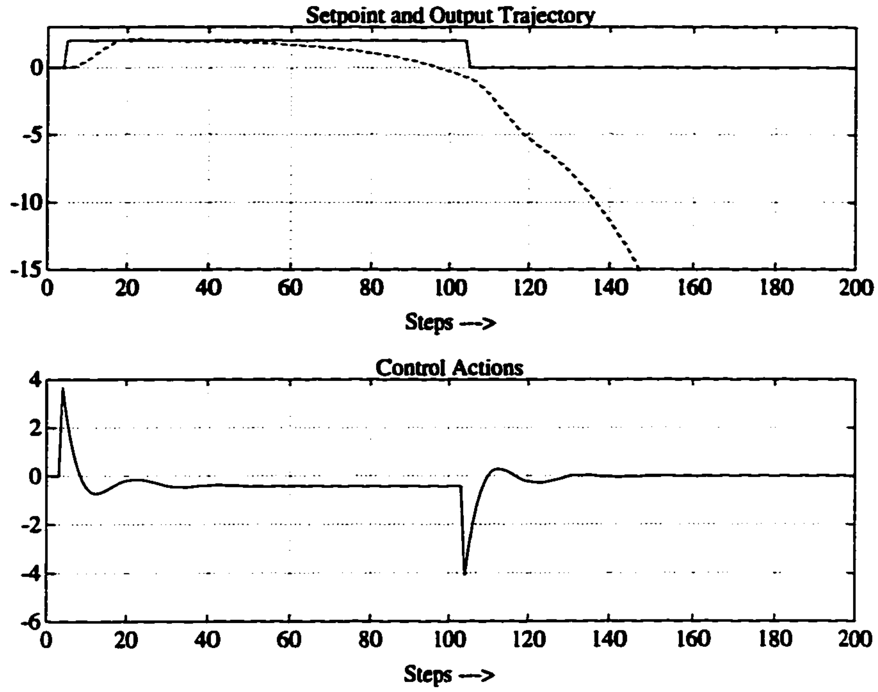

Figure 2.5:DMPC Control of An Unstable Process Without Feedback

2.8 Conclusion

A new dual-model state space formulation is developed to describe the process dynamics. It combines an explicit step response model and an implicit DARMA model. This formulation is a general process representation which keeps the advantages of using non-parametric step response coefficients to represent the fast dynamics and the compact parametric DARMA model to handle slow dynamics and/or unstable modes. The inclusion of a DARMA model significantly reduces the high dimension and computation required by non-parametric (step or impulse response) process representations. The special definition of the state variables facilitates state related analysis, state estimation, controllability analysis and leads to control relevant model identification. Simulations results show that the control performance using the reduced dimensional DMPC is equivalent to those of full order MPC of open loop stable processes.

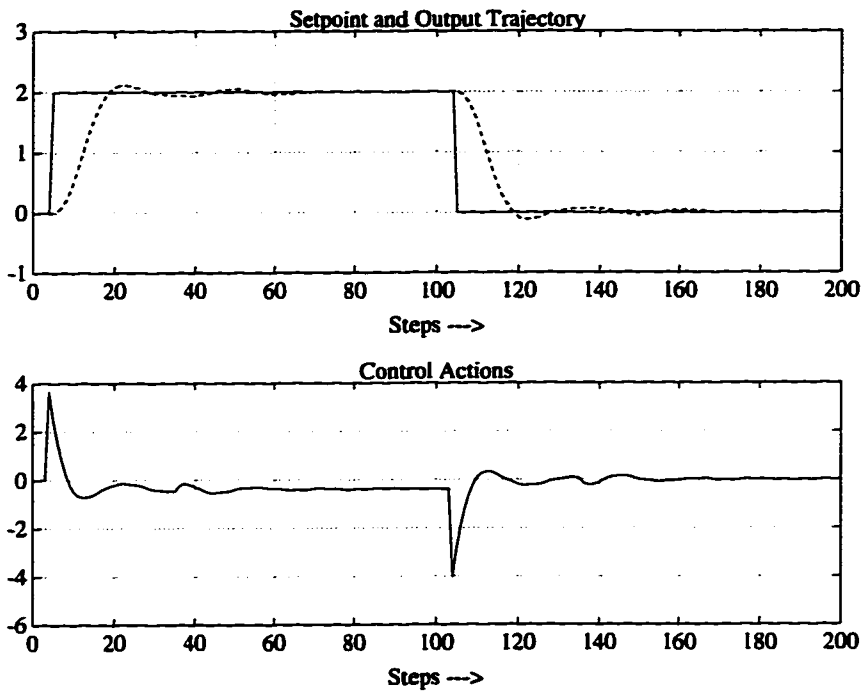

Figure 2.6:DMPC Closed Loop Control of An Unstable Process

Chapter 3

Control Relevant Dual-Model Identification

3.1 Introduction

Usually, the process model based on theoretical analysis is either impossible to derive rigorously or unsuitable for process control applications. In many practical situations, theoretical analysis or conceptual design can only give a rough idea about the process such as the independent/dependent variables. According to their cause-effect relationships, these variables are further defined as the controlled outputs, manipulated inputs and/or disturbance variables. Then, preliminary tests or pilot-plant tests are performed to estimate the structure of the process in terms of the steady state gain, over-damped or under-damped dynamic responses, etc. The detailed structure and parameters of the process model have to be obtained by model identification using dynamic input/output data.

Model identification itself is a very broad research field which has attracted extensive investigations for many years. How well the mathematical model can represent the process depends on many factors, from the types of excitation signals, the filters used in the data processing, to the algorithms used in estimating the model parameters. Obviously, different identification strategies can achieve different models even for the same process. Even though all models could give parameter convergent and therefore valid descriptions of the process, different models are used for different applications. For example, from the view point of control, a first order model is adequate for control of the slow dynamics of the process. On the other hand, in order to improve the specific control performance, the estimated model should be coordinated with the control objectives, *i.e.* control-relevant model identification.

Generally speaking, the least squares based algorithms are the most widely used parameter estimation methods, whether they are in BLS, RLS, AUDI and variations for the input/output model, or EKF for the linear/nonlinear state space models. Nevertheless, there are lots of adjustable parameters used in these algorithms which can be used to satisfy specific control requirements. For example, the objective function of BLS can be extended to match that of long range predictive control. For the input/output model (*e.g.* DARMA) used by Generalized Predictive Control, the conventional BLS which minimizes the difference between the actual output and estimated output is extended to Long Range Predictive Identification(LRPI) to cover the future differences between actual and predicted future output trajectories (Shook et al. 1991). In general, it is a nonlinear estimation algorithm requiring numerical searching. However, a simplified solution (which requires a prior information of the process) is also available which applies a LRPI filter to the input/output data.

For a model in state space form, instead of minimizing the output error, the general objective of model identification is to minimize the *state* variances. Several estimation algorithms have been developed and successfully applied in the past to achieve this objective. Extensive research in aerospace exploration and aircraft applications has provided solid theoretical support in the control area, such as LQC. Applying the Kalman Filter to estimating unknown parameters, the so-called Extended Kalman Filter approach has been developed to calculate the parameters recursively. Now, after many years of development both in the theory and practice, the EKF is recognized as a standard algorithm for state space model parameter estimation.

The state space model usually has two typical problems compared with input/output models. In matrix form, the state space model has more parameters than that the input/output transfer function format has. Since unknown parameters are treated as new variables, the dimension of the estimation problem is much larger. Fortunately, most matrices are sparse matrices with zeros or fixed values that significantly reduce the number of unknown parameters. Non-uniqueness is another problem of the state space model. For a given process, there exist several state space realizations, and the state variables in many state space representations do not have physical meanings. Therefore, most applications define the state variables and model structure first. Then, the state/parameter variables are estimated recursively to give a convergent result. Obviously, the choice of the state definition and the model structure is very important. Another approach does not pre-define the state variables and the model structure. Instead, it uses the numerical correlation inside the experimental data to determine the model. As a typical example, Canonical Variance Analysis(CVA) is a numerical algorithm which captures the principle (numerical) properties of the experimental data, with state variables defined as the principle components (Larimore 1990).

The objective of the estimation algorithm in the state space formulation is to minimize the state variance. Therefore, the definition of the state variables is very important. Poor choices often make direct measurement of the states impossible and generate impractical requirements for the EKF. For example, even if a state variable has no physical meaning, in order to reduce its variance, the EKF must make an extra effort during estimation. As a result of this unnecessary requirement, the main objective may be compromised. A minimal order state space based predictive control scheme (Ricker 1991, Balchen, Ljungquist & Strand 1992), designed to take advantage of the state space formulation and its solid theoretical supports, has the same problem in terms of parameter identification. As a matter of fact, using a minimal order description of the process leads eventually to a scheme equivalent to GPC.

A major requirement for predictive control is an accurate estimate of the future output predictions. Therefore, if the output predictions are defined as the state variables, a parameter identification algorithm which minimizes the variances of output predictions can best serve the predictive control's objective. The main purpose of this chapter is to combine the dual-model formulation with EKF to obtain estimates of the model parameters. Relevant issues such as parameter convergence and parameter uncertainties are also discussed in the following sections.

3.2 Dual-Model with Augmented State Variables

Define Θ as the unknown parameter vector. Then the dual-model state space formulation in Equation (2.16) with a disturbance/noise term can be rewritten as:

$$\begin{aligned} X(k+1) &= \Phi(\Theta)\,X(k) + \theta(\Theta)\,u(k) + \gamma(\Theta)\xi(k) \\ Y(k) &= H\,X(k) + v(k) \end{aligned} \tag{3.1}$$

Note that in the dual-model structure, the parameter vector Θ includes both the impulse response coefficients with respect to the absolute input $u(k)$ and the AR model parameters which define the unstable or slow dynamics .

$$\Theta = [h_1, h_2, \cdots, h_{n+1}, r_1, \cdots, r_{n_r}]^T$$

The unknown disturbance model coefficients can also be handled by Θ in the same way as other parameters. The resulting EKF estimation algorithm is essentially the same when disturbances are included although numerical difficulties may occur. For reasons of simplicity, they are not included in the future derivations of this thesis.

The parameter updating equation is

$$\Theta(k+1) = \Theta(k) + w(k) \tag{3.2}$$

Define an augmented state variable as:

$$Z(k) = \begin{bmatrix} X(k) \\ \Theta(k) \end{bmatrix}$$

Then, the corresponding state space formulation becomes:

$$\begin{aligned} Z(k+1) &= \mathcal{A}(\Theta)\,Z(k) + \mathcal{B}(\Theta)\,u(k) \\ Y(k) &= \mathcal{C}\,Z(k) + v(k) \end{aligned} \tag{3.3}$$

where the system matrices $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ are

$$\begin{aligned} \mathcal{A}(\Theta) &= \begin{bmatrix} \Phi(\Theta) & \mathcal{M}(\Theta, X, u) \\ 0 & I \end{bmatrix} \\ \mathcal{B}(\Theta) &= \begin{bmatrix} \theta \\ 0 \end{bmatrix} \\ \mathcal{C} &= [H,\ 0] \end{aligned} \tag{3.4}$$

and

$$\mathcal{M}(\Theta, X, u) = \frac{\partial}{\partial\Theta}[\Phi(\Theta)X(k) + \theta(\Theta)u(k)]|_{\Theta=\hat{\Theta}}$$

$$= \begin{bmatrix} u(k) & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & u(k) & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & u(k) & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & u(k) & x_{n+2-n_r}(k) & \cdots & x_n(k) & x_{n+1}(k) \end{bmatrix}_{(n+1)\times(n+n_r+1)}$$

Note that the elements in the last row of $\mathcal{M}$-matrix are the absolute values of the state estimates. If the original dual-model state space formulation in Equation (2.13) were used here, the elements of the last row would become the state differences Δx_i instead. Since the terms in the last row are the real driving force for parameter convergence, the closeness of consecutive state variables may lead to slow convergence or divergence of the parameters.

3.3 Predictive Criterion for Parameter Estimation

The EKF algorithm applied to the dual-model formulation Equation (3.3) has the following objective function:

$$\begin{aligned} \mathcal{J} &= \text{Var}[Z(k)] \\ &= \text{Var}[X(k)] + \text{Var}[\Theta(k)] + \text{others} \\ &= \mathcal{J}_1 + \mathcal{J}_2 + \text{others} \end{aligned}$$

where the interactions between the parameter and state estimates are included in the 'others' term.

The dual-model's state definition is:

$$\begin{aligned} X(k) &= Y_m(\cdot|k) \\ \hat{X}(k) &= \hat{Y}_m(\cdot|k) \end{aligned}$$

Therefore, the direct application of EKF to the dual-model(DMEKF) gives:

- the unknown parameters in Θ by minimizing the variance of the future output prediction;
- optimal estimates of the output predictions which can be used to calculate future control moves in the state feedback controller;

The first part of the DMEKF objective function, $\mathcal{J}_1$, can be further extended to a form similar to the LRPI form (Shook et al. 1991).

$$
\begin{aligned}
\mathcal{J}_1 &= \text{Var}[Y_m(\cdot|k)] \\
&= E\{[Y_m(\cdot|k) - \hat{Y}_m(\cdot|k)]^T [Y_m(\cdot|k) - \hat{Y}_m(\cdot|k)]\} \\
&= E\{[Y_p(k) - \hat{Y}_m(\cdot|k) - A\Delta U]^T [Y_p(k) - \hat{Y}_m(\cdot|k) - A\Delta U]\} \\
&= \text{Var}[Y_p(k) - \hat{Y}_m(\cdot|k)] + \text{Var}(A\Delta U) + \text{others} \\
&= \mathcal{J}_{11} + \mathcal{J}_{12} + \text{others}
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{J}_{11} &= \text{Var}[Y(k) - \hat{Y}_m(\cdot|k)] \\
&= E\{\sum_{j=1}^{n+1} [y(k+j) - \hat{y}_m(k+j|k)]^2\} \\
&\approx \frac{1}{N_n - n} \sum_{k=1}^{N_n - n} \sum_{j=1}^{n+1} [y(k+j) - \hat{y}_m(k+j|k)]^2 \\
&\approx \text{LRPI} \quad (\text{with } N_2 = n+1)
\end{aligned}
$$

REMARKS:

1. Both DMEKF and LRPI are predictive control relevant parameter estimation methods. DMEKF consists of a LRPI term plus extra terms related to parameter variances;

2. The order of the dual-model state space formulation, *i.e.* n, determines how many output predictions are considered by the parameter identification algorithm. To match the control performance, it is better to choose n equal to or greater than the prediction horizon P. Note that it is not a restriction on the choice of the dual-model order. If $n < P$, the estimation algorithm would consider less predictions but still give a better result than ordinary LS algorithms.

3. The extra terms in the DMEKF objective functions, *i.e.* $\mathcal{J}_2, \mathcal{J}_{12}$, restrict large changes in the parameter estimates and thereby promote parameter averaging or smoothing.

4. DMEKF can simultaneously estimate the states and parameters for the purpose of adaptive control (even though the adaptive control is not covered in this thesis);

5. Because of the integrated framework, a trade-off can be easily made between state estimation and parameter estimation. For the purpose of state estimation, the parameter part can be turned off (like state observer). On the other hand, better parameter convergence can be achieved by putting more weightings on this term.

3.4 Parameter/State Observability Analysis

In state space formulations, a fundamental observability analysis should be carried out before any attempt is made to estimate the unmeasured state variables and/or unknown parameters. A standard observability analysis can tell whether it is possible to obtain a convergent estimate of unknowns from a finite number of measurements of the process inputs and outputs.

3.4.1 Observability with Augmented State Variables

The original dual-model state space structure is changed by extending the state variable to cover unknown parameters which directly affects the state observability. As discussed in Chapter 2, with known model coefficients, the state variables in the dual-model formulation are *equally* estimated(observed) from the output measurements. The state feedback controller can then treat them *equally* for future control calculations.

When the unknown parameters are also included in the state variables, the observability matrix of the augmented equation (3.3) is:

$$
\begin{aligned}
\mathcal{O} &= \begin{bmatrix} \mathcal{C} \\ \mathcal{C}\mathcal{A} \\ \vdots \\ \mathcal{C}\mathcal{A}^{n} \\ \mathcal{C}\mathcal{A}^{n+1} \\ \vdots \\ \mathcal{C}\mathcal{A}^{2n+2+n_r} \end{bmatrix} \\
&= \left[\begin{array}{c|c} H & H\Phi^{n+1} \\ H\Phi & H\Phi^{n+2} \\ \vdots & \vdots \\ H\Phi^{n+1} & H\Phi^{2n+2+n_r} \\ \hline 0 & H\sum_{i=0}^{n}\mathcal{A}^i\mathcal{M} \\ H\mathcal{M} & H\sum_{i=0}^{n+1}\mathcal{A}^i\mathcal{M} \\ \vdots & \vdots \\ H\sum_{i=0}^{n-1}\mathcal{A}^i\mathcal{M} & H\sum_{i=0}^{2n+n_r}\mathcal{A}^i\mathcal{M} \end{array}\right] \\
&= \begin{bmatrix} \mathcal{O}_{11} & \mathcal{O}_{12} \\ \mathcal{O}_{21} & \mathcal{O}_{22} \end{bmatrix}
\end{aligned}
$$

For simplicity, the sub-matrices in $\mathcal{O}$ can be written here, with $n_r = 1$, as

$$\mathcal{O}_{11} = I_{n+1}$$

$$\mathcal{O}_{12} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 & 0 \\ u(k) & 0 & \cdots & 0 & 0 & 0 \\ u(k) & u(k) & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ u(k) & u(k) & \cdots & u(k) & 0 & 0 \\ u(k) & u(k) & \cdots & u(k) & 0 & 0 \end{bmatrix}_{(n+1)\times(n+2)}$$

$$\mathcal{O}_{21} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 & r \\ 0 & 0 & \cdots & 0 & 0 & r^2 \\ 0 & 0 & \cdots & 0 & 0 & r^3 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & r^{n+1} \\ 0 & 0 & \cdots & 0 & 0 & r^{n+1} \end{bmatrix}_{(n+2)\times(n+1)}$$

$$\mathcal{O}_{22} = \begin{bmatrix} u(k) & u(k) & \cdots & u(k) & u(k) & x_{n+1}(k) \\ u(k) & u(k) & \cdots & u(k) & u(k)(1+r) & x_{n+1}(k)(1+r) \\ u(k) & u(k) & \cdots & u(k) & u(k)(1+r+r^2) & x_{n+1}(k)(1+r+r^2) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ u(k) & u(k) & \cdots & u(k) & u(k)\sum_{j=0}^{n} r^j & x_{n+1}(k)\sum_{j=0}^{n} r^j \\ u(k) & u(k) & \cdots & u(k) & u(k)\sum_{j=0}^{n+1} r^j & x_{n+1}(k)\sum_{j=0}^{n+1} r^j \end{bmatrix}_{(n+2)\times(n+2)}$$

The observability matrix for the augmented dual-model formulation has two obvious features:

1. The observability matrix includes only one input $u(k)$, the states, and the parameters. Usually, the latter can be replaced by their estimates.

2. For the simple case of $n_r = 1$, only the estimates of the state variable $x_{n+1}(k)$ and the last parameter estimation r appear in the matrix, and hence play particularly important roles for the properties of the observability.

Since the columns from $(n+2)$ to $(2n+1)$ of the observability matrix $\mathcal{O}$ can be deleted by simple transformation, it is not difficult to observe that, no matter what values are used in $u(k)$, $x_{n+1}(k)$ and r, the maximum column rank of $\mathcal{O}$ is $n+2$. This implies that, theoretically, some parameters and/or their combinations defined in the dual-model state space formulation are not observable. This property definitely affects the convergence of parameter/state estimations.

3.4.2 Parameter Estimation Convergence

Observability analysis is actually a batch form of the state/parameter estimation problem. A set of linear algebraic equations can be made up of $(n+n_r+1)$ unknown variables and $(n+n_r+1)$ output measurements. In order to make the batch estimation algorithm convergent, generally, a sufficient condition is that the observability matrix be full rank. With this condition, the unknown states and parameters can be uniquely estimated by inverting the observability matrix.

Even though the Kalman filter algorithm appears in a recursive form, questions about the convergence of the algorithm are also answered by examining its corresponding batch form, *i.e.* the observability. Theoretically, the estimation procedure is convergent within finite steps only if the formulation is fully observable. In those non-observable situations, the original formulation should be adjusted to improve the properties of the observability matrix. For example, previous measurements of the output variable can be used (Gudi, Shah & Gray 1994). A general method is to use an initial covariance matrix in the recursive estimation algorithm (detailed in the next section). An explanation in terms of the batch format observability analysis is that this constant matrix is added to improve the condition of the observability matrix(which is equivalent to the control weighting constant λ commonly used in predictive control calculation). The result, as expected, is that more than $(n + n_r + 1)$ output measurements are required to estimate the $(n + n_r + 1)$ unknowns.

3.5 Parameter/State Estimation by Extended Kalman Filter

After more than 30 years of development, the EKF technique is quite mature. The special structure of the dual-model further simplifies the formulation.

3.5.1 Formulation

The general state space formulation

$$\begin{aligned} X(k+1) &= \Phi(\Theta)\,X(k) + \theta(\Theta)\,u(k) + \gamma(\Theta)\xi(k) \\ Y(k) &= H\,X(k) + v(k) \end{aligned}$$

requires prior knowledge of covariances of the noise terms :

$$\begin{aligned} E(\xi\xi^T) &= Q^\xi \\ E(vv^T) &= Q^v \\ E(\xi v^T) &= Q^c \end{aligned}$$

With measured process input/output data:

$$u_0, y_0, u_1, y_1, \ldots$$

a two step EKF algorithm can be obtained as (Ljung 1979):

$$\hat{X}(k+1) = \Phi_k \hat{X}(k) + \theta_k u(k) + K_k[Y(k) - H\hat{X}(k)] \tag{3.5}$$

$$\hat{\Theta}(k+1) = \hat{\Theta}(k) + L_k[Y(k) - H\hat{\Theta}(k)] \tag{3.6}$$

$$\hat{X}(k+1) = 0 \tag{3.7}$$

$$\hat{\Theta}(k+1) = \Theta_0 \tag{3.8}$$

The gain vectors for states and parameters, K_k and L_k, are calculated recursively by:

$$
\begin{aligned}
K_k &= [\Phi_k P_{1,k} H^T + M_k P_{2,k}^T H^T + Q^c] S_k^{-1} \\
L_k &= [P_{2,k}^T H^T] S_k^{-1} \\
S_k &= H P_{1,k} H^T + Q^v \\
P_{1,k+1} &= \Phi_k P_{1,k} \Phi_k^T + \Phi_k P_{2,k} M_k^T + M_k P_{2,k}^T \Phi_k^T + M_k P_{3,k} M_k^T - K_k S_k K_k^T + \gamma Q^\xi \gamma^T \\
P_{2,k+1} &= \Phi_k P_{2,k} + M_k P_{3,k} - K_k S_k L_k^T \\
P_{3,k+1} &= \Phi_k P_{3,k} - L_k S_k L_k^T \\
P_{1,0} &= \Pi_0 \\
P_{2,0} &= 0 \\
P_{3,0} &= \Sigma_0
\end{aligned}
$$

Π_0 and Σ_0 are the variances of the initial estimates of states and parameters respectively. Together, they make up the covariance matrix of the augmented state $Z(0)$, *i.e.* the initial guess,

$$P_0 = E(Z(0)Z(0)^T) = \begin{bmatrix} P_{1,0} & P_{2,0} \\ 0 & P_{3,0} \end{bmatrix} = \begin{bmatrix} \Pi_0 & 0 \\ 0 & \Sigma_0 \end{bmatrix}$$

where $P_{3,0}$ corresponds to the initial parameter covariance matrix used in the input/output DARMA recursive least squares algorithms (Shah & Cluett 1991). In practice, these initial covariance matrices become tuning parameters in the model identification algorithm which actually specify the weighting matrix in the EKF objective function. The commonly suggested choices are:

$$
\begin{aligned}
X(0) = 0 \quad &, \quad \Pi_0 = Var(y) \\
\Theta(0) = \Theta_0 \quad &, \quad \Sigma_0 = 100 \times Var(y)
\end{aligned}
$$

Special Considerations:

1. In the Recursive Least Squares(RLS) algorithm, the diagonal elements of the parameter covariance matrix should be large enough to achieve both parameter convergence and algorithm alertness (Shah & Cluett 1991). Therefore, the corresponding matrix $P_{3,0}$ should have large absolute values for diagonal elements and large relative values compared to those in the matrix $P_{1,0}$;

2. The unknown parameters for the AR model in the dual-model description usually determine the dominant pole locations of the process. Special emphases should be given to their estimation, *e.g.* large weightings should be put in $P_{3,0}$. For example, a typical covariance matrix P_0 for parameter estimation is

$$P_0 = \begin{bmatrix} I_{(n+1)} & 0 & 0 \\ 0 & 100 I_{(n+1)} & 0 \\ 0 & 0 & 500 I_{n_r} \end{bmatrix} Var(y)$$

An important feature of the DMEKF is that it explicitly illustrates the relationship between state and parameter estimation in the same covariance matrix. The so called "relative importance" of the parameters/states is reflected by the properties of the covariance matrix P_k. At the initial stage, a trade-off can be easily obtained by manipulating the diagonal elements in the P_0 matrix. For example, by applying an EKF to an 8th order dual-model formulation, Figure 3.1 shows the estimation trajectories (the solid line for the actual value and dotted line for the estimated value) of the last parameter(9th) and the last state variable(9th), for an initial weighting matrix

$$P_0 = \begin{bmatrix} I_9 & 0 & 0 \\ 0 & 100I_9 & 0 \\ 0 & 0 & 100 \end{bmatrix} Var(y)$$

The initial conditions for parameters and state variables are

$$X(0) = y(0) \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}; \; \Theta(0) = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}; \; r(0) = 0.6$$

Note that the parameter estimates converge within 100 steps and the state estimate converges much fast (< 40 steps). Even though more weighting is applied on the parameters (100 times more), the parameter estimation still converges more slowly than the state variable. This is very reasonable since the parameter updating equation (3.2) is artificially introduced to reduce the differences in the state variables.

With a different weighting on the parameter estimation part in P_0,

$$P_0 = \begin{bmatrix} I_9 & 0 & 0 \\ 0 & 500I_9 & 0 \\ 0 & 0 & 500 \end{bmatrix} Var(y)$$

the same parameter/state estimation set is shown in Figure 3.2. Note that now the parameter estimate converges faster (< 50 steps) at the expense of larger variations in the state estimates.

3.5.2 Parameter/State Uncertainty Estimation

By definition, the covariance matrix P_k includes all the information about the quality of the estimates at the kth-instant,

$$P_k = E(Z(k)Z(k)^T) = \begin{bmatrix} P_{1,k} & P_{2,k} \\ 0 & P_{3,k} \end{bmatrix}$$

Just as in the RLS algorithm for estimation of the DARMA model parameters, $P(k)$ has to be large for algorithm alertness and small for good estimation accuracy. Therefore, usually, the initial values used for P_0 elements are relative large so that the estimation algorithm can sense the estimation errors and correct the initial estimates of

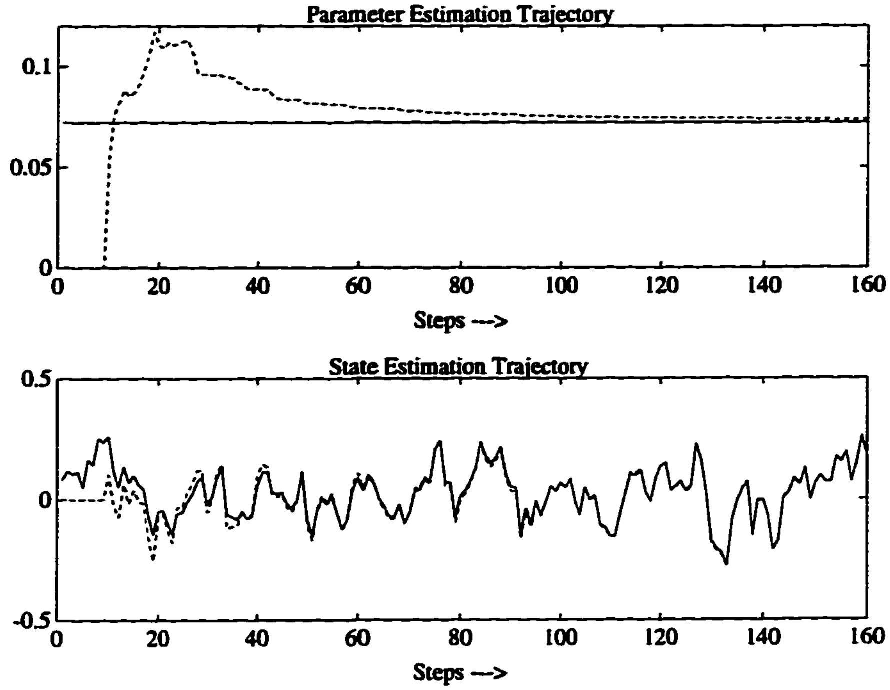

Figure 3.1:The Effect of P_0 on State and Parameter Estimates

the state and parameters. As the recursive process goes, the covariance is gradually reduced and a better estimate is obtained.

From the estimated covariance matrix, a 95% confidence interval can be obtained for the estimation of parameters and states. This provides information about the model uncertainty bounds. Note that even though the EKF simultaneously gives estimates and uncertainties of both the states and parameters, only the parameters and their uncertainties affect the closed loop stability (detailed in Chapter 6). Therefore, for robustness analysis, parameter estimation needs more attention than state estimation. It is recommended that a separate feedback observer be used for state estimation (Chapter 5). The state uncertainty does affect the optimality of the predictive control performance. The state estimation uncertainty may also be helpful to invalidate points in the output trajectory if their estimates are bad, *e.g.* to put lower weight on the invalidated output estimates when completing the control calculation.

3.5.3 Determination of the Optimal Order of the Dual-Model

As shown in Chapter 1, the dual-model formulation is a special state space realization with the order greater than the minimal order realization form but less than the order of the full FIR model(*e.g.*DMC). There is a wide range for the order specification of the dual-model, with some restrictions. It would be desirable to generate an optimal model order of the dual-model from the input/output measurements.

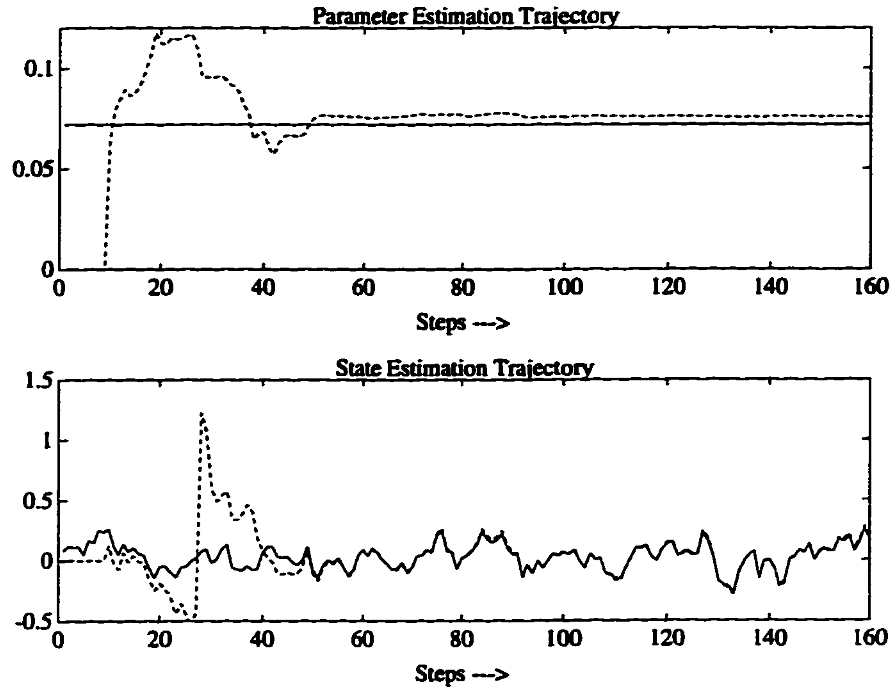

Figure 3.2:The Effect of P_0 on State and Parameter Estimates

Several methods have been used in the past to find the optimal model order including the general AIC criterion. Simple methods have been obtained by using the residuals of the actual output measurements and model outputs for determining the best order of DARMA models (Niu & Fisher 1994). Another upcoming method is based on CVA analysis for a state space model (Larimore 1990) which determines the model order based on the numerical significance of the state variables. Even though it gives a minimum order state space formulation, the CVA model usually does not have a physical meaning for its state variables. Therefore, the residual method, which minimizes the difference between the actual output measurements and the model outputs, is used here.

A nonlinear optimization can be formulated to find the optimal model order n,

$$
\begin{aligned}
\min_{n} \quad & \mathcal{J} = \sum_{k=1}^{N} [Y(k) - \hat{Y}(k)]^2 \\
s.t. \quad & n > 0 \\
& \hat{X}(k+1) = \Phi_k \hat{X}(k) + \theta_k u(k) + K_k(Y(k) - H\hat{X}(k)) \\
& \hat{Y}(k) = H\hat{X}(k)
\end{aligned}
$$

This procedure is straightforward since the order is simply an integer in a finite range.

3.6 Simulation Results

The process used in the previous sections is used again as an example of the DMEKF algorithm. From the observability analysis, an appropriate initial covariance matrix P_0 has to be used to obtain a convergent parameter set. The resulting model coefficients are compared with those by LRPI and BLS. Parameter uncertainty is also obtained for the later illustration of robustness analysis.

The true process is a third order system $1/(s+1)(3s+1)(5s+1)$. A PRBS signal sequence is used to excite the process for model identification with the sampling interval $T_s = 1$. A first order model is assumed for the slow AR model, *i.e.* one unknown model parameter. Therefore, the total number of unknown parameters is $n+2$. As the model order increases, the residual decreases to minimum point around $n = 8$ (Figure 3.3). After the parameter estimation converges, the estimated step responses of the process can be calculated as shown in Figure 3.4 and Figure 3.5. Obviously, without the addition of measurement noise, the estimated step response moves closer to the true one (the solid curve) as the model order increases. With noise, the best model order is $n = 7$ since increasing model order drives the model to fit the random noise. As shown in Figure 3.5, the estimated step responses with the model order $n = 8$ and $n = 11$ move away from the true one.

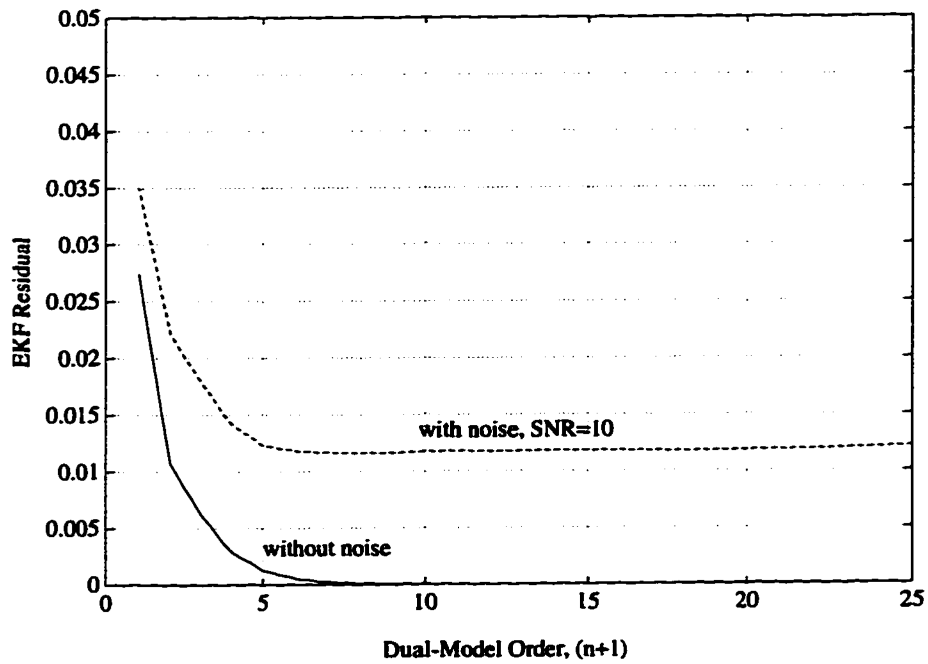

Figure 3.3:The Effect of Model Dimensions on the Residual

After fitting a DARMA model to the measured data, the step responses generated using the parameters estimated by different parameter estimation algorithms, *i.e.*BLS, LRPI and DMEKF, were calculated and plotted in Figure 3.6. Note that for a fair comparison, a low order dual-model, $n = 1$, is used. Therefore, DMEKF has 3

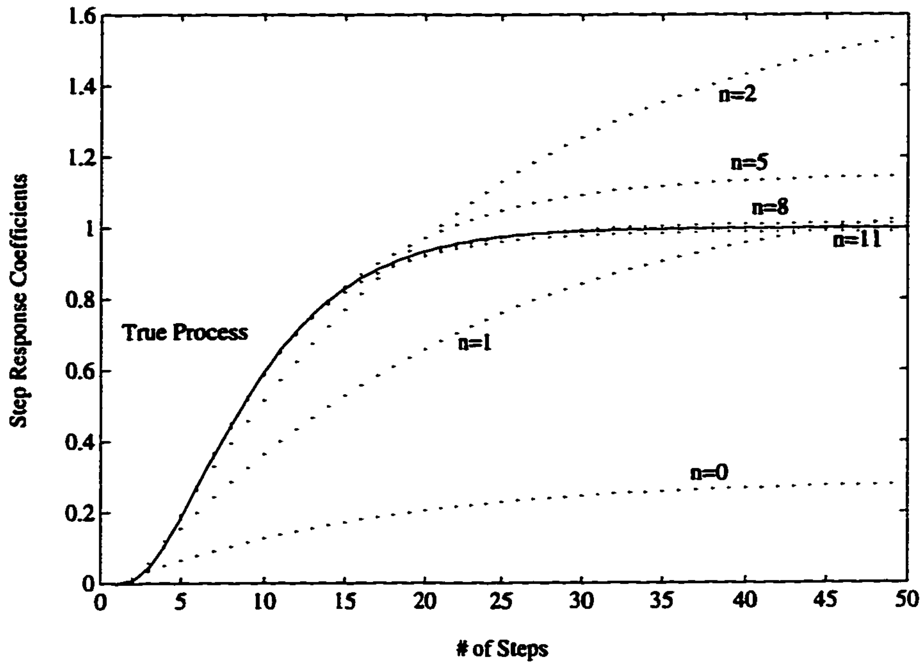

Figure 3.4:The Step Response Coefficients Using DMEKF, No Noise

($= n + 2$) unknown parameters and 2 ($= n + 1$) output predictions in its estimation objective function. The corresponding LRPI uses first order polynomials for both $A(q^{-1})$ and $B(q^{-1})$ in the DARMA model (*i.e.* 3 unknown parameters), and $N_2 = 2$ as the prediction horizon. The simple BLS has both $A(q^{-1})$ and $B(q^{-1})$ as first order polynomials as well. Even with a model order less than optimal, DMEKF gives much better estimation, due to its control relevant formulation.

3.7 Conclusion

The extended Kalman filter algorithm applied to the dual-model representation leads to a model well suited for predictive control. Detailed observability analyses have been used to develop methods of achieving fast parameter convergence and to make trade-off between the state estimation and parameter estimation. The same algorithm can also be used to estimate state variables for feedback control and the parameter uncertainties for robustness analysis. Analysis and simulation show superior results relative to other model structures and estimation algorithms. Finally, another advantage of using the EKF algorithm in the dual-model formulation is that it can also be extended to multivariable processes without any changes.

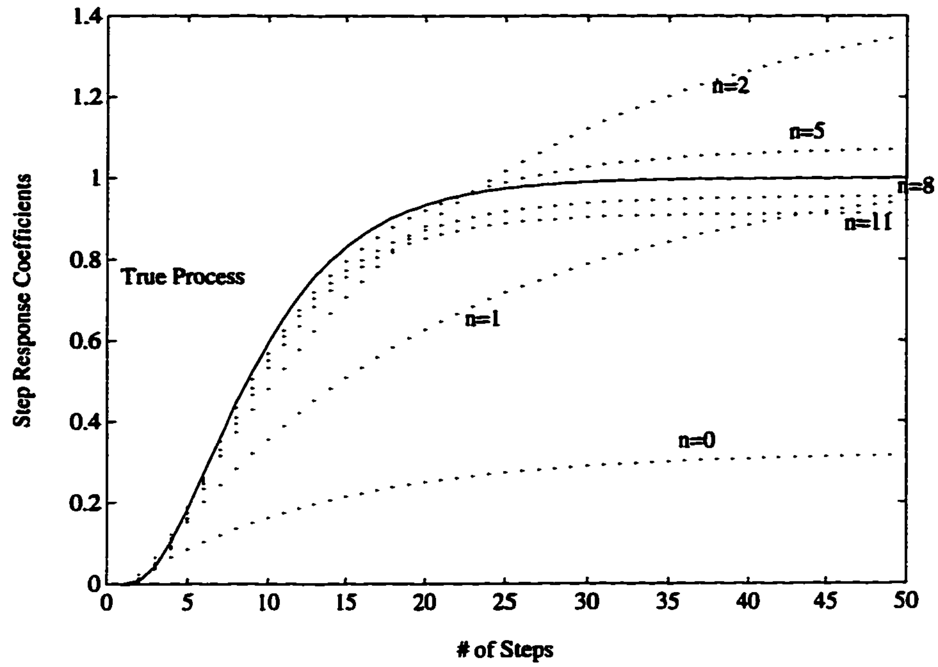

Figure 3.5:The Step Response Coefficients Using DMEKF, With Noise

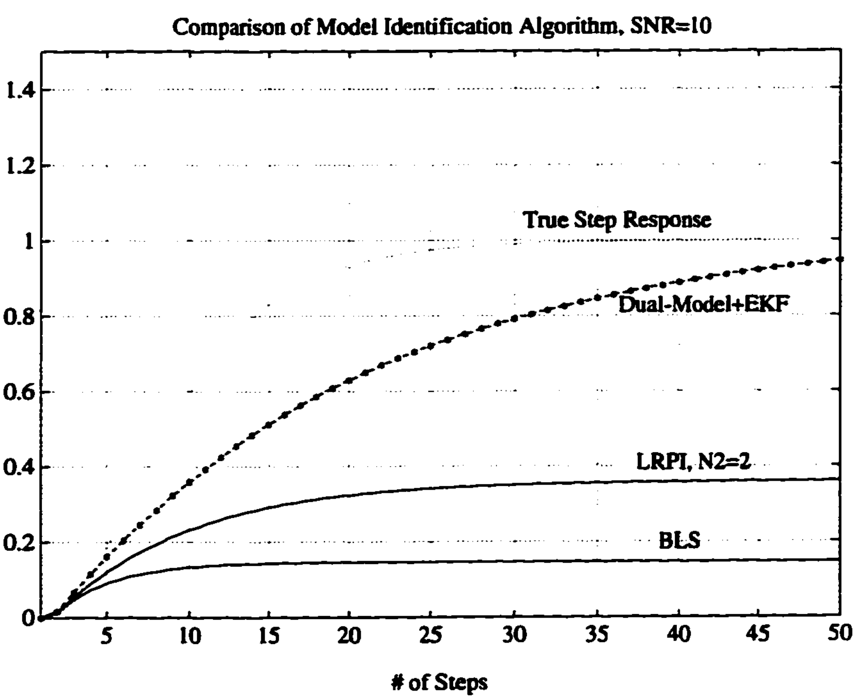

Figure 3.6:A Comparison of DMEKF and LRPI algorithms

Chapter 4

Nominal Controller Design Based on The Dynamic Matrix

4.1 Introduction

Generally speaking, predictive control is an optimization problem with or without constraints. It calculates the future control sequences, based on open loop process model predictions, that minimize the differences between predicted future outputs and the desired output trajectory. In the state space domain, the unconstrained predictive controller can be considered as a special type of state feedback controller with a fixed controller gain. Therefore, its closed loop performance, *i.e.* pole locations, stability as well as robustness can be evaluated (Lee et al. 1993, Qi & Fisher 1994). The controller gain is a function of the MPC tuning parameters, *e.g.* integer numbers such as prediction horizon, control horizon, and continuous numbers such as input/output weightings, etc.

In contrast to other control tuning parameters such as PID parameters or pole locations in several pole assignment controller design methods, the MPC tuning parameters usually have very intuitive explanations. For example, the output prediction horizon is how far the controller looks into the future, and the control horizon is how many future control moves are used to correct the differences between the setpoint and output trajectories. The control weighting is simply a penalty on large input moves. As a result of being easily understood, they are widely accepted for practical applications. On the other hand, mathematically, they are also incorporated into the optimization formulation. Therefore, in addition to the process modelling, another major part of controller design is the choice of these MPC parameters. This is a rather more difficult task than understanding the predictive control theory.

Since the first commercial MPC control software - DMC emerged in the early 80's, there have been many variations and similar products marketed with specific features for different control applications. Many MPC tuning guidelines, have been put forward in the past few years based on either academic developments or industrial experience (Ricker 1991). However, the fundamental concept and the general structure of the predictive controller are still the same and include the same *dynamic* *matrix*. The dynamic matrix representing the process dynamics is used to calculate future control moves and therefore its matrix properties are important factors for the control performance. In spite of this, many MPC tuning rules were developed intuitively from either physical explanations or extensions of the optimization objective function without evaluating their effects on the properties of the dynamic matrix.

Another problem with predictive controller design is that although these control parameters do have explicit physical explanations, they do not have a unique independent impact on the control performance. This increases the difficulty of tuning the multivariable predictive controller.

The MPC design problem is therefore re-investigated from the view point of controllability analysis. In order to make the controlled variables, i.e. process outputs, follow the future desired trajectory, future control profiles need to be calculated. Obviously, model inverse designs such as Internal Model Control (Morari et al. 1989) are the theoretical solution. However, in reality, various restrictions apply due to either constraints on the process variables or inaccurate modelling. Similar problems exist for the design of the predictive controller. Perfect tracking of future outputs requires an exact inverse of the dynamic matrix. However, in most processes, the full dimensional dynamic matrix is poorly conditioned which either does not allow a matrix inverse or results in severe problems due to model uncertainties. Several methods have been used over the years to make the trade-off between performance and robustness. In the following sections, the effect of several intuitive control parameters used to manipulate the matrix structure is explained in terms of changing the condition number of the dynamic matrix. Therefore, a systematic design scheme for predictive controller can be obtained based on formal analysis of the dynamic matrix.

4.2 Output Prediction Equation

The future output trajectory including the effects of future control moves can be represented as:

$$Y_p(k) = \Phi_p X(k) + A\Delta U(k) \tag{4.1}$$

where the M-step future control vector, ΔU, and the $P \times M$ dynamic matrix A, are given by

$$\Delta U(k) = [\Delta u(k),\ \Delta u(k+1),\ \cdots,\ \Delta u(k+M-1)]^T$$

$$A = \begin{bmatrix} S_1 & 0 & 0 & \cdots & 0 \\ S_2 & S_1 & 0 & \cdots & 0 \\ S_3 & S_2 & S_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ S_M & S_{M-1} & S_{M-2} & \cdots & S_1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ S_P & S_{P-1} & S_{P-2} & \cdots & S_{P-M+1} \end{bmatrix}_{P\times M}$$

The constant matrix Φ_p is used to balance the dimensional difference between the state vector $X(k)$ and the prediction output vector $Y_p(k)$,

$$\Phi_p = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & \cdots & 0 \end{bmatrix}_{p\times(n+1)}$$

Then, for every setpoint trajectory, an unconstrained MPC solution can be obtained by calculating the control trajectory, ΔU, that minimizes an appropriate performance index such as:

$$\mathcal{J} = \sum_{j=1}^{P} [y_{sp}(k+j) - y_p(k+j)]^2 \tag{4.2}$$

Using the output prediction Equation (4.1) defined above leads to the following control law:

$$\Delta U(k) = A^*[Y_{sp}(k) - \Phi_p X(k)]$$

where the pseudo-inverse of the dynamic matrix A is

$$A^* = (A^T A)^{-1} A^T$$

REMARKS:

In order to make the future outputs $Y_p(k)$ achieve the desired trajectory perfectly, two conditions must be satisfied:

1. The number of future control moves should be equal to or greater than the output predictions, *i.e.* $M \geq P$;
2. The dynamic matrix, A, should be full rank, *i.e.* a square matrix A_a with $M = P$ for the dynamic matrix A.

However, for most processes, although the dynamic matrices A_a are full rank mathematically, they are ill-conditioned numerically. Direct application of Equation (4.1) usually results in aggressive control with a very poor robustness since a small change in the process model, *i.e.* the elements of A_a, produces a large error in the calculated control moves. Therefore, many MPC tuning parameters have been introduced to improve the numerical condition of the matrix.

4.3 The State Feedback Form of Predictive Control

In practice, the control profile $\Delta U(k)$ obtained from optimization is not used to drive the actuator over the full control trajectory. Instead, only the first element, $\Delta u(k)$, is implemented at time k and the control law is executed at every control interval in accordance with the widely used *receding horizon* principle. This enables the controller to pick up new disturbances coming into the process and make corrections. Therefore,

$$\begin{aligned}\Delta u(k) &= C^T \Delta U(k) \\ &= C^T A^* [Y_{sp}(k) - \Phi_p X(k)] \qquad (4.3) \\ &= -K_{mpc} X(k) + R Y_{sp} \qquad (4.4)\end{aligned}$$

where $C^T = [1,\ 0,\ \cdots,\ 0]$. Note that this is a state feedback controller and, in general, the feedback gain is time invariant but is a function of the MPC tuning parameters as shown by (4.4), *i.e.*

$$\begin{aligned}K_{mpc} &= \mathcal{F}_1(\lambda, M, P) = C^T A^* \Phi_p \qquad (4.5) \\ R &= \mathcal{F}_2(\lambda, M, P) = C^T A^*\end{aligned}$$

Using the dual model formulation Equation (2.13) for the open loop process, the closed loop formulation can be described as

$$\begin{aligned}X(k+1) &= (\Phi - \theta K_{mpc}) X(k) + \theta R Y_{sp}(k) \qquad (4.6) \\ Y(k) &= H X(k)\end{aligned}$$

REMARKS:

1. For an asymptotically closed loop stable system, all eigenvalues of the discrete control matrix (*i.e.* $\Phi - \theta K_{mpc}$) must be within the unit circle.

   Note that the eigenvalue condition for state space system stability is a much stronger condition than the pole location requirements in other input/output descriptions. As a matter of fact, since some states and/or their combinations are not completely controllable, (unstable) zero and pole cancellation can happen when converting the state space formulation into an input/output description. Therefore, this sufficient and necessary *state* stable condition of the state space system becomes only a sufficient condition for the input/output system. A direct effect of using this stronger stability condition can be found in the robustness analysis (in Chapter 6) where sufficient conditions result in conservative robust criteria.

2. If the process model is accurate and all the parameters are known then the design procedure is straightforward and the performance and stability of the closed loop system can be evaluated directly. However, this conventional approach can not be used if MPM is significant. For example, deadbeat control design normally places all the poles at the origin. However, the pole locations are strongly affected by MPM so that the actual system stability margins and performance are unknown. The robustness issue will be analyzed in Chapter 6.

3. So far, the state variables have been assumed to be available for use in state feedback control. In fact, only the process output is directly measured and hence the state variables have to be estimated by a state observer which will be discussed in Chapter 5 in detail.

4.4 Dynamic Matrix Conditioning

Two effective methods, matrix decomposition and matrix weighting, can be used to improve the numerical condition of the dynamic matrix at the cost of slowing the control response. Even though it means the controlled output values from the prediction equation (4.1) will not exactly match the desired trajectory (the control calculation becomes a least squares solution), the benefits are significant enough to overcome the negatives. Actually, the controlled output predictions will reach the desired trajectory after additional control steps ($> P$).

The condition of a matrix can be evaluated by a scalar, the condition number. Mathematically, the condition number can be calculated by the ratio of the largest eigenvalue to the smallest eigenvalue and therefore is a relative measure of the matrix condition. For a deficient matrix D, $Cond(D) = \infty$.

4.4.1 Matrix Decomposition

A subset matrix with better conditioning can be obtained by matrix decomposition. Then the inversion of the sub-matrix does not result in the same numerical problems.

1. **Principle:**

   Decompose the full sized, dynamic matrix A_a into two parts:

$$A_a = T_a \cdot P_a^T + E$$

   where

$$\begin{aligned} A_a &\in \Re^{P \times P} \\ T_a &\in \Re^{P \times a} \\ P_a^T &\in \Re^{a \times P} \end{aligned}$$

   Then the least squares solution using the sub-matrix T_a can be obtained as

$$\Delta U_a(k) = T_a^*(Y_{sp}(k) - \Phi_p X(k)) \tag{4.7}$$

   and the original control moves $\Delta U(k)$ can be retrieved, if necessary, from $\Delta U_a(k)$ using the relation

$$\Delta U_a(k) = P_a^T \Delta U(k) \tag{4.8}$$

   The objective of matrix decomposition is to find a well-conditioned T_a and matrix P_a which best represents the original matrix A_a. The residual matrix E should be as small as possible. Matrix decomposition also involves finding the best dimension a for the matrices T_a and P_a.

Principal Component Analysis(PCA) is one approach that can be applied to decompose the dynamic matrix into two parts (Wilkinson et al. 1994). The T_a matrix consists of principle eigenvalues and the P_a matrix consists of the corresponding eigenvectors. Since only significant eigenvalues are included in T_a, good conditioning of this matrix can be obtained and hence the numerical properties of the $\Delta U_a(k)$ calculation in Equation (4.7) are improved. However, since ΔU_a does not contain future control moves to be implemented, the true control moves $\Delta U(k)$ have to be retrieved using Equation (4.8). Remember that since P_a^T is a $(a \times P)$ matrix with more columns than rows, there are more unknown variables than equations. The solution for $\Delta U(k)$ is not trivial. Mathematically, there is an infinite number of solutions $\Delta U(k)$ for a specific $\Delta U_a(k)$. Additional information has to be used to select the best one to implement as the control move.

A different approach, the finite horizon method, is used to decompose the dynamic matrix since it is much simpler and efficient.

2. **Finite Control Horizon Method:**

The finite control horizon method uses

(a) A pre-specified matrix A to replace T_a. Note that the matrix A is the dynamic matrix commonly used in MPC with $M < P$. The first M columns of A are identical to those in A_a. Mathematically, the dynamic matrix A_a is decomposed as

$$A_a = A \cdot P_a^T + E$$

(b) The calculated ΔU_a (instead of $\Delta U(k)$) in Equation (4.8) is implemented directly.

This arrangement has several advantages including a better matrix condition of A, a meaningful description of the process and a unique least squares solution for the control moves.

The finite control horizon method has been utilized in predictive control for many years, without theoretical proof or analysis. It is well known that the control horizon M is a very critical tuning parameter for the total control performance. But the choice of the control horizon has been intuitive in nature and lacked theoretical support. For SISO processes, the choice is quite intuitive and straightforward. But, for MIMO processes, there has been no general rule to follow and intuitive choices have lead to problems.

Based on matrix decomposition theory, this problem can be investigated to determine the optimal value of M. Obviously, the choice of M should result in

- Better conditioning of the dynamic matrix A;
- Minimum residuals in E.

Note that for MIMO processes, the control horizon M is the summation of all control horizons of all input channels and P is the sum of all output prediction horizons.

3. Decomposition Procedure

Mathematically, the general objective is

$$\begin{array}{ll} \min_M & \|A_a - AP_a^T\| \\ s.t. & 1 \le M \le P \\ & Cond(A) \le specified\ \ value \quad e.g.\ 100 \end{array}$$

This can be solved by integer optimization programs. However, since M is simply a bounded integer, a much simpler recursive numerical procedure for this purpose can be summarized as:

- **STEP 1:** Build the full sized $P \times P$ dynamic matrix A_a using the unit step response coefficients of the process, $A_a \in \Re^{P \times P}$;
- **STEP 2:** Start from $M = P - 1$;
- **STEP 3:** Decompose $A_a = AP_a^T$, which leads to a least squares solution:

$$P_a^T = A^* A_a$$

  where A^* is the pseudo-inverse of the $P \times M$ dynamic matrix.
- **STEP 4:** Calculate the error matrix

$$E = A_a - AP_a^T$$

- **STEP 5:** Evaluate the error by Variance-Explained (VE)

$$r = 1 - \frac{Var(E)}{Var(A_a)}$$

  where $r \in [0, 1]$. A perfect decomposition gives $r = 1$, *i.e.* 100% explained.
- **STEP 6:** If r and $Cond(A)$ are acceptable, *e.g.* $r \ge 95\%$(user specified) and $Cond(A) \le 100$(user specified), then stop and choose the M as the control horizon. Otherwise, go back to **STEP 2** with $M = M - 1$.

Note that the variance of a matrix is defined as the sum of the variances of the column vectors, *e.g.*

$$Var(E) = \sum_{j=1}^{P} Var(\bar{e}_j)$$

Since $M = 1$ gives a matrix A with a condition number $Cond(A) = 1$, this optimization problem is guaranteed to have a solution.

After the decomposition, typically 95% of the dynamics in the full sized matrix A_a can be explained by its subset A. In other words, the residual part in E is not statistically significant. An example in a later section will illustrate the above procedure.

4. **Control Action**

The least squares solution for the control moves over a finite horizon in the future

$$\Delta U_m(k) = A^*(Y_{sp}(k) - \Phi_p X(k))$$

can not make the future output perfectly match the setpoint trajectory. Actually, the full sized control action $\Delta U(k)$ is usually called 'deadbeat' control since it can make any arbitrary state reach the desired location within P steps. The new reduced horizon control profile is a linear combination of the 'deadbeat' profile in which

$$\Delta U_m(k) = P_a^T \Delta U(k)$$

But it is not necessary to re-calculate $\Delta U(k)$ by inverting the P_a^T matrix since ΔU_m has a very clear physical meaning. Even though the reduced control input profile does not produce perfect control, it has better robustness. The output deviation produced as a result of this slower control action can be evaluated as follows.

The future outputs produced by the reduced horizon control profile are:

$$\hat{Y}_p(k) = \Phi_p X(k) + A \Delta U_m(k)$$

and the deviation vector

$$\begin{aligned} e(k) &= Y_p(k) - \hat{Y}_p(k) \\ &= A_a \Delta U(k) - A \Delta U_m(k) \\ &= A P^T \Delta U(k) + E \Delta U(k) - A \Delta U_m(k) \\ &= E \Delta U(k) \end{aligned}$$

With a successful decomposition, most components of A_a are represented by A and P_a. The variance of the error matrix, E, is typically less than 5% of that of A_a, *i.e.*

$$\begin{aligned} \|e(k)\| &\leq 5\% \|A_a \Delta U(k)\| \\ &\leq 5\% \|Y_{sp}(k) - \Phi_p X(k)\| \end{aligned}$$

where the $Y_{sp}(k) - \Phi_p X(k)$ term is the feedback residual term used for the state feedback control. The above derivation clearly shows that, after the implementation of the first reduced horizon control profile, the difference between the setpoint and the process output has been reduced by 95% (Note that only one 'deadbeat' control profile $\Delta U(k)$ is needed to completely eliminate the difference). For practical applications, the performance using a reduced control horizon can be quite satisfactory. Actually only two more control moves are needed to reduce the output error close to zero, *i.e.* $(5\%)^3 = 0.0125\%$. In other words, if the deadbeat control can make the output match the desired setpoint within P steps, the reduced control horizon method can almost certainly make it with $P+2$ steps. The advantage is that the reduced control horizon method gives a much better robustness without losing too much of the control speed and accuracy.

5. Illustration

Consider the design of a predictive controller for a SISO process with an under-damped response. The process transfer function is

$$\frac{y(q^{-1})}{u(q^{-1})} = \frac{0.0014q^{-1} + 0.0054q^{-2}}{1 - 1.94q^{-1} + 0.9527q^{-2}}$$

Due to its under-damped response, this process would require over 150 steps to reach the steady state (Figure 4.1). But, for predictive control purposes, a smaller value of the prediction horizon, *e.g.* $P = 20$, is better from a computational point of view. A full sized dynamic matrix A_a is 20 by 20 with poor conditioning (condition number $= 7.44 \times 10^{14}$). Reduced control horizon methodology can be applied to this process. The corresponding matrix decomposition results are summarized in Table (4.1).

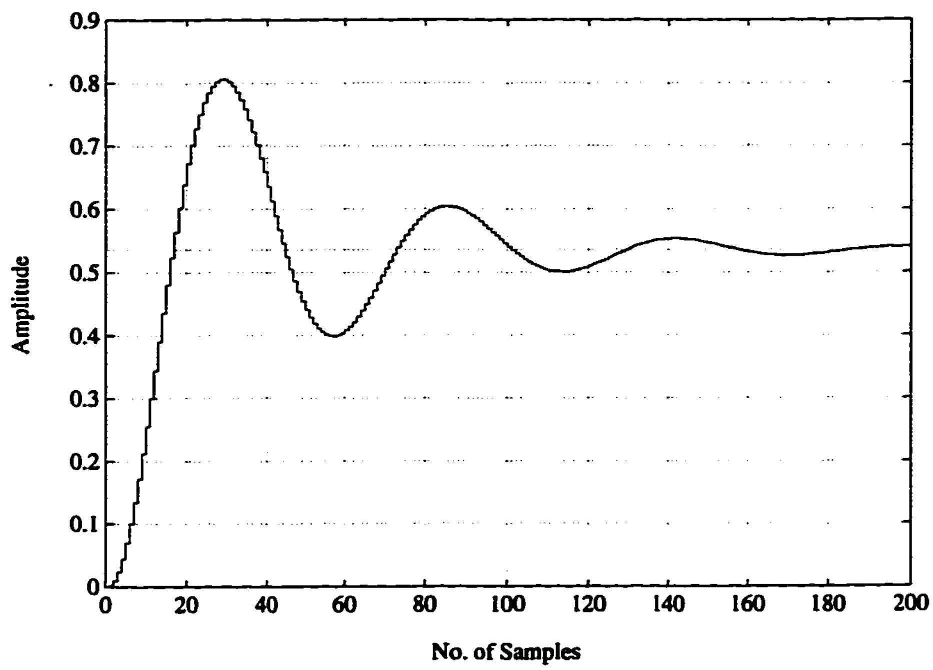

Figure 4.1:Example: The Step Response of An Under-damped Process

Obviously, for this process, the control horizon can be chosen as $M = 2$ or $M = 3$ which gives a good conditioning of the dynamic matrix ($\sim$ 100) and small matrix error ($< 1\%$). The corresponding MPC control system is stable with a dominant (discrete) pole of 0.8008 or 0.2277 respectively.

Table 4.1:Dynamic Matrix Conditioning using Reduced Control Horizon

Control Horizon M	Cond. Number	Var. Explained, r	Closed Loop Pole
1	1	95.93%	$0.928 + 0.0973j$
2	41.97	99.28%	0.8008
3	668.81	99.65%	0.2277
4	2.49×10^3	99.75%	$-0.2428 + 0.0073j$
20	7.44×10^{14}	100%	-0.2593

4.4.2 Matrix Weighting

Another method to change the matrix condition is to put penalty factors in the optimization objective function for predictive control, *e.g.*

$$\mathcal{J} = \sum_{j=1}^{P} \gamma_j [y_{sp}(k+j) - y(k+j)]^2 + \sum_{j=1}^{M} \lambda[\Delta u(k+j-1)]^2 + \gamma_s [y_{sp}(\infty) - y(\infty)]^2 \quad (4.9)$$

which is equivalent to using weightings on the elements of the dynamic matrix. A generalized matrix can be obtained which improves the condition number of the dynamic matrix substantially. The pseudo inverse required for the control calculation becomes

$$(A^T \Gamma_y A + \lambda I + \gamma_s A_s^T A_s)^{-1} A^T \Gamma_y$$

The commonly used weightings include control weighting λ, steady state weighting γ_s and output weighting γ_j (Γ_y is a matrix with γ_j as its diagonal elements). Generally, λ and γ_s improve the control performance by avoiding overly aggressive control. The negative part is that if they are too large, the optimization puts the emphasis on reducing the control moves instead of the output errors, which results in slow output tracking and poor control accuracy. Therefore, the recommended procedure for designing a predictive controller is to first specify a reduced control horizon that will bring the matrix condition to an acceptable region. Then, weighting parameters can be used to fine tune the matrix conditioning and control performance. In this way, the weightings turn out to be reasonable values.

1. **Control Weighting, λ**

The concept of using control weighting λ was introduced long before the predictive control technique emerged. A successful application is Generalized Minimum Variance(GMV) control which uses λ-weighting to overcome the internal instability of Minimum Variance(MV) control. λ-weighting can also be used to constrain the control movement of non-square MIMO systems, *e.g.* (Treiber 1984). Since MV control can be considered as a special case of predictive control with $M = P = 1$, it is quite natural to use the control weighting in MPC to avoid overly aggressive control action. As a matter of fact, it is the most commonly used and effective tuning parameter in practical applications of predictive control. The general tuning guideline is that the larger the weighting, the slower the closed loop response and the more robust the control system. However, without a quantitative design procedure that determines its effect on the controller, the control weighting λ can only be determined by trial and error. Just recently, the effect of λ on the dynamic matrix condition has been studied (Wilkinson et al. 1994).

Obviously, with control weighting, the numerical condition of the matrix to be inverted when calculating the control action, *i.e.* $A^TA + \lambda I$, can be improved significantly.

Assuming that the largest and smallest singular values (which are defined as the square root of the eigenvalues of the matrix A^TA) of the original dynamic matrix A are S_{max} and S_{min} respectively, then,

$$Cond(A) = \frac{S_{max}}{S_{min}} = C_0$$

With the control weighting λ, the condition number becomes

$$Cond(A, \lambda) = \left(\frac{S_{max}^2 + \lambda}{S_{min}^2 + \lambda}\right)^{1/2} \tag{4.10}$$

It is very easy to prove that the condition of A can be improved since

$$C_0 = Cond(A, \lambda = 0) > Cond(A, \lambda > 0)$$

Note that Equation (4.10) can be used directly to calculate a value of λ that results in good numerical conditioning of the generalized matrix. By specifying a desired condition number C_{max},

$$Cond(A, \lambda) = \left(\frac{S_{max}^2 + \lambda}{S_{min}^2 + \lambda}\right)^{1/2} \leq C_{max} \tag{4.11}$$

which leads to

$$\lambda \geq \frac{S_{min}^2(C_0^2 - C_{max}^2)}{C_{max}^2 - 1}$$

For example, to change the original matrix with $C_0 = 1.0 \times 10^5$ to $C_{max} = 50$, the control weighting should be $\lambda > 1.35$.

REMARKS:

- Generally, the worse the matrix condition, the larger the control weighting required;
- The magnitude of the control weighting is gain dependent since S_{min} is a function of the process gain. If a normalized dynamic matrix is used, then a revised relationship for the control weighting is

$$\frac{\lambda}{G^2} \geq \frac{S_{n,min}^2(C_0^2 - C_{max}^2)}{C_{max}^2 - 1}$$

  where G is the process gain. Obviously, to obtain the same matrix condition, a high gain process needs a high λ while a low gain process requires a lower λ. This conclusion can also be obtained from looking at the objective functions where the control weighting term has to be gain dependent to balance the the first output error term.
- This method of calculating the control weighting is also applicable to the design of a predictive controller to put different weight on each individual control move, *e.g.* λ_1 for $\Delta u(k)$ and λ_2 for $\Delta u(k+1)$ or for MIMO processes. But, there is no explicit design procedure to calculate the weightings. A trial and error procedure must be performed for proper choice of the weighting parameters.

Since the control weighting parameter can significantly improve the predictive controller, it is frequently used for tuning the controller and sometimes the only parameter provided by some commercial predictive control software, *e.g.* the control suppression factor in DMC. Even though most parameters are determined off-line, the control weighting can also be adjusted on-line to adapt to changes in the process dynamics (Wilkinson et al. 1994).

2. **Steady State Weighting, γ_s**

Steady state weighting is another important tuning parameter for predictive controllers (Kwok & Shah 1994, Saudagar 1995). For open loop stable processes, it emphasizes the predicted steady state error term relative to the dynamic errors. Conceptually (as well as mathematically), it stabilizes the controller. Due to the lack of quantitative guidelines for the selection of γ_s, it is generally chosen by trial and error.

From the point of view of dynamic matrix conditioning, this parameter can be easily incorporated since the matrix inverse becomes

$$A^+ = (A^T \Gamma_y A + \gamma_s^2 A_s^T A_s)^{-1}$$

where the matrix A_s consists of the steady state gain of the process (as discussed in Chapter 2.)

Obviously, a quantitative measurement of the effect of γ_s can be made in a way similar to that used for the control weighting λ. The differences are that

(a) γ_s, together with the steady state gain, is added to every element of the dynamic matrix since A^+ can be written as

$$A^+ = (A^T\gamma_y A + \gamma_s^2 S_s^2 \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix})^{-1}$$

while λ changes the diagonal elements only.

(b) the use of γ_s introduces a rank deficient unit matrix into the dynamic matrix. Mathematically, it can not change the rank of the dynamic matrix but may affect its conditioning. As discussed above, the dynamic matrix is full rank but may be ill-conditioned. The proposed design procedure selects a value for γ_s which improves the conditioning of the matrix.

(c) the effect of γ_s is directly affected by MPM in the process gain while that of λ is indirectly influenced.

(d) The addition of γ_s also introduces an extra state variable, *i.e.* the steady state variable, into the feedback controller and thereby stabilizes the controller further. However, these effects are not reflected by the condition of the dynamic matrix. This and related issues are beyond the scope of dynamic matrix conditioning and therefore are not covered here.

The effect of γ_s on the dynamic matrix is usually evaluated by calculating the eigenvalues and condition number of A^+ numerically. A typical example is shown in Figure 4.2 where the same process in Figure 4.1 is used to build the dynamic matrix. Obviously, as γ_s increases, the dynamic matrix condition of A^+ improves. The difference between using either λ or γ_s to improve the matrix condition is also clearly shown in Figure 4.2. A small increase in λ results in a dramatic improvement in terms of the matrix condition. On the other hand, the effect of γ_s is much slower and limited. (However, the actual closed loop performance should also be examined to determine the full effect of γ_s).

A similar method has been included in the commercial DMC algorithm by the addition of input steady-state weighting λ_s (Qin & Badgwell 1996). At steady state, input and output variables simply follow the state gain relationship. Mathematically, for the unconstrained solution, it is equivalent to the output steady-state weighting γ_s discussed here. Therefore, the methodology for selecting γ_s can be also used to choose the input steady-state weighting. In DMC, the physical interpretation of λ_s is very interesting. It includes a new equation restricting that the sum of all future control inputs equal to the optimal steady-state target, *i.e.*

$$u_{k-1} + \{\underbrace{\Delta u(k|k) + \Delta u(k+1|k) + \ldots + \Delta u(k+M-1|k)}_{\text{M-step control moves}}\} = U_{ss}$$

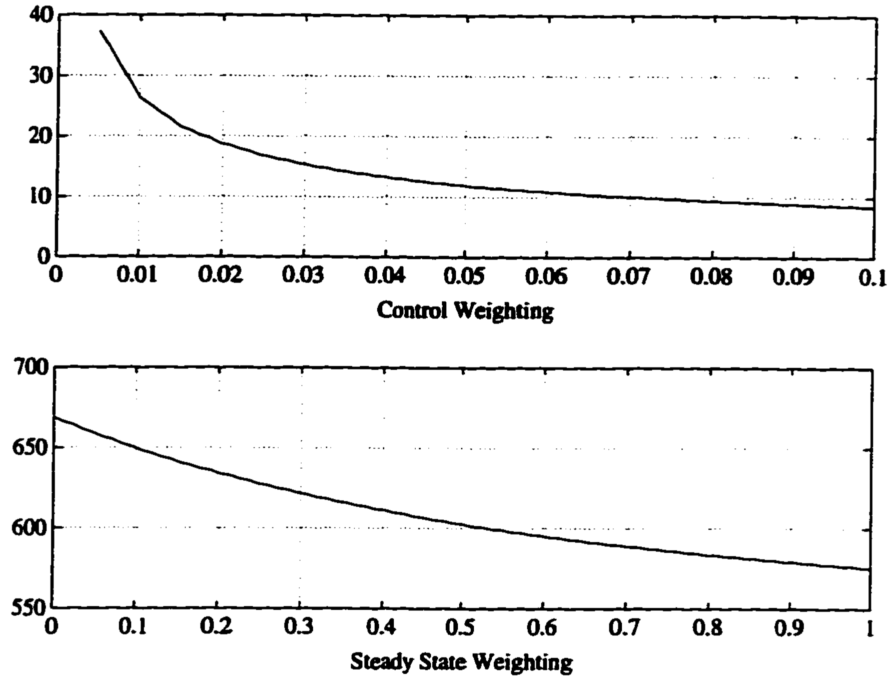

Figure 4.2: Effect of λ (top, with $\gamma_s = 0$) and γ_s(bottom, with $\lambda = 0$) on the condition of the dynamic matrix

This equation is not a hard constraint but imposed in the least squares sense in the control implementation instead. After simple manipulation, the equation becomes

$$[1,\ 1,\ \cdots,\ 1]\begin{bmatrix} \Delta u(t|t) \\ \Delta u(t+1|t) \\ \vdots \\ \Delta u(t+M-1|t) \end{bmatrix} = U_{ss} - u_{t-1}$$

and can be added as an extra row element, with a weighting factor, in the dynamic matrix. Obviously, it has the same effect as γ_s on the dynamic matrix.

Note that mathematically, the condition of the dynamic matrix can also be changed by the output weighting matrix Γ_y. Practically, these parameters are related to the relative importance of output deviations as determined by process requirements, economics and quality control. Therefore, usually, their values can not be chosen freely for the purpose of condition improvement.

4.4.3 Summary and Illustrations

The recommended procedure for the design of the nominal model predictive controller can be summarized as:

- **STEP 1:** choose the output prediction horizon P and associated relative output weightings Γ_y;

- **STEP 2:** build the full sized P by P dynamic matrix A_a;
- **STEP 3:** find the reduced control horizon M that improves the condition of the dynamic matrix A to within an acceptable region, *e.g.* ≤ 100;
- **STEP 4:** use the control weighting λ and/or the steady state weighting γ_s to fine tune the matrix conditioning and if desired check the closed loop performance by simulation;
- **STEP 5:** implementation.

A MIMO distillation process is used here to illustrate the design procedure of MPC. This process has 2 inputs and 2 outputs and was modeled by Wood and Berry (Wood & Berry 1973) as:

$$\begin{bmatrix} X_d \\ X_b \end{bmatrix} = \begin{bmatrix} \frac{12.8e^{-s}}{16.7s+1} & \frac{-18.9e^{-3s}}{21.0s+1} \\ \frac{6.6e^{-7s}}{10.9s+1} & \frac{-19.4e^{-3s}}{14.4s+1} \end{bmatrix} \begin{bmatrix} R_f \\ V_b \end{bmatrix}$$

R_f: the reflux flow.

V_b: the vapour boil-up rate.

X_d: the mole fraction methanol in the distillate.

X_b: the mole fraction methanol in the bottoms.

This is a strongly interacting, 2×2, process with time-delays. With a control interval $T_s = 2.5 \;\; min$, the maximum number of steps to reach steady state is 40 (Figure 4.3). Choosing the output prediction horizon P as 20, the full sized dynamic matrix A_a for this MIMO process results in an ill-conditioned, 40×40 matrix.

A reduced control horizon M is used first to improve the condition of the matrix, followed by the addition of control weighting λ. The whole procedure is shown in Table 4.2. Note that the dominant closed loop pole is also calculated to check the stability of the MPC control system.

Note that for this 2×2 distillation process, a closed loop stable MPC system can be achieved by using a reduced control horizon. For example, choosing $M = 8$ gives a stable MPC system but an ill-conditioned dynamic matrix. A control weighting of $\lambda = 100$ can be used to improve the matrix conditioning. Increasing λ does not significantly decrease the controller performance under nominal conditions, *e.g.* the dominant closed loop pole location is only a little worse than that without λ weighting , 0.9846 *vs.* 0.9816. But increasing λ significantly improves the robustness of the control system. The condition number of the dynamic matrix is improved from 451.73 to 152.47. This example shows that, in general, there is trade-off between performance and robustness, *i.e. increased robustness is obtained at the cost of poorer closed-loop performance* but the trade-off is example dependent.

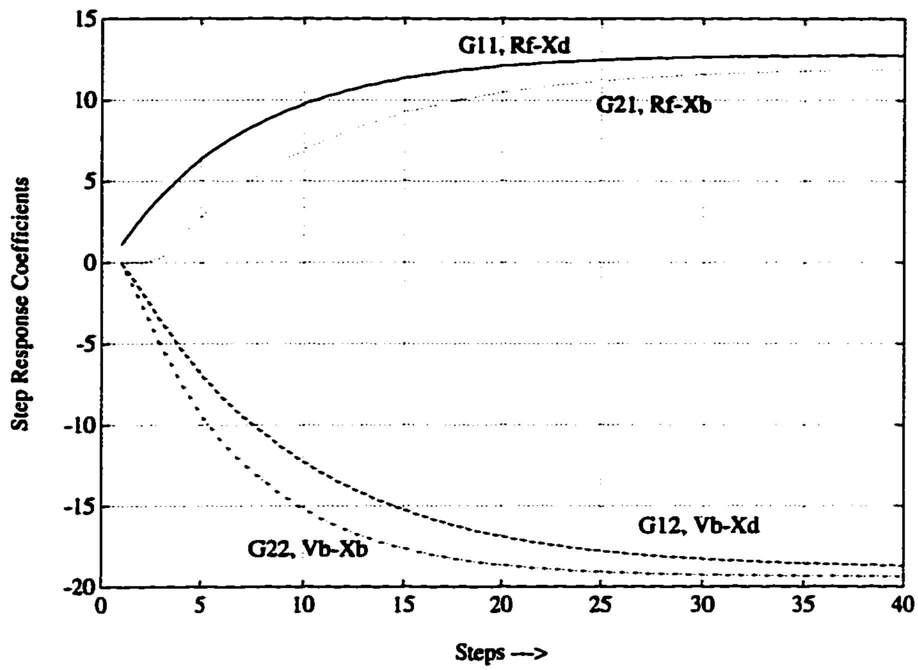

Figure 4.3: The Step Responses of the Wood and Berry Distillation Column

Table 4.2:The Selection of Control Parameters

Methods	Condition Number of A	% VE	Closed-Loop Pole
"Perfect Control" $P = M = 20$	∞	100	0.0
Reduced control horizon with $\lambda = 0$			
$M = 1$	1.0	84.23	0.9719
$M = 2$	28.23	88.32	0.9850
$M = 3$	127.90	89.45	0.9827
$M = 5$	302.52	91.02	0.9819
$M = 8$	451.73	92.52	0.9816
Control Weighting with $M = 8$, $\gamma_s = 0$ $\lambda = 100$	152.47	N/A	0.9846

The controller gain matrix in Equation (4.5) has the following structure

$$K_{mpc}^T = \left[\begin{array}{c|c}
0 & 0 \\
0.8227 & 0.0183 \\
0.0622 & -0.0279 \\
0.0044 & 0.0231 \\
-0.0794 & -0.0042 \\
0.1273 & -0.0209 \\
-0.1092 & 0.0379 \\
0.0487 & -0.0341 \\
-0.0034 & -0.0011 \\
-0.0102 & 0.0075 \\
-0.0075 & 0.0057 \\
-0.0051 & 0.0041 \\
-0.0031 & 0.0028 \\
-0.0013 & 0.0017 \\
0.0003 & 0.0008 \\
0.0017 & -0.0000 \\
0.0029 & -0.0007 \\
0.0040 & -0.0013 \\
0.0050 & -0.0017 \\
0.0058 & -0.0022 \\
0.0065 & -0.0025 \\
------ & -------- \\
0 & 0 \\
-0.0419 & -0.3791 \\
-0.0015 & -0.0155 \\
0.0515 & 0.0031 \\
-0.0831 & 0.0136 \\
0.0700 & -0.0255 \\
-0.0186 & 0.0251 \\
-0.0586 & -0.0096 \\
0.0993 & -0.0152 \\
-0.0132 & 0.0069 \\
-0.0107 & 0.0054 \\
-0.0085 & 0.0041 \\
-0.0066 & 0.0028 \\
-0.0051 & 0.0018 \\
-0.0037 & 0.0008 \\
-0.0026 & -0.0001 \\
-0.0017 & -0.0009 \\
-0.0009 & -0.0016 \\
-0.0002 & -0.0022 \\
0.0004 & -0.0028 \\
0 & 0
\end{array}\right]$$

4.5 Special Cases

Since the dynamic matrix elements are step response coefficients, its condition is sensitive to the characteristics of the actual process. Two special process characteristics, time delay and nonminimum phase, deserve more attention during the design of predictive controllers. The use of unit step response coefficients in MPC means that both time delay and nonminimum phase appear explicitly.

4.5.1 Processes with Time Delay

The time delay of a process appears as zero elements in the initial step response. If included in the dynamic matrix, it would certainly degrade the numerical condition of the matrix. Even though the known time delay can be excluded from the dynamic matrix, unknown time delays and/or its drift can affect the condition dramatically.

Predictive control, with its long range prediction, can handle time delays through proper use of the tuning parameters. For example, to control a process with time delay, a full sized predictive controller can not be used since its matrix inverse does not exist. The reduced control horizon method and/or control weighting method can, however, be applied to handle the unknown time delay.

After building the dynamic matrix, the controller parameters can be selected in the same way as discussed before. Due to the effect of time delay, the actual effective prediction horizon has changed from P to $P-d$ (where d is the implicit time delay). This can be easily proven by

$$A_{P\times M} = \begin{bmatrix} 0_{d\times M} \\ A_{(P-d)\times M} \end{bmatrix}$$

and

$$A^T_{P\times M}A_{P\times M} = A^T_{(P-d)\times M}A_{(P-d)\times M}$$

Obviously, the output prediction horizon P must be greater than the possible time delay d.

4.5.2 Processes with Nonminimum Phase Behaviour

Unstable zeros in a process have an effect similar to time delays. In the step response coefficients, they usually appear as a fast but reverse response during the initial process response. Conceptually, in order to capture the process response required for effective control, the output prediction horizon should be much larger than the nonminimum phase period. Another effective approach is to use an initial prediction horizon (*e.g.* N_1 in GPC) to exclude the first few rows of the dynamic matrix. Unlike the time delay, the NMP will still have negative effects on the dynamic matrix even with the application of the initial prediction horizon. Generally, the control parameter selections are trial and error procedures. The effect of NMP can also be evaluated by the condition number of the dynamic matrix to ensure satisfactory performance.

4.6 Conclusion

A predictive control design procedure has been developed based on the structure and numerical properties of the dynamic matrix. Using a single analytical objective, improving the matrix condition, all tuning parameters in the predictive controller can be selected. The suggested procedure for the design of a predictive controller is to decompose the dynamic matrix first to determine the control horizon. Then, weightings are selected to further improve the condition of the matrix. This new design scheme facilitates the control design especially for multivariable systems. A simulation example using a 2×2 interacting process illustrates the design for an application of predictive control.

Chapter 5

Disturbance Handling: Feedforward and Feedback

5.1 Introduction

In practice, a process is usually subjected to many disturbances from either known or unknown sources. Therefore, the major controlled variables, the process outputs, are functions of not only the manipulated variables but also of disturbances. For the setpoint tracking problem of MPC, many input/output models, including DARMA, step response coefficient, regular state space and the dual-model state space discussed in the previous chapters, have been extensively studied for the output predictions. On the other hand, the regulatory control problem which includes estimating output predictions as a function of disturbances remains a challenging research area. This problem is especially important for chemical process control where the primary concern is usually to maintain the process conditions in the presence of disturbances.

Generally, the disturbances include measurable/unmeasurable variables, random noise and/or modelling error. Both deterministic and stochastic models can be used to describe how the process outputs are affected. Obviously, in contrast to traditional PID control, predictive control should have the ability to estimate what will happen in the future so that immediate control action can be taken to compensate these future upsets. Predictive control is an effective method for regulatory control and has been successfully applied to many chemical processes.

Many disturbances to a process can be identified easily and are practically measurable. For example, ambient conditions such as temperatures, pressure and/or humidity have a significant influence on some petrochemical processes. For measurable disturbances, a feedforward scheme can be used. With deterministic models, the future output profiles due to these measurable disturbances can be readily calculated. Then, traditionally, they are subtracted from the setpoint trajectory. During the optimal minimization, the MPC controller considers them as the setpoint changes and make control moves to compensate these measurable disturbances. An important issue, *how well can the MPC controller eliminate disturbance*, was often ignored in the past due to the fact that most predictive controllers come with feedback options which estimate the output deviations caused by the disturbances. *Actually, MPC has a 'built-in' model feedback loop (Figure 5.1) as a result of the minimization index.* Even without the extra output feedback loop, the internal model feedback still works to minimize the error caused by either setpoints or disturbances in the forward channel. In a classical control sense, some compensation is made via feedback rather than feedforward. This increases the difficulty for feedback design especially for a state estimation based method.

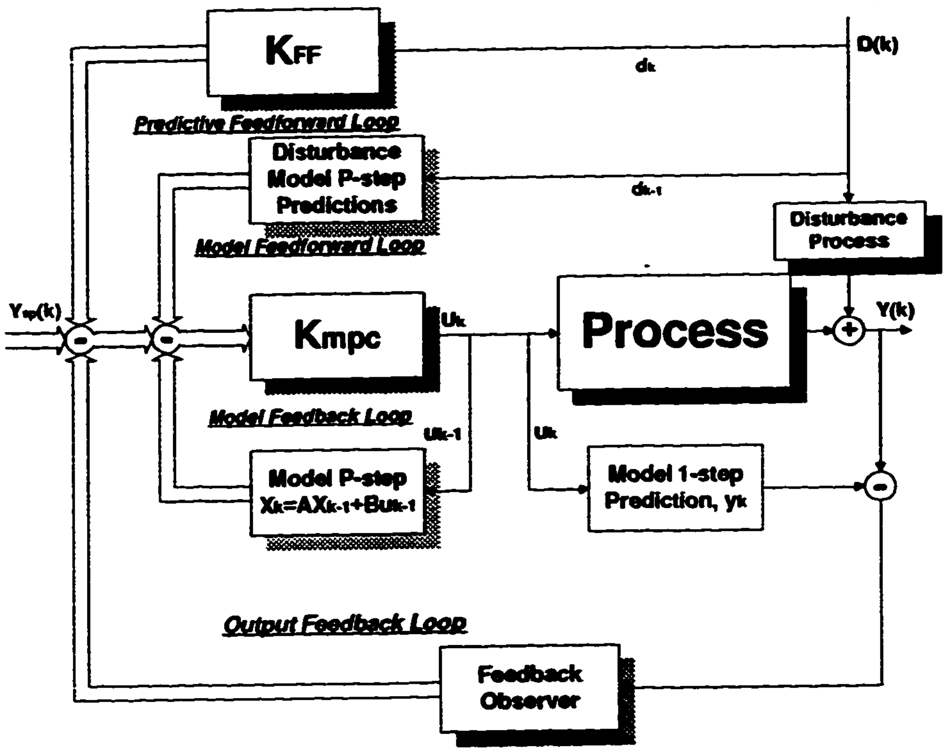

Figure 5.1: Disturbances and Feedforward, Feedback Loops in The MPC Control Scheme

In the first part of this chapter, it is shown that the capabilities of state feedback based MPC controllers to eliminate disturbances have been overestimated. It can not eliminate the disturbance in a feedforward sense and hence relies on feedback. A new separate feedforward control scheme, *predictive feedforward*, is put forward to address this design deficiency.

For unmeasurable disturbances, model errors, *etc*, the effects on the process outputs have to be estimated. Convergence and optimality are major concerns for these estimations. In the dual model state space formulation, the state variables are directly defined as the process output predictions. Therefore, classical state observer theory can be applied. Direct application of optimal based, state observer design to predictive control in a state space form leads to a large dimensional algebraic Riccati solution (Navratil et al. 1988). Later, Lee and Morari proposed a much simplified observer design scheme for MPC which assumes a special model for the disturbance (equivalent to introducing a integrator in the feedback loop) (Lee et al. 1993). In the second section of this chapter, a different approach based on pole-placement is developed. Effective approaches which use a calculated feedback gain, a feedback horizon (Saudagar 1995) and a rotating factor, are developed.

5.2 Dual Model with Disturbances

Disturbance variables can be easily incorporated into the dual model state space formulation in the form of measurable disturbances and/or unmeasurable disturbances/random noises. In addition to the manipulated variable $\Delta u_(k)$, process noise and measurement noise can be represented as:

$$\begin{aligned} X(k) &= \Phi X(k-1) + \theta \Delta u(k-1) + \Gamma \Delta d(k-1) + T \Delta w(k-1) \quad (5.1) \\ Y(k) &= H X(k) + v(k) \end{aligned}$$

where Γ consists of the step response coefficients of the process output with respect to the measured disturbances $d(k)$. T is a suitable step response model for the unmeasured or random input $w(k)$. For example, white noise is represented by a T-vector with unit elements, and coloured noise is represented by a T-vector containing elements with a first order or second order dynamics. This arrangement splits the measurable disturbance term and the unmeasurable or random noise term to facilitate feedforward and feedback design.

The future prediction in Equation (4.1) can be rewritten as:

$$Y_p(k) = \Phi_p X(k) + A\Delta U(k) + B\Delta D(k) + C\Delta W(k) \quad (5.2)$$

The A matrix consists of the step response coefficients and is usually called the 'Dynamic Matrix'. Correspondingly, the B and C are also in matrix form with a structure similar to A. It is reasonable to define them as 'Dynamic Feedforward Matrix' and 'Dynamic Feedback Matrix'.

The $D(k)$ and $W(k)$ vectors include *future* deterministic disturbances and random noise in a manner analogous to $U(k)$. Obviously, by assumption, the disturbance $d(k)$ at the current time step is measurable so that at least the first element of $D(k)$ is available for control applications. On the other hand, the effect of future disturbances and the random noise $W(k)$ on the state variables can only be estimated. Therefore, in the next two sections, these two issues will be considered in more detail.

5.3 Predictive Feedforward Design

Considering the measurable disturbances only, the future output prediction in Equation (5.2) becomes:

$$Y_p(k) = \Phi_p X(k) + A\Delta U(k) + B\Delta D(k) \quad (5.3)$$

To design a feedforward controller successfully, both the B matrix (*i.e.* the disturbance model) and the future disturbance inputs should be available. Otherwise, this term would have to be treated via feedback design instead of feedforward. Feedforward control is preferred since it does not cause any stability problems and theoretically perfect disturbance rejection can be achieved. As shown later, the predictive feedforward approach has many advantages over traditional feedforward design methods.

5.3.1 Disturbance Profile

For simplicity, consider a single disturbance trajectory $\Delta D(k)$ with a dimension of N_{FF}. Note that an explicit explanation for this assumption is that there will be N_{FF} steps in the future disturbance. The disturbance is assumed to be constant after N_{FF} steps. With $N_{FF} = 1$, only one disturbance is assumed to occur. Since the value of the only disturbance can be determined by on-line measurement, no prediction of future values of this disturbance is needed. This method has been widely used in the past for the feedforward design in predictive controllers. However, in general

$$\Delta D(k) = [\Delta d(k),\ \Delta d(k+1),\ \ldots,\ \Delta d(k+N_{FF})]^T \tag{5.4}$$

The only available measurement of the disturbance $d(k)$ is at time instant k. Future disturbances may or may not have a fixed relationship with current and past disturbance values. However, for the purpose of simplicity, introduce a vector K_{FF} which satisfies:

$$\begin{aligned} \Delta D(k) &= [\Delta d(k),\ \Delta d(k+1),\ \ldots,\ \Delta d(k+N_{FF}-1)]^T \\ &= [f_1,\ f_2,\ \ldots,\ f_{N_{FF}}]^T \Delta d(k) \\ &= K_{FF}\Delta d(k) \end{aligned} \tag{5.5}$$

where, traditionally in MPC, *e.g.* DMC, $f_1 = 1, f_2 = \cdots = f_{N_{FF}} = 0$.

5.3.2 Feedforward Compensator

For regulatory control, the prediction equation (5.3) defines the objective as

$$\Phi_p X(k) + A\Delta U(k) + BK_{FF}\Delta d(k) = 0 \tag{5.6}$$

To compensate the disturbance, the required future control action $U(k)$ is:

$$\begin{aligned} A\Delta U(k) &= -\Phi_p X(k) - BK_{FF}\Delta d(k) \\ &\text{or} \\ \Delta U(k) &= A^*[-\Phi_p X(k) - BK_{FF}\Delta d(k)] \end{aligned} \tag{5.7}$$

Note that the pseudo-inverse of the dynamic matrix includes all the design parameters discussed in Chapter 4. By applying the receding horizon principle and separating the feedback and the feedforward terms, the current control action required to reject the disturbance can be represented as:

$$\Delta u(k) = -K_{mpc}X(k) + C^T A^* B K_{FF} \Delta d(k) \tag{5.8}$$

Applying this control action to the state space equation, yields

$$\begin{aligned} X(k) &= (\Phi - \theta K_{mpc})X(k-1) + (-\theta C^T A^* B K_{FF} + \Gamma)\Delta d(k-1) \\ Y(k) &= H\,X(k) + v(k) \end{aligned} \tag{5.9}$$

Equation (5.9) shows that the MPC controller uses state feedback to correct the residual effect of measurable disturbances, if they are not completely rejected. The 'built-in' state feedback, $\Phi - \theta K_{mpc}$, comes from the quadratic optimization. Obviously, the disturbance term in (5.9) does not change the state feedback property $(\Phi - \theta K_{mpc})$ (which determines the closed loop stability). As long as the system is stable, state feedback alone would eliminate the effect of the disturbance on the state and/or output. However, if possible, it is better to use the disturbance term in (5.9) to improve regulatory control performance.

1. **Dynamic State Compensator**

   In Equation (5.9), the effect of the disturbance on all state variables may be eliminated if

$$-FK_{FF} + \Gamma = 0 \tag{5.10}$$

   where

$$F = \theta C^T A^* B$$

   Every matrix/vector in Equation (5.10) is known except the disturbance profile vector K_{FF} which has a dimension of $N_{FF} \times 1$ for a single disturbance variable. The state feedforward controller can be obtained as:

$$K_{FF} = (F)^* \Gamma$$

   where F^* is, in general, a pseudo-inverse. Usually, this solution can not completely remove the disturbance effect in the state equation because:

   - With $N_{FF} < n$, it is under-determined, *i.e.* has fewer unknown parameters than equations. Therefore, a least square solution which minimizes the effect of the disturbance on the states is used.
   - With $N_{FF} = n$, it is theoretically possible to get a perfect solution depending on the condition of the matrix $F^T F$. If this matrix is full rank, then a complete rejection of the disturbance can be achieved.
   - Considering the components of the matrix F, it is most likely not full rank. In that case, a weighting matrix γ similar to the control penalty matrix should be used to improve the condition of the matrix. The feedforward controller gain can be calculated by:

$$K_{FF} = (F^T F + \gamma)^{-1} F^T \Gamma$$

- Problems also arise due to time delays in the input and disturbance models Θ, B and A. In general, this requires impractical solutions for singular matrix F.

Using a third order process model $1/(s+1)(3s+1)(5s+1)$, the dynamic compensator design can be shown as follows. The control parameters $P = 9, M = 2, \lambda = 0$ give a stable state feedback MPC controller (dominant poles $0.3402 \pm 0.1731j$). The disturbance model is $1/(10s+1)$. The feedforward gain K_{FF} can be obtained as:

$$
\begin{array}{ll}
N_{FF} = 1 & K_{FF} = 0.3104 \\
N_{FF} = 2 & K_{FF} = [0.2161,\ \ 0.1427]^T \\
N_{FF} = 5 & K_{FF} = [0.1968,\ \ 0.1300,\ \ 0.0596,\ \ -0.0014,\ \ -0.0420]^T
\end{array}
$$

Note that for $N_{FF} \geq 2$ cases, a small weighting $\gamma = 0.001$ should be used in the F matrix to avoid problems due to ill-conditioning.

2. **Steady-State Output Compensator**

Ideally, the control objective is to remove all effects of the disturbance on the state variables by feedforward. However, this requirement is often too strict to implement and may cause ill-conditioning problems. The control requirements can be relaxed by

(a) considering only the effects of the disturbance at the steady state;

(b) removing the effect of disturbance on the process output only, *i.e.* some linear combinations of the state variables instead of all state variables.

Then, assuming a steady state can be reached such that $X(k+1) = X(k), Y(k) = 0$, the state equation in (5.9) can be rewritten as:

$$
\begin{aligned}
(I - \Phi + \theta K_{mpc})X(k) &= (-\theta K_{mpc} B K_{FF} + \Gamma)\Delta d(k-1) \\
&\quad \textit{or} \\
X(k) &= (I - \Phi + \theta K_{mpc})^{-1}(-\theta K_{mpc} B K_{FF} + \Gamma)\Delta d(k-1)
\end{aligned}
$$

which gives the process output at the steady state as

$$
\begin{aligned}
Y(k) &= H\,X(k) \\
&= H(I - \Phi + \theta K_{mpc})^{-1}(-\theta K_{mpc} B K_{FF} + \Gamma)\Delta d(k-1) \qquad (5.11)
\end{aligned}
$$

Therefore, the objective for steady state feedforward control is

$$
H(I - \Phi + \theta K_{mpc})^{-1}(-\theta K_{mpc} B K_{FF} + \Gamma) = 0 \qquad (5.12)
$$

The steady-state feedforward controller requires solving the matrix equation,

$$
H(I - \Phi + \theta K_{mpc})^{-1}\theta K_{mpc} B K_{FF} = H(I - \Phi + \theta K_{mpc})^{-1}\Gamma \qquad (5.13)
$$

Now, the conventional method of including feedforward in the design of predictive controllers where $N_{FF} = 1, K_{FF} = f_1 = 1$ can be evaluated. Substitution of these values into (5.13) shows that there is no other tuning parameter to adjust the controller and hence the disturbance rejection performance is very limited. Even though it is simple in design, feedback has to function to reject the disturbance which may result in many problems including instability.

Using the same process model and disturbance model as in the previous example, the following values for K_{FF} are obtained

$$\begin{array}{ll} N_{FF} = 1 & K_{FF} = 1.8170 \\ N_{FF} = 2 & K_{FF} = [1.8170,\ \ 0]^T \end{array}$$

The disturbance rejection performance is shown in Figure 5.2 for the process outputs, and Figure 5.3 for the control efforts.

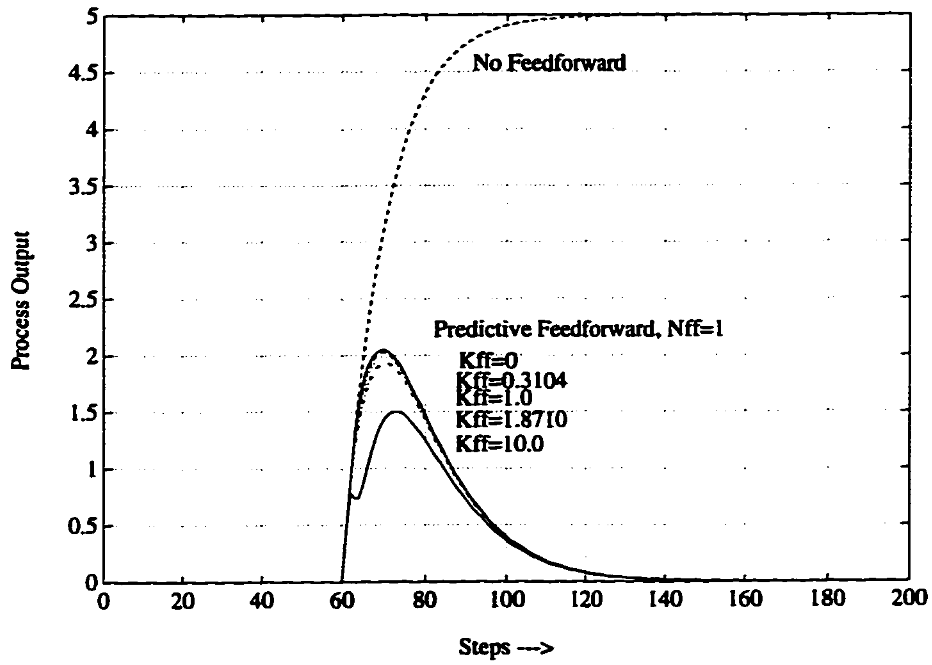

Figure 5.2:Disturbance Rejection by Predictive Feedforward

In these simulations, a disturbance horizon $N_{FF} = 1$ was chosen and different values of K_{FF} were applied. As mentioned above, even though this MPC feedforward scheme relies on the state feedback part, the feedforward parameter K_{FF} does influence the disturbance rejection performance. As K_{FF} increases, the controller rejects the disturbance faster by generating larger control inputs. A conclusion from this simulation is that the K_{FF}, as defined in Equation (5.5), can be considered as a tuning parameter in the MPC control scheme, particularly for the rejection of measurable disturbances.

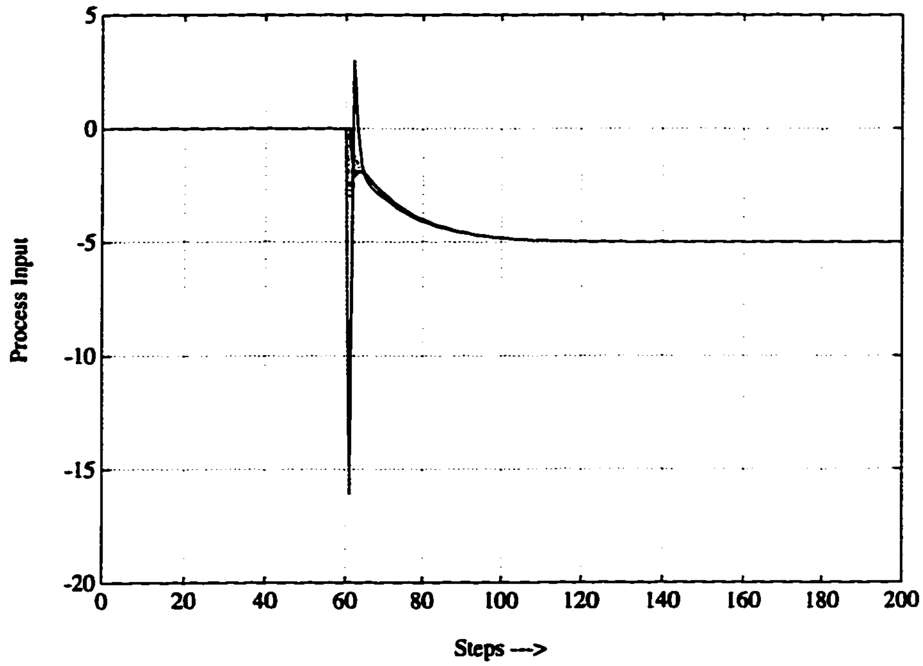

Figure 5.3:Control Actions to Reject Disturbance

5.4 Output Feedback Design Using A State Observer

In addition to the 'built-in' model feedback, output feedback is always included in MPC for two major reasons:

1. The state variables in the dual-model formulation are required to calculate the next control move for predictive control. However, only the first element can be directly measured from the process output. The other (observable) states need to be estimated from the current output measurements.

2. The existence of random noise, model uncertainty and unmeasurable disturbances makes the process deviate from the desired trajectory. Feedback should be used to estimate the effects so that the controller can make corresponding adjustments.

For state space models, classical state observer theory has existed for years and has been applied successfully as part of LQC type control. While algebraic derivations are sufficient to obtain the state variables from the output measurements of deterministic processes, optimal state observer design methodology such as the Kalman filter is required for processes with stochastic disturbances.

5.4.1 State Observer for The Dual-Model Formulation

With stochastic noise terms, the process with process noise and measurement noise can be described by a dual-model formulation as:

$$\begin{aligned} X(k) &= \Phi X(k-1) + \theta\,\Delta u(k-1) + T\,\Delta w(k-1) \\ Y(k) &= H\,X(k) + v(k) \end{aligned} \tag{5.14}$$

Note that the disturbance model is represented by both T and Φ. As defined in the dual model formulation, the T vector is the unit step response of the process output to the disturbance. The matrix Φ also includes the slow or unstable modes of the disturbance dynamics. This can be shown by using an input/output description of the process.

Assume a Box-Jenkins model with a disturbance term,

$$y(k) = \frac{B(q^{-1})}{A(q^{-1})}u(k-1) + \frac{C(q^{-1})}{D(q^{-1})}w(k-1)$$

For this process, assume that the denominator parts can be decomposed into two parts, fast and slow modes, as

$$A(q^{-1}) = A_1(q^{-1})A_2(q^{-1})$$

$$D(q^{-1}) = D_1(q^{-1})D_2(q^{-1})$$

Then, the process can be rewritten as

$$A_2(q^{-1})D_2(q^{-1})\Delta y(k) = \frac{B(q^{-1})D_2(q^{-1})}{A_1(q^{-1})}\Delta u(k-1) + \frac{C(q^{-1})A_2(q^{-1})}{D_1(q^{-1})}\Delta w(k-1)$$

Obviously, the same procedure used in Chapter 2 can be applied to obtain the parameters in the Φ matrix. The last row of the Φ matrix in the dual model formulation (Equation (2.13)) is

$$A_2 D_2 \Delta = 1 - r_1 q^{-1} - r_2 q^{-2} - \cdots - r_{n_r} q^{-n_r}$$

Note that, strictly speaking, there is no existing acronym that correctly describes this model form with the addition of disturbance term. But DARMA (arguably) is the closest one and is used in this thesis. Therefore, the similar procedure in Chapter 2 is applied here to handle the disturbance model.

REMARKS:

1. The inclusion of the slow disturbance model $D_2(q^{-1})$ inside the system matrix Φ significantly improves the disturbance rejection properties of MPC. Especially for slow disturbances similar to ramps, traditional DMC feedback design is unable to eliminate their effects. This issue will be illustrated later in an example.

2. The fast models for both the process and the disturbance are still described by the step response coefficients in θ and T.

Since the dual model formulation in Equation (5.14) is in a standard state space form with complete state observability, standard observer theory including Kalman Filter theory can be applied (Navratil et al. 1988, Morari & Lee 1991). For example, the two stage type observer gives:

$$\begin{aligned} \widehat{X}(k) &= \Phi X^*(k-1) + \theta \Delta u(k-1) \\ \widehat{Y}(k) &= H\widehat{X}(k) \\ X^*(k) &= \widehat{X}(k) + K[Y(k) - \widehat{Y}(k)] \end{aligned} \tag{5.15}$$

where $\widehat{X}(k)$ is the model based estimate of the states. $X^*(k)$ is the estimated state variable vector to be used by the control algorithm. $Y(k)$ is the actual output measurement in (2.13) and K is the generalized feedback observer gain.

Note that this is a very general model-based formulation for estimating state variables from process outputs. Many methods, differing mainly in how to calculate the feedback gain K, are available. The state convergence and optimality are major concerns of the state observer design.

5.4.2 Optimal State Observer Design

Given the covariance matrices W and V of the random noise terms $w(k)$ and $v(k)$, the observer gain based on Kalman filter theory can be obtained as a solution of the Riccati equation,

$$\begin{aligned} K &= PH^T(HPH^T + V)^{-1} \\ P_k &= \Phi P_{k-1}\Phi^T - \Phi P_{k-1}H^T(HP_{k-1}H^T + V)^{-1}HP_{k-1}\Phi^T + TWT^T \end{aligned} \tag{5.16}$$

The Kalman filter method, gives minimum variance estimates of the states. However, some problems arise in practical implementations of MPC.

- It requires the solution of a large dimensional complicated Riccati matrix equation;
- It needs detailed information about the disturbance model and statistical property, *i.e.* T, W and V.

Lee and Morari (1991) assumed a unit step type disturbance in the state equation, *i.e.*

$$T = [1,\ 1,\ \ldots,\ 1,\ 1]^T$$

and

$$T\Delta = [0,\ 0,\ \ldots,\ 0,\ 1]^T$$

Note that, mathematically, $T\Delta$ can be either $[0, 0, \ldots, 1]$ or $[1, 0, \ldots, 0]$. The former is preferred so that the noise is added to the last state variable first and gradually propagated to other state variables. Recall that in the dual-model formulation, the last state variable has more controllability than the others, so this arrangement of $T\Delta$ certainly has advantages for disturbance rejection.

With the assumption of T as a step type disturbance model and white noise in $\Delta w(k)$, the noise term $w(k)$ is an integrated noise. Therefore, this type of disturbance is equivalent to the integrated noise model in DARMA model used by GPC (without T-filter), and also the fundamental disturbance model assumed by DMC.

Then, the solution of the Riccati equation is much simplified, *i.e.* (Morari & Lee 1991):

- Open Loop Stable Processes,

$$K = f[1,\ 1,\ \ldots,\ 1]^T$$

  where $f \in [0,1]$ is a scalar depending on the Signal-to-Noise ratio of the output measurement;

- Integrating Processes,

$$K = f_a[1,\ 1,\ 1,\ \ldots,\ 1]^T + f_b[0,\ 1,\ 2,\ \ldots,\ n]$$

  where $f_a, f_b \in [0,1]$ are scalars.

Obviously, this disturbance model results in a very simple observer design which is optimal for some specific disturbance forms, *i.e.* step type disturbances. The scalars are treated as tuning parameters and knowledge of the noise properties is not required. However, as pointed out before, this formulation is identical to the integrated noise model for the DARMA model of GPC. It is also an implicit assumption by DMC, which functions well for step type disturbances. For general disturbance models, this feedback design may not work satisfactorily.

5.4.3 State Observer Design by Pole-Placement

Instead of dealing with the Riccati equation which may result in practical problems, a state observer can be obtained based on pole placement methods (Qi & Fisher 1993).

Define the state error vector

$$\overline{X}(k) = X(k) - X^*(k) \tag{5.17}$$

and combine Equation (2.13), (5.15) and (5.17). The following stochastic equation is obtained

$$\overline{X}(k) = (I - KH)\Phi\overline{X}(k-1) + (I - KH)T\Delta w(k-1) \tag{5.18}$$

For an asymptotically stable, *i.e.* state convergent, predictive observer, it is required that

$$\lim_{k\to\infty} \overline{X}(k) = 0$$

Since the characteristic equation of the observer is:

$$det[\lambda I \; - \; (I - KH)\Phi] = 0 \tag{5.19}$$

for an asymptotically stable observer, all eigenvalues of the observer should be within the unit circle. Without the extra output feedback, *i.e.* $K = 0$, the state convergence is determined by eigenvalues of the open loop process Φ. The feedback gain K can be designed to shift the original eigenvalues of Φ to the desired locations.

Assume

$$K = [k_1, \; k_2, \; \cdots, \; k_{n+1}]^T$$

Equation (5.19) can be expanded in terms of eigenvalues as:

$$\begin{aligned} det[\lambda I - (I - KH)\Phi] &= \lambda^{n+1} \\ &+ (k_2 - r_1)\lambda^n \\ &+ (k_3 - r_1 k_2 - r_2)\lambda^{n-1} \\ &+ \cdots \\ &+ (k_{i+1} - \sum_{j=0}^{i-1} r_{j+1} k_{i-j})\lambda^{n+1-i} \\ &+ \cdots \\ &+ (k_{n+1} - \sum_{j=0}^{n-1} r_{j+1} k_{i-j})\lambda \\ &= 0 \end{aligned} \tag{5.20}$$

where $r_j = 0$, if $j > n_r$.

From this equation, the properties of the MPC observer can be summarized as:

- The coefficients of the characteristic equation (CE) are independent of k_1. For convenience, let $k_1 = 1$.
- Equation (5.20) is a $(n+1)$th order polynomial but without a constant term so that at least one root is at the origin. *i.e.* one eigenvalue is $\lambda = 0$.

The feedback observer gain K can therefore be calculated using standard pole placement techniques to obtain the desired observer performance. Usually fast convergence for the state variables is required so that the observer design does not affect the closed loop control system dynamics and provides accurate state information to the MPC controller. Therefore, deadbeat design, which is rarely used for controllers, is very common for state observer design.

If all coefficients in Equation (5.20) are assigned to zero, all the eigenvalues of the observer are equal to zero and a deadbeat performance can be obtained. The deadbeat feedback observer gain K^{db} must satisfy following recursive equations:

$$\begin{aligned} k_{i+1} &= \sum_{j=0}^{i-1} r_{j+1} k_{i-j}, \; i = 1, 2, \ldots, n \\ k_1 &= 1 \end{aligned} \tag{5.21}$$

- For example, open loop stable processes with the model order n large enough to cover the whole dynamics of the process give $r_0 = 1$ and $r_2 = r_3 = \ldots = r_{n+1} = 0$. The solution of Equation (5.21) becomes

$$K^{db} = [1,\ 1,\ \cdots,\ 1]^T$$

  i.e. the basic DMC feedback option is a deadbeat observer for open loop stable processes.

- For integrating processes, $n_r = 1$, $r_0 = 2$, $r_1 = -1$, the solution of Equation (5.21) is

$$K^{db} = [1,\ 2,\ \cdots,\ n+1]^T$$

  This is the feedback design obtained by Morari *et al.*(1991).

- For general processes with open loop unstable poles or slow dynamics, a deadbeat observer can be designed by solving the simple algebraic Equation (5.21).

As a matter of fact, DMC always assumes a step type disturbance added to the process output. This assumption, together with the full step response model description, gives a simple feedback gain $K = [1,\ 1,\ \cdots,\ 1]^T$ which works very well for most control applications in the petrochemical industry. But it can not handle ramp type disturbances and has difficulties with slow disturbances.

The inclusion of the slow dynamics $D_2(q^{-1})$ in Φ gives a much better result. For example, as shown in Figure 5.4, a ramp disturbance $D_2 = 1 - q^{-1}$ gives a upward straight line for the feedback gain. The step disturbances $D_2 = 1$ results in a flat curve and a general first order dynamics in $D_2 = 1 - 0.8q^{-1}$ gives a first order damped response for the feedback gain vector.

5.5 Dynamic Tuning of the Feedback Observer

There is a potential problem in the feedback design based on the pole-placement method. The stochastic equation (5.18) has a noise term $\Delta w(k-1)$ with parameter vector $(I - KH)T$. Obviously, the feedback gain, K, would affect the way the noise is added to the state variables. For example,

- with $K = [1, 1, 1, 1]^T$, $(I - KH)T\Delta = [0, (t_1 - t_2), (t_2 - t_3), (t_3 - t_4)]^T$. The effect is reduced by the impulse response coefficient $\delta t_i = (t_i - t_{i-1})$ of the disturbance model. If the number of model coefficients is large enough, δt_i approaches zero to give a complete rejection of the disturbance.

- with $K = [1,\ 2,\ 3,\ 4]^T$, $(I - KH)T\Delta = [0,\ (2t_1 - t_2),\ (t_1 + t_2 - t_3),\ (t_1 + t_3 - t_4)]^T$. There would be a constant noise addition, $t_1 w(k-1)$, to the steady state estimation.

- with $[1,\ 1,\ 1,\ 1]^T < K < [1,\ 2,\ 3,\ 4]^T$, the noise term is added more to the first few state variables than to the steady state variable. The noise addition is slowly reduced to zero as it approaches the steady state.

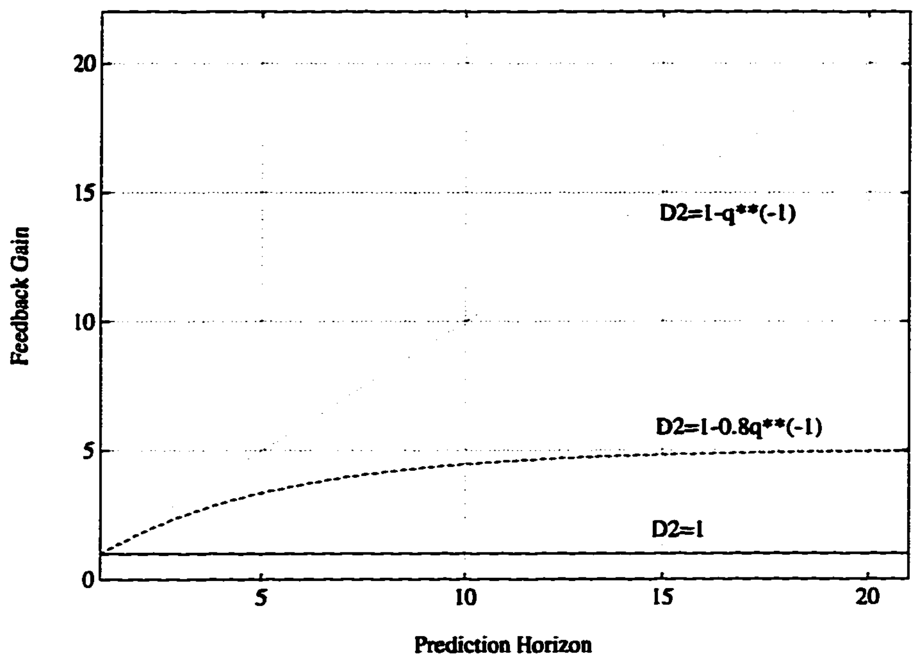

Figure 5.4:The Deadbeat State Feedback Gain Trajectory

Therefore, the feedback gain trajectory obtained from the pole-placement algorithm is usually too large to be used in practical applications. Further adjustments are required to improve the feedback performance. In addition to changing the gain arbitrarily, two formal methods can be applied. One is to use an integer 'feedback horizon' to modify the feedback gain. Another is to 'rotate' the whole gain trajectory to a reasonable region (Figure 5.5). These methods are discussed in the following subsection.

5.5.1 Feedback Horizon, N_{FB}

After obtaining the feedback observer gain, a feedback horizon, N_{FB}, can be introduced to further adjust the elements (especially the steady state elements) of the gain vector. While the effects of N_{FB} on the control performance have been illustrated extensively by Saudagar (1995), the effect of N_{FB} on the state observer is evaluated here.

With the feedback horizon N_{FB}, the gain vector is modified as:

$$K = [\underbrace{k_1,\ k_2,\ \cdots, k_{N_{FB}}}_{N_{FB}}, k_{N_{FB}}, \cdots, k_{N_{FB}}]^T$$

This modification changes the pole locations of the state observer, *i.e.* the eigenvalues of $(I - KH)\Phi$, and the properties of the noise addition to the states, *i.e.* the elements of $(I - KH)T\Delta$ as discussed previously.

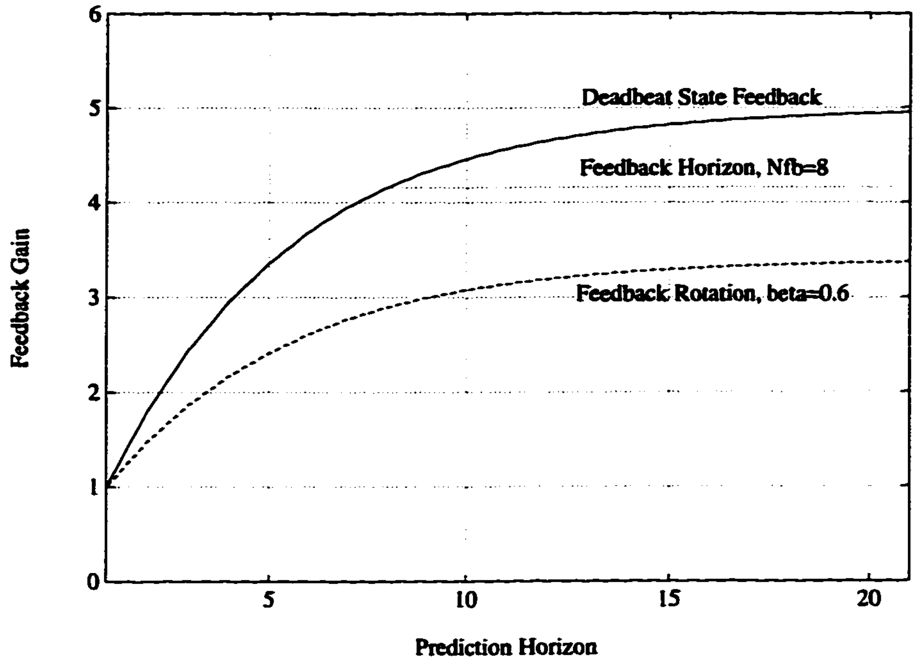

Figure 5.5:Modifications of The Feedback Gain Trajectory

Assuming a first order model for the A_2 polynomial in the dual-model formulation, and a deadbeat state feedback design in K, the effects of N_{FB} on the poles of the state observer can then be analyzed as follows.

With $A_2 = 1 + r_2 q^{-1}$, yields

$$A_2\Delta = 1 - (1 - r_2)q^{-1} - r_2 q^{-2}$$

and

$$r_1 = 1 - r_2$$

The characteristic equation of the observer with feedback horizon N_{FB} is a simplified version of the general equation in (5.20) as:

$$\begin{aligned} det[\lambda I - (I - KH)\Phi] &= \lambda^{n+1} \\ &+ (k_2 - r_1)\lambda^n \\ &+ (k_3 - r_1 k_2 - r_2)\lambda^{n-1} \\ &+ \cdots \\ &+ (k_{N_{FB}} - r_1 k_{N_{FB}-1} - r_2 k_{N_{FB}-2})\lambda^{n+2-N_{FB}} \\ &+ (k_{N_{FB}} - r_1 k_{N_{FB}} - r_2 k_{N_{FB}-1})\lambda^{n+1-N_{FB}} \\ &+ (k_{N_{FB}} - r_1 k_{N_{FB}} - r_2 k_{N_{FB}})\lambda^{n-N_{FB}} \\ &+ \cdots \\ &+ (k_{N_{FB}} - r_1 k_{N_{FB}} - r_2 k_{N_{FB}})\lambda \\ &= 0 \end{aligned} \tag{5.22}$$

Obviously, with the deadbeat observer design, the coefficients of the first $(N_{FB} - 1)$ terms, *i.e.* from λ^n to $\lambda^{n+2-N_{FB}}$, are still equal to zero. The coefficients of the last $(n - N_{FB})$ terms, *i.e.* from $\lambda^{n-N_{FB}}$ to λ, are the same as $(k_{N_{FB}} - r_1 k_{N_{FB}} - r_2 k_{N_{FB}})$. Using the relationship between r_1 and r_2, these coefficients are further simplified as:

$$\begin{aligned} & k_{N_{FB}} - r_1 k_{N_{FB}} - r_2 k_{N_{FB}} \\ = & \; k_{N_{FB}}(1 - r_1 - r_2) \\ = & \; 0 \end{aligned}$$

Therefore, the feedback horizon only changes one term in the characteristic equation, the coefficient of the $\lambda^{n+1-N_{FB}}$ term. This coefficient is simplified as:

$$\begin{aligned} & k_{N_{FB}} - r_1 k_{N_{FB}} - r_2 k_{N_{FB}-1} \\ = & \; k_{N_{FB}} - (1 - r_2) k_{N_{FB}} - r_2 k_{N_{FB}-1} \\ = & \; r_2(k_{N_{FB}} - k_{N_{FB}-1}) \end{aligned}$$

The characteristic equation in (5.22) becomes:

$$det[\lambda I - (I - KH)\Phi] = \lambda^{n+1} + r_2(k_{N_{FB}} - k_{N_{FB}-1})\lambda^{n+1-N_{FB}}$$

where as many as $(n + 1 - N_{FB})$ eigenvalues are located on the origin and N_{FB} eigenvalues are relocated on a circle with a radius of

$$\lambda = [r_2(k_{N_{FB}} - k_{N_{FB}-1})]^{1/N_{FB}}$$

REMARKS:

- Use of the feedback horizon N_{FB} relocates the eigenvalues of the state observer and hence affects the convergence speed of the state estimates;
- From the point of view of state convergence speed, N_{FB} should be selected carefully at a point where the deadbeat gain does not change too much, *i.e.* keep $\Delta k_{N_{FB}} (= k_{N_{FB}} - k_{N_{FB}-1})$ smaller;
- The dominant pole of the open-loop process r_2 also affects the feedback observer. Obviously, the feedback horizon method can not be used for open loop unstable processes because $|r_2| \geq 1$ and $\Delta k_{N_{FB}} \geq 1$ yield an unstable pole for the state observer.

As illustrated by simulations, using a feedback horizon $N_{FB} = 8$, the eigenvalues of the state observer are changed significantly from the origin to 0.8 (Figure 5.6).

Also shown by simulations, the noise addition to the first N_{FB} state variables does not change with changes in the feedback horizon (Figure 5.7). But they are completely cut-off beyond the feedback horizon. Therefore, even though it would decrease the rate of convergence of the state estimation, N_{FB} does have the advantage of limiting the noise propagation into the state variables.

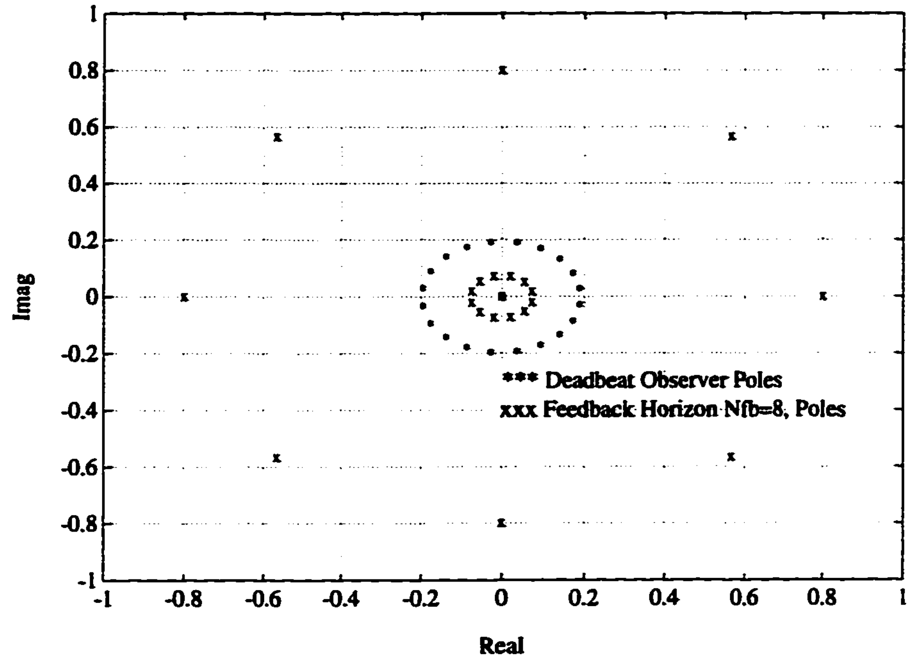

Figure 5.6:Eigenvalue Distributions of the Feedback Observer

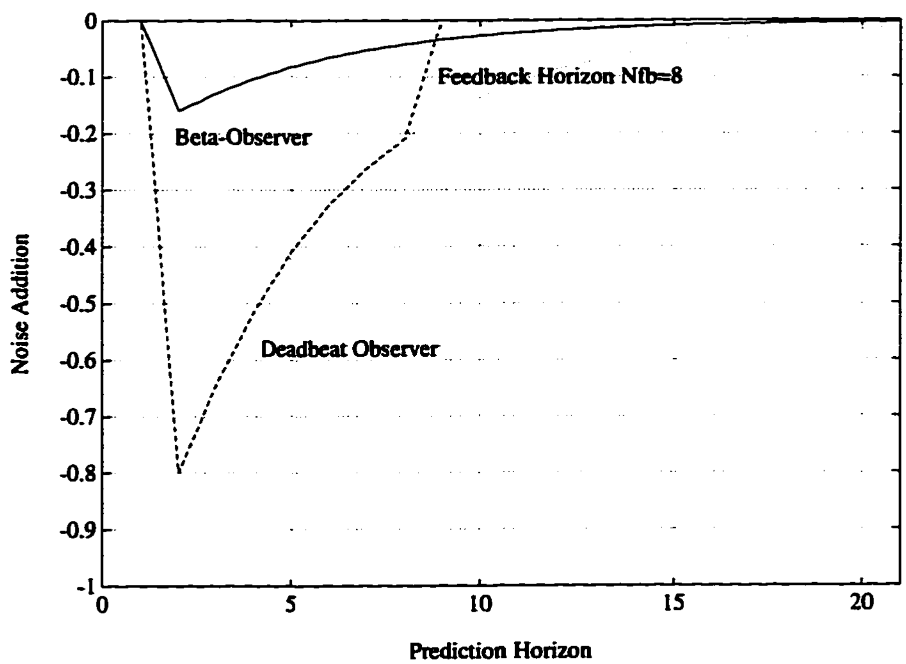

Figure 5.7:Effects of Noise on the State Estimates

5.5.2 Feedback Gain Rotation, β

State observer design based on pole placement is obviously very simple. It avoids the sophisticated Riccati equation and also does not require any assumptions about the disturbance model. Therefore, the disturbance model (if available) can still be used in the state space formulation for calculating the future output prediction. The pole locations give a clear picture on how fast the observer can achieve the true state estimates which are used in the controller design and calculation. *This information can be used to coordinate the design of the observer and controller (as opposed to treating them as two independent sequential design steps).* For example, the weighting factors on the state estimates used in the control optimization can be adjusted as a function of observer design/performance. The only drawback is that optimality of the state estimation is unknown.

Different methods for designing the feedback observer are summarized in Table 5.1.

Table 5.1:Summary of Feedback Observer Design

Method	Disturbance Model T	Gain K	Pole	Comments
Navratil, *et al* (1988)	General	Optimal Riccati solution	N/A	Large dimension Riccati equation
Lee and Morari (1991)	Step Type $[0 \cdots 0\,1]$	Optimal $f[1 \cdots 1\,1]$	1-f	Simple
Qi and Fisher (1993)	General	Deadbeat K^{db}	0	Algebraic solution
β-Observer	General	$1+\beta(K^{db}-1)$	$-(1-\beta)r_2$	Algebraic solution

A scalar parameter β, similar to the f parameter used by Lee and Morari, can be used for on-line tuning. Then the feedback gain K is given by

$$K = 1 + \beta(K^{db} - 1)$$

which yields

$$k_j = (1-\beta) + \beta k_j^{db}, \quad j = 1, 2, \ldots, n+1$$

Again, for the simple case $A_2 = 1 + r_2 q^{-1}$, it is easy to verify that the coefficients of the last $(n-1)$ terms, i.e. from λ^{n-1} to λ, of the characteristic equation (5.20) are still equal to zero. The coefficient of λ^n becomes:

$$\begin{aligned} k_2 - r_1 &= (1-\beta) + \beta k_2^{db} - r_1 \\ &= (1-\beta)(1-r_1) \\ &= (1-\beta)r_2 \end{aligned}$$

The characteristic equation is therefore simplified as:

$$\lambda^{n+1} + (1-\beta)r_2\lambda^n = 0$$

which yields n roots at the origin and one extra solution as $\lambda = -(1-\beta)r_2$. Therefore, the rotating factor, β, moves the original pole of deadbeat observer from the origin to a new place $-(1-\beta)r_2$ where r_2 is the slow mode pole of the process and/or disturbance. As shown in Figure 5.8, for $\beta = 0.2$, the dominant pole of the observer becomes $(1-0.2)*0.80 = 0.64$ as compared to zero for the ideal deadbeat observer design.

For general structures of the feedback gain K^1 and matrix Φ, the following lemmas are useful to obtain the dominant pole location of the feedback observer.

Lemma 5.1 *The dominant poles of the state observer $(I-KH)\Phi$ are equal to the open loop poles of the process if a unit element feedback gain K is used,* i.e.

$$K = [1, 1, \cdots, 1]^T$$

yields

$$det[\lambda I - (I-KH)\Phi] = A_2(\lambda^{-1})\lambda^n$$

Proof:

With $K = [1, 1, \cdots, 1]^T$, the characteristic equation of the state observer in Equation (5.20) becomes

$$\begin{aligned} det[\lambda I - (I-KH)\Phi] &= \lambda^{n+1} \\ &+ (1-r_1)\lambda^n \\ &+ (1-r_1-r_2)\lambda^{n-1} \\ &+ \cdots \\ &+ (1-r_1-r_2-\cdots-r_{n_r})\lambda^{n+2-n_r} \\ &+ (1-r_1-r_2-\cdots-r_{n_r})\lambda^{n+1-n_r} \\ &+ \cdots \\ &+ (1-r_1-r_2-\cdots-r_{n_r})\lambda \end{aligned}$$

$$
\begin{aligned}
&= \sum_{j=1}^{n+1}\lambda^j - r_1\sum_{j=1}^{n}\lambda^j - r_2\sum_{j=1}^{n-1}\lambda^j - \cdots - r_{n_r}\sum_{j=1}^{n+1-n_r}\lambda^j \\
&= (\lambda-\lambda^{n+1})/(1-\lambda) \\
&- r_1(\lambda-\lambda^{n})/(1-\lambda) \\
&- r_2(\lambda-\lambda^{n-1})/(1-\lambda) \\
&- \cdots \\
&- r_{n_r}(\lambda-\lambda^{n+1-n_r})/(1-\lambda) \\
&= (1-r_1\lambda^{-1}-r_2\lambda^{-2}-\cdots-r_{n_r}\lambda^{-n_r})\lambda^n/(1-\lambda^{-1}) \\
&+ (1-r_1-r_2-\cdots-r_{n_r})/(1-\lambda)
\end{aligned} \tag{5.23}
$$

Note that since

$$A_2(q^{-1})(1-q^{-1}) = 1 - r_1q^{-1} - r_2q^{-2} - \cdots - r_{n_r}q^{-n_r}$$

yields

$$A_2(q^{-1}) = (1 - r_1q^{-1} - r_2q^{-2} - \cdots - r_{n_r}q^{-n_r})/(1-q^{-1})$$

and

$$A_2(q^{-1})(1-q^{-1})|_{(q=1)} = 1 - r_1 - r_2 - \cdots - r_{n_r} = 0$$

The characteristic equation can then be rewritten as:

$$det[\lambda I - (I-KH)\Phi] = A_2(\lambda^{-1})\lambda^n$$

□

As discussed before, the original DMC feedback gain is exactly the same as this case. It would result in poor state convergence if used directly inside the dual-model formulation. The rotating factor β can be used to modify the observer performance so that it lies between the DMC feedback design and other feedback design options. Following Lemma describes the state convergence property.

Lemma 5.2 *Assuming $\mathcal{SR}^0$ and $\mathcal{SR}^1$ are the spectral radius of the state observer with K^0 and K^1 respectively, the spectral radius $\mathcal{SR}^\beta$ of the state observer with $K = K^0 + \beta(K^1 - K^0)$ satisfies*

$$\mathcal{SR}^\beta \le (1-\beta)\mathcal{SR}^0 + \beta\mathcal{SR}^1$$

Proof:

With $K = K^0 + \beta(K^1 - K^0)$, the state observer

$$
\begin{aligned}
\mathcal{SR}^\beta &= \| (I-KH)\Phi \|_2 \\
&= \| (1-\beta)(I-K^0H)\Phi + \beta(I-K^1H)\Phi \|_2 \\
&\le (1-\beta)\| (I-K^0H)\Phi \|_2 + \beta \| (I-K^1H)\Phi \|_2 \\
&\le (1-\beta)\mathcal{SR}^0 + \beta\mathcal{SR}^1
\end{aligned}
$$

□

Obviously, if the feedback gains $K^0 = K^{DMC}$ and $K^1 = K^{db}$, Lemma 5.2 shows that the β-observer has $\mathcal{SR}^{\beta} = (1-\beta)r_2$ (where r_2 is the dominant pole of the open loop process). This β-observer has better state convergence properties than the feedback horizon approach discussed in Section 5.5.1. It also simplifies on-line tuning and provides an optimal state estimates for some specific disturbance models.

The noise addition to the state estimates is also reduced significantly (Figure 5.7). In this way, the noise propagation can be reduced. It can also be concluded that a combination of using the β-observer and the feedback horizon N_{FB} can give a better feedback performance and eliminate the effect of noise in the long term.

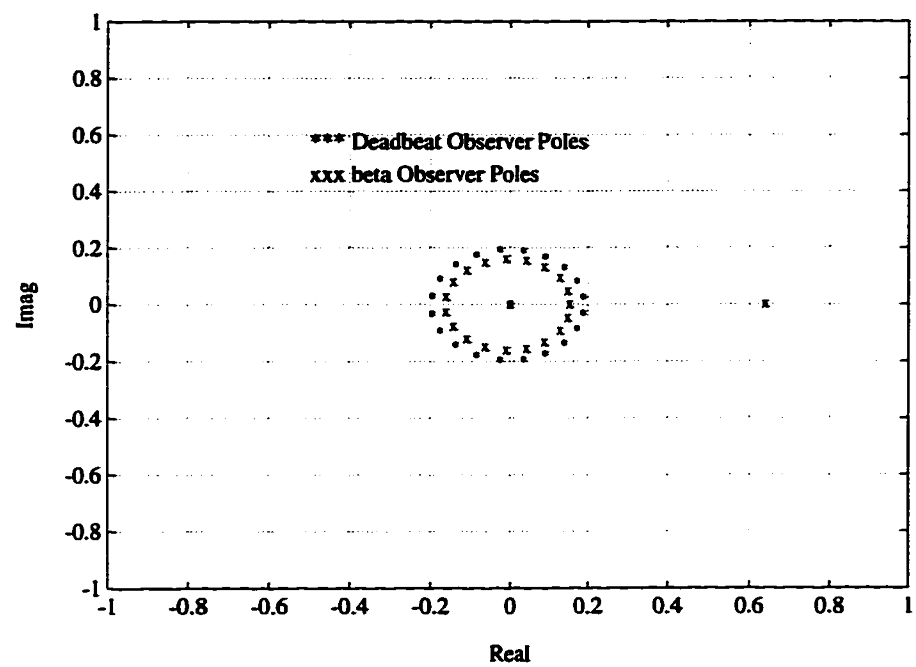

Figure 5.8: Eigenvalue Distributions of the Feedback Observer with β-Factor

The new observer design procedure with dynamic tuning consists of 5 steps:

- STEP 1: Build a dual-model state space formulation with Φ, θ, T;
- STEP 2: Design the deadbeat observer, K^{db};
- STEP 3: Design the β-observer, $K = 1 + \beta(K^{db} - 1)$ to facilitate dynamic tuning;
- STEP 4: Define a disturbance horizon, N_{FB}, to limit the addition of noise to the state vector;
- STEP 5: Tune β on-line.

In the next section, an example will be used to show the performance of β-observer.

5.5.3 Simulations

The plant used here is a process with slow disturbance dynamics.

$$Y(s) = \frac{1}{10s+1}e^{-s}U(s) + \frac{1}{20s+1}D(s)$$

With a sampling rate of 1 minute, this process has a over-damped dynamics (Figure 5.9). For this type of pseudo-ramp disturbance, Lee and Morari suggested using an integrator to replace the disturbance model in the feedback loop (Morari & Lee 1991). Four feedback designs are considered in this simulation:

$$\begin{array}{rcl} DMC: K &=& [1,\ 1,\ 1,\ 1,\ 1,\ 1,\ \ldots,\ 1]^T \\ Integrator: K &=& [1,\ 2.90,\ 5.63,\ 9.09,\ 13.22,\ 19.97,\ \ldots,\ 142.34]^T \\ Deadbeat: K &=& [1,\ 2.85,\ 5.44,\ 8.64,\ 12.35,\ 16.49,\ \ldots,\ 98.23]^T \\ \beta - Observer: K &=& [1,\ 1.37,\ 1.88,\ 2.52,\ 3.27,\ 4.10,\ \ldots,\ 20.45]^T \end{array}$$

where $\beta = 0.2$ is used.

An MPC controller with $P = 20, M = 2, \lambda = 0$ is used for this example. For a step change in the disturbance, as shown in Figures 5.10 and 5.11, DMC feedback (the dot-dashed line) takes a long time to reject the disturbance. Integrator feedback (the dotted line) over-reacts and introduces a very large 'pulse' into the input and output variables. Deadbeat feedback (the solid line) eliminates the disturbance almost immediately but requires strong control effort. β-feedback (the dashed line) is a compromise where disturbance rejection is slower than with the deadbeat design but the control action is more practical. The output performance is still very good, *i.e.* better than DMC and integrator feedback.

5.6 Conclusion

A predictive feedforward controller designed specifically for predictive control is developed to reject measurable disturbances. Feedforward horizon, together with several design methods, is put forward to improve the performance. Simulation results show much better performance over conventional methods.

Several output feedback design methods, feedback horizon and rotating factor, are developed/evaluated for better estimate convergence and noise reduction. Their effects on the state estimation convergence are discussed in detail. The dual-model state feedback design based on pole placement simplified the solution of state observer. While applicable to general disturbance dynamics, the simple parameter β can also be used to dynamically tune the controller performance on-line.

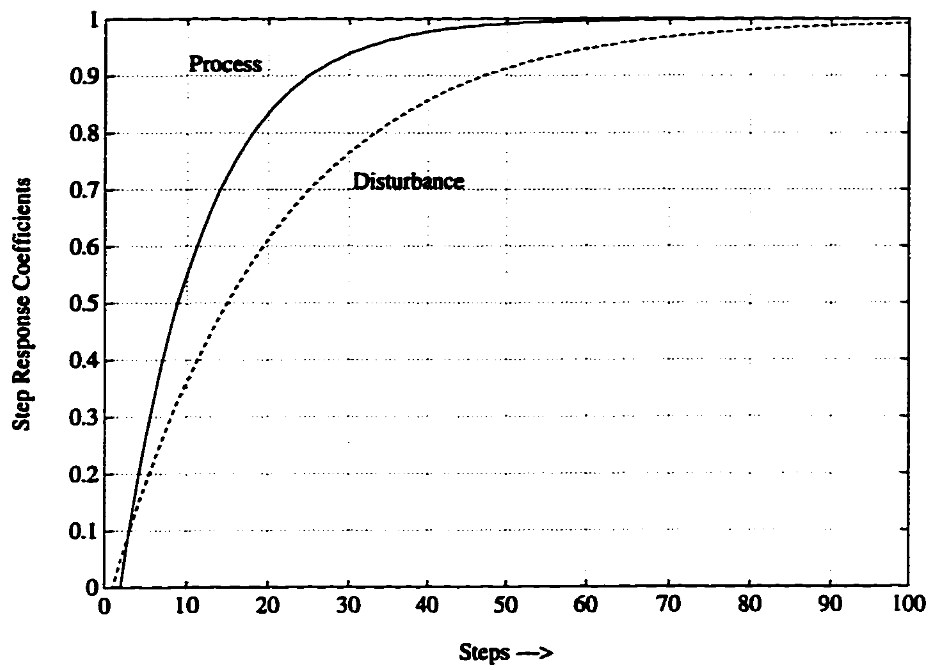

Figure 5.9:Process and Disturbance Step Responses

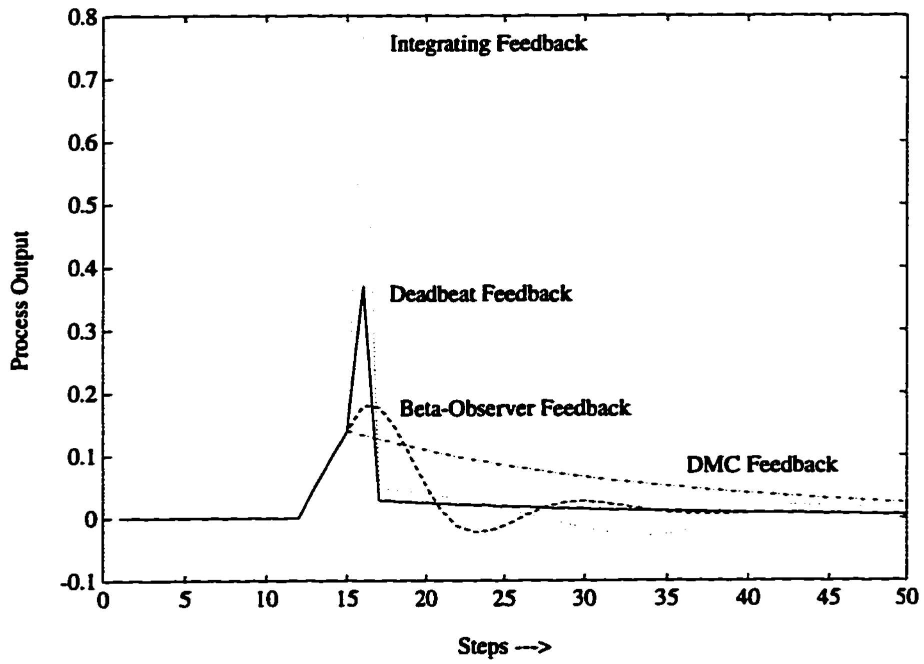

Figure 5.10:Disturbance Rejection of MPC Control

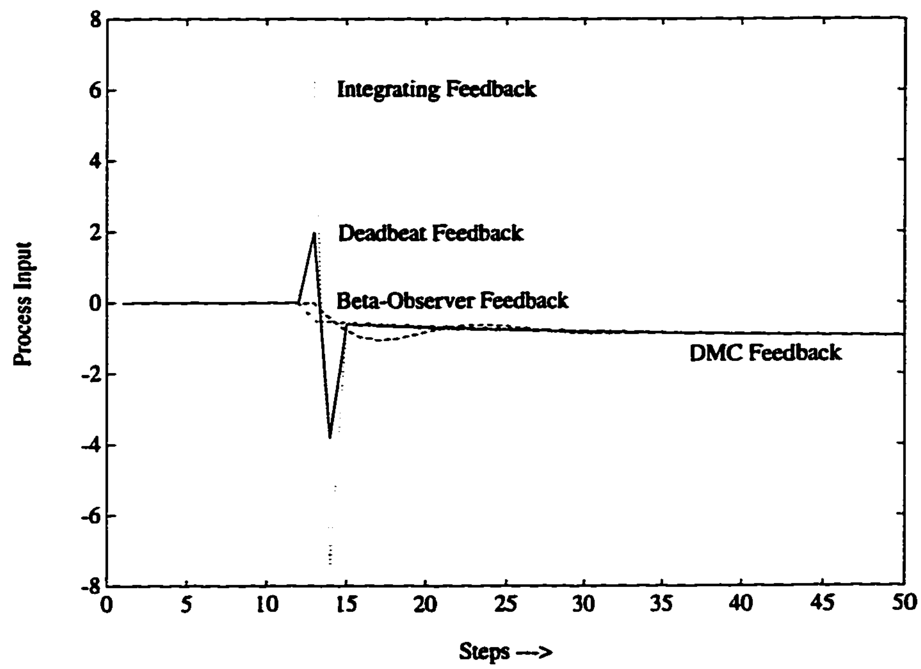

Figure 5.11:MPC Control Actions Required to Reject Disturbance

Chapter 6

Robust Stability Analysis of Unconstrained DMPC: A Parameter Perturbation Method

6.1 Introduction

Robustness analysis of a control system deals with the control performance in the presence of unexpected internal disturbances in the process. Usually, a nominal controller is designed for process under ideal working conditions. Process deviations from the ideal conditions would affect (degrade) the well designed nominal performance. Therefore, at the design stage, it is desirable to obtain some knowledge on how the control performance would change, and how to tune the nominal controller to accommodate model-plant-mismatch(MPM).

Model predictive control needs a process description in explicit form, either a parametric form such as DARMA, state space, *etc.*, or a non-parametric step (impulse) response coefficients form and its performance depends strongly on the process modelling. One of the major issues in MPC is robustness analysis. Given the realities of process identification, modelling error is unavoidable. Therefore, analysis of the performance of MPC in the presence of model error is very important. Even without theoretical support, it is a common feeling supported by numerous simulation results that predictive control is quite robust relative to other modern multivariable control techniques (Asbjornsen 1988).

After obtaining the process model, the differences between the actual process response and the model response can be evaluated. Model uncertainties can be captured in two ways:

- **Method 1: Parametric Uncertainty**

  The first method uses model parameter uncertainties to characterize the changes or errors in the process model. This form is especially suitable for processes

[0] A short version of this chapter was presented at the 1994 American Control Conference, Maryland, July 1994

with slowly-drifting dynamics. The parameter uncertainties often have explicit physical meanings. The changes in the mass of a moving vehicle or the catalyst activity degradation in the chemical reactors are typical examples of parameter changes of the process. The parameter uncertainties are usually obtained at the same time as the model parameters are estimated.

- **Method 2: Non-parametric Uncertainty**

  Another method is to use post-identification estimation. The model response is compared with the actual process response subject to the same excitation. Then, the model uncertainties are estimated from the information contained in the residual and the excitation signals. There are two choices for the uncertainty estimations, parametric or non-parametric. Obviously, if the same parametric structure as the model is applied here, the model uncertainty is exactly the same as the parameter uncertainties discussed above. Non-parametric methods which are based on spectrum analysis in the frequency domain are usually used for the general purpose of estimating the model uncertainties.

  Signal analysis is a very broad area including various FFT algorithms, power spectrums, window functions, etc. Due to its unlimited number of parameters, it can be used for the estimation of most model uncertainties, provided that the input excitation is rich enough. For SISO systems, it has been successfully applied to estimate the model uncertainties (Banerjee & Shah 1995).

  Generally speaking, the model uncertainties can be estimated very accurately using signal analysis methods. Based on this accurate information, the subsequent nominal controller can be tuned accordingly. However, there are two problems related to this model uncertainty description. One is that the model uncertainty spectrum usually does not have direct physical meaning which may make it difficult to improve the process modelling. Another problem is that for MIMO systems, there is no mature technology for model uncertainty estimation.

Model uncertainties usually degrade all aspects of MPC performance, but the major concern is their effect on the closed loop stability. Since system stability can be analyzed in both the time and the frequency domains, robustness analysis can be evaluated in both domains too. For example, the Small Gain Theorem(SGT) can be applied to examine the GPC control system robustness in the (spectral) frequency domain (Clarke 1991) and the effects of GPC tuning parameters on the robustness can be evaluated. Guidelines toward improving the robustness by using better tuned parameters have been developed (Banerjee & Shah 1992). Since both GPC and most of the commonly used MPC formulations have the same tuning parameters, these guidelines are applicable to other MPC control algorithms as well.

In the time domain, extensive research results have been developed to analyze the robustness of a control system given a state space description. Robustness criteria can be obtained corresponding to the stability criteria in either Lyapunov function form or the closed loop eigenvalue locations (Yedavalli 1985, Qiu & Davison 1986, Dickman 1987, Kolla et al. 1989, Qu & Dorsey 1990, Niu, Abreu-Garcia & Yaz 1992*b*).

Several new robustness criteria have been reported by describing model uncertainties as interval matrices (Han & Lee 1994, Keel & Bhattacharyya 1995). Since robustness criteria usually involve sufficient rather than necessary conditions, extensive research has been directed towards obtaining less conservative conditions.

Since the proposed dual-model formulation is in state space form, its robustness properties can be examined by directly applying recent research results from the robustness analysis area. Therefore, the intention of this chapter is not to develop new robust criteria but to apply existing literature results to the dual model formulation. First, it assumes the parameter uncertainty form. Then, it takes the advantage of the special structure of the dual model MPC formulation to generate explicit representations of common types of model uncertainties. This in turn reduces the conservativeness of the robust stability bounds derived using matrix perturbation theory. One advantage of handling systems in state space is that the robustness analysis can be easily extended to MIMO systems without technical problems.

6.2 Model-Plant-Mismatch in the Dual-Model Representation

MPM is characterized by parameter perturbations or uncertainty in the system matrix Φ and/or the control matrix θ of the state space model (2.13). (Note that the order of the model is determined by the prediction and control horizons specified by the designer in Equation (4.2)). In most applications, bounds on the parameter uncertainty can normally be derived a priori from process analysis or estimated based on experimental process identification tests. Once bounds on the parameter uncertainty have been established the special structure of the dual model state space formulation (2.13) can be used to get robustness bounds for MPC that are simple and less conservative than those based on general systems theory.

If the uncertainties in the system and control matrices of Equation (2.13) are represented by $\Delta\Phi$ and $\Delta\theta$ respectively then it is straightforward to show that the closed loop MPC formulation (4.6) becomes

$$\begin{bmatrix} X(k+1) \\ \overline{X}(k+1) \end{bmatrix} =$$

$$\begin{bmatrix} \Phi - \theta K_{mpc} + \Delta\Phi - \Delta\theta K_{mpc} & \theta K_{mpc} + \Delta\theta K_{mpc} \\ (I-KH)(\Delta\Phi - \Delta\theta K_{mpc}) & (I-KH)\Phi + (1-KH)\Delta\theta K_{mpc} \end{bmatrix} \begin{bmatrix} X(k) \\ \overline{X}(k) \end{bmatrix}$$

Obviously, if there is any uncertainty in the process model, the controller and observer design can not be easily separated as they were in the nominal case. If A_c is defined as the nominal closed loop matrix in Equation (4.6), then the perturbation matrix (ΔA_c) in additive form is:

$$\Delta A_c = \begin{bmatrix} \Delta\Phi - \Delta\theta K_{mpc} & \Delta\theta K_{mpc} \\ (I-KH)(\Delta\Phi - \Delta\theta K_{mpc}) & (1-KH)\Delta\theta K_{mpc} \end{bmatrix}$$

$$= \begin{bmatrix} I & I \\ (I-KH) & (I-KH) \end{bmatrix} \begin{bmatrix} \Delta\Phi - \Delta\theta K_{mpc} & 0 \\ 0 & \Delta\theta K_{mpc} \end{bmatrix} \tag{6.1}$$

The following subsections discuss four specific types of MPM that are common in process models used for MPC. It is important to note that the dual model state space formulation (2.13) assumes that the first n points of the open-loop step response are represented by the elements of θ and the remainder of the step response is represented by the parametric model defined by r-elements in the last row of Φ. For convenience, these are referred to as the "fast" and the "slow" dynamics of the process.

6.2.1 Uncertainty in the Slow Process Dynamics, Φ

MPM in the slow process dynamics can be represented by parameter perturbations in the matrix Φ. Assume that there is no MPM in the fast dynamics of the process, *i.e.* $\Delta\theta = 0$. (Note that mismatch in the process gain and time delay are discussed later). Since the elements of Φ are either 0 or 1 except for those in the last row, the matrix perturbation $\Delta\Phi$ has the following structural form (where $\Delta\phi$ represents the variation in the last row of matrix Φ) :

$$\Delta\Phi = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \Delta\phi \end{bmatrix}$$

and the closed loop matrix perturbation is

$$\Delta A_c = \begin{bmatrix} \Delta\Phi & 0 \\ (I-KH)\Delta\Phi & 0 \end{bmatrix} \tag{6.2}$$

Given the special structure of K, H and $\Delta\Phi$ in the dual-model formulation, it follows that $KH\Delta\Phi = 0$. Therefore, the system matrix perturbation can be simplified to

$$\Delta A_{\Phi} = \begin{bmatrix} \Delta\Phi & 0 \\ \Delta\Phi & 0 \end{bmatrix} \tag{6.3}$$

This special structure of $\Delta\Phi$ and ΔA_{Φ} simplifies the robustness analysis as shown in Section 6.3.

6.2.2 Uncertainty in the Fast Process Dynamics, θ

Let $\Delta\Phi = 0$ in Equation (6.1), and assume that the uncertainty in $\Delta\theta$ is described as variations of the step response coefficients, $\delta S_i = \bar{S}_i - S_i$, *i.e.*

$$\Delta\theta^T = [\delta S_1 \; \delta S_2 \; \cdots \; \delta S_{n+1}]$$

Then the system matrix perturbation becomes

$$\Delta A_\theta = \begin{bmatrix} -I & I \\ -(I-KH) & (I-KH) \end{bmatrix} \begin{bmatrix} \Delta\theta K_{mpc} & 0 \\ 0 & \Delta\theta K_{mpc} \end{bmatrix} \tag{6.4}$$

Gain mismatch and time-delay mismatch which are special cases of $\Delta\theta$ are discussed below.

6.2.3 Gain Mismatch

The gain mismatch can be defined as Δk and satisfies

$$\bar{S}_i = (1+\Delta k)S_i \quad i = 1, 2, \ldots, n+1$$

where Δk is a gain difference ratio, *i.e.*

$$\Delta k = \frac{K_a - K_n}{K_n} = \frac{\text{Actual Gain - Nominal Gain}}{\text{Nominal Gain}}$$

The fast dynamics perturbation vector $\Delta\theta$ is

$$\Delta\theta = \Delta k\theta$$

and the closed loop system perturbation matrix is

$$\Delta A_k = \Delta k \begin{bmatrix} -\theta K_{mpc} & \theta K_{mpc} \\ -(I-KH)\theta K_{mpc} & (1-KH)\theta K_{mpc} \end{bmatrix} \tag{6.5}$$

The ΔA_k matrix is completely known except for the gain uncertainty Δk.

6.2.4 Time Delay Mismatch

Process time-delay mismatch is defined by assuming that the nominal time-delay d is the minimal one. Time-delay mismatch for the MPC formulation is then represented by

$$\Delta\theta = \Phi_s C_d$$

where

$$\Phi_s = \begin{bmatrix} S_1 & 0 & 0 & 0 & \cdots & 0 \\ S_2 & S_1 & 0 & 0 & \cdots & 0 \\ S_3 & S_2 & S_1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ S_{d_m} & S_{d_m-1} & S_{d_m-2} & S_{d_m-3} & \cdots & S_1 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ S_{n+1} & S_n & S_{n-1} & S_{n-2} & \cdots & S_{n-d_m+1} \end{bmatrix}_{(n+1)\times(d_m+1)}$$

$$C_d^T = \left[\overbrace{1\,0\,\cdots\,0\,-1}^{\Delta d+1}\,0\,\cdots\,0 \right]_{1\times(d_m+1)}$$

and d_m is the maximum possible process time-delay.

Unfortunately, there is no further simplification. The time-delay uncertainty can be considered as a special form of Equation (6.4).

6.3 Robustness Analysis Using Matrix Perturbation Theory

There is a large body of literature dealing with the robust-stability analysis of linear time invariant (LTI) systems. In 1977, a general methodology was proposed to calculate explicit bounds on model uncertainties using time domain analysis (Patel, Todd & Sridhar 1977). Improved results have been developed by taking advantage of the known structure of particular linear perturbations (Yedavalli 1985). Recent publications have focused on developing alternative criteria or reducing the degree of conservativeness in the sufficient (but not necessary) conditions for robust stability.

However, generally speaking, there are two main approaches used for determining robust stability criteria, time domain analysis based on the solution of Lyapunov matrix equations or frequency domain techniques. The Lyapunov approach is difficult if the system matrix is of high dimension or is poorly structured. Therefore, this chapter is based on the frequency domain approach.

For special cases involving only one variable parameter, *e.g.* gain uncertainty in a SISO system, root locus techniques can be used to get both necessary and sufficient criteria.

Juang *et al.* developed robustness criteria by analyzing the eigenvalues of continuous LTI systems in the presence of MPM (Juang et al. 1986). The following section extends these results to discrete systems and applies them to MPC.

6.3.1 Formulation

Consider the multivariable linear discrete closed-loop system

$$X_{k+1} = A_c X_k \tag{6.6}$$

When there is parameter uncertainty (MPM) in A_c, the system is represented by

$$X_{k+1} = \tilde{A}_c X_k \tag{6.7}$$

Two widely used definitions of $\tilde{A}$ are:

- Additive perturbation

$$\tilde{A}_c = (A_c + \Delta A_c)$$

- Multiplicative perturbation

$$\tilde{A}_c = (I + \Delta A_c)A_c$$

The additive perturbation model is more convenient for MPC type formulations and is the only one considered in this paper. The following types of perturbation are defined based on how much information is known about the perturbations (*i.e. model uncertainties*):

1. **Highly structured perturbation:** The perturbation model structure and bounds on each element of the perturbation matrix are known.
2. **Weakly structured perturbation:** The perturbation model structure is known, but only a spectral norm bound on the perturbation matrix is known.
3. **Unstructured perturbation:** Perturbation model structure is unknown.

The study of robust stability usually assumes that the nominal system is stable, *i.e.* all eigenvalues of A_c are within the unit circle. Then, the system in (6.7) is said to be robust if under perturbations in the elements of the actual system ($\tilde{A}_c = A_c + \Delta A_c$), the eigenvalues of $\tilde{A}_c$ are still within the unit circle.

6.3.2 Stability Criteria

Theorem 6.1 *The perturbed system in (6.7) is asymptotically stable if* $(A_c(\epsilon) - e^{j\omega}I)$ *is invertible for all* $\omega \geq 0$ *and* $\epsilon \in [0, 1]$. i.e.

$$det(A_c(\epsilon) - e^{j\omega}I) \neq 0 \qquad (6.8)$$

where $A_c(\epsilon)$ *is defined as a continuous matrix function of* ϵ, *and satisfies*

$$A_c(0) = A_c, \quad A_c(1) = A_c + \Delta A_c$$

Proof:

$A_c(\epsilon)$ represents the matrix variation from A_c to $A_c + \Delta A_c$, as ϵ varies continuously from 0 to 1. If there is a eigenvalue of $A_c + \Delta A_c$ outside the unit circle, there must exist at least one $\epsilon \in [0, 1]$ such that $A_c(\epsilon)$ has eigenvalues on the unit circle. The eigenvalue on the unit circle $\lambda_i = e^{j\omega_i}$ would make the matrix $[A_c(\epsilon) - e^{j\omega}I)]$ become singular at certain ϵ, ω_i, *i.e.*

$$det(A_c(\epsilon) - e^{j\omega_i}I) = 0$$

This contradicts the condition in Equation (6.8). So, the result is proven. □

A similar result has been given for continuous system (Juang et al. 1986).

As one special case of $A_c(\epsilon)$, the actual system matrix $A_c + \Delta A_c$ also satisfies the equation in (6.8) and leads to following theorem.

Theorem 6.2 *The perturbed system in (6.7) is asymptotically stable if*

1. *The nominal system in (6.6) is asymptotically stable, and*
2. $\| \Delta A_c(\omega) \| \cdot \| M(\omega) \| < 1 \qquad \omega \geq 0$
   where $M(\omega) \equiv (A_c - e^{j\omega} I)^{-1}$

Proof: Since A_c is stable, $A_c - e^{j\omega}I$ is always invertible. The perturbed system matrix is expandable as:

$$A_c + \Delta A_c - e^{jw}I = (A_c - e^{jw}I)[I + (A_c - e^{jw}I)^{-1}\Delta A_c]$$

where one of the two terms, $A_c - e^{j\omega}I$ is always invertible. According to Theorem 6.1, the invertibility of $(A_c + \Delta A_c - e^{j\epsilon}I)$ must be ensured. It is *sufficient* to require that

$$\begin{aligned} \|(A_c - e^{jw}I)^{-1}\Delta A_c\| &\leq \|(A_c - e^{jw}I)^{-1}\| \|\Delta A_c\| \\ &< 1 \end{aligned}$$

which gives

$$\|\Delta A_c(\omega)\| < \frac{1}{\|M(\omega)\|}$$

□

The following corollary applies to the case where the error matrix can be written as the product of two matrices, an unknown part $E(\omega)$ and a known part $B(\omega)$,

$$\Delta A_c(\omega) = E(\omega)B(\omega)$$

Corollary 6.1 *The perturbed system in (6.7) with* $\Delta A_c(\omega) \equiv E(\omega)B(\omega)$ *is asymptotically stable if*

1. *The nominal system in (6.6) is asymptotically stable, and*
2. $\| E(\omega) \| \cdot \| B(\omega)M(\omega) \| < 1 \qquad \omega \geq 0$
   where $M(\omega) \equiv (A_c - e^{j\omega} I)^{-1}$

The proof is quite straightforward.

REMARKS:

1. These theorems and the corollary are only sufficient conditions for system stability. The norm inequality

   $$\|E \times B\| \leq \|E\| \, \|B\|$$

   introduces extra conservativeness into the robustness bound. Therefore, it is desirable to extract known information in matrix B as much as possible. Then, the known part of the B-matrix can be considered together with $M(\omega)$ to reduce the conservativeness. The special structure of the dual-model description helps obtain this objective as shown in a later section.

2. For weakly structured perturbations, this theorem is used to obtain allowable bounds for the error matrix since only the norm information is required for the model error.

3. Any matrix norm can be used.

4. For highly structured perturbations, the magnitude bounds on each element of the error matrix can be defined. The error matrix can therefore be decomposed accordingly. The following corollary will give less conservative results than Corollary (6.1).

Corollary 6.2 *The perturbed system in (6.7) with* $| E(\omega) | \leq \mu(\omega)\Pi(\omega)$ *is asymptotically stable if*

1. *The nominal system in (6.6) is asymptotically stable.*

2. $\mu(\omega) < 1/ \| \Pi(\omega)B(\omega)M(\omega) \| \qquad \omega \geq 0$

   where $\Pi(\omega)$ *is a matrix consisting of the absolute uncertainty bound of each element in* $E(\omega)$ *while* $\mu(\omega)$ *is a scalar.*

A typical example is the gain uncertainty in the dual-model description where the gain mismatch term can be extracted from the rest of the matrices.

6.3.3 Robust Stability Bounds for Dual-Model MPC

The main objective of this section is to determine robust stability bounds, $\mathcal{RB}$, (*i.e.* the degree of MPM that can be accommodated without causing instability) for MPC systems in the presence of MPM, *i.e.*

$$\|E(\omega)\| \leq \mathcal{RB}$$

where the robustness bound $\mathcal{RB}$ includes the nominal process model parameters, tuning parameters, etc. Even without the knowledge of modelling error $E(\omega)$, the nominal controller can be designed to improve the ability of the closed loop control system to handle unexpected process variations. This can be done using the perturbation theorems and corollaries presented in the previous section as:

$$\mathcal{RB} = \frac{1}{\|B(\omega)(A_c - e^{j\omega}I)\|}$$

The four specific types of MPM discussed in Section 6.2 are:

- Errors in the model of the slow process dynamics, $\Delta\Phi$.
- Errors in the model of the fast process dynamics, $\Delta\theta$.
- Gain mismatch, Δk.

Table 6.1:Error Matrix of MPC Uncertainty

	$E(\omega)$	$B(\omega)$
General	$\begin{bmatrix} \Delta\Phi - \Delta\theta K_{mpc} & 0 \\ 0 & \Delta\theta K_{mpc} \end{bmatrix}$	$\begin{bmatrix} I & I \\ (I-KH) & (I-KH) \end{bmatrix}$
$\Delta\Phi$	$\begin{bmatrix} \Delta\Phi & 0 \\ \Delta\Phi & 0 \end{bmatrix}$	I
$\Delta\theta$	$\begin{bmatrix} -\Delta\theta K_{mpc} & 0 \\ 0 & \Delta\theta K_{mpc} \end{bmatrix}$	$\begin{bmatrix} I & I \\ (I-KH) & (I-KH) \end{bmatrix}$
Δk	Δk	$\begin{bmatrix} -\theta K_{mpc} & \theta K_{mpc} \\ -(I-KH)\theta K_{mpc} & (I-KH)\theta K_{mpc} \end{bmatrix}$
Δd	$\begin{bmatrix} -\Phi_s C_d K_{mpc} & 0 \\ 0 & \Phi_s C_d K_{mpc} \end{bmatrix}$	$\begin{bmatrix} I & I \\ (I-KH) & (I-KH) \end{bmatrix}$

- Process time-delay mismatch, Δd.

Table 6.1 contains the $B(\omega), E(\omega)$ matrices required to apply the perturbation theorems to calculate the $\mathcal{RB}$ and to evaluate the system stability in the presence of each of the four different types of mismatch. The perturbation theorems and matrix definitions in Table 6.1 are very general in the sense that they can be used with any matrix norm. However, if a specific norm is used the robustness analysis can be further simplified, especially when the special structure of the dual-model state space formulation is used.

Table 6.2 shows the stability requirements using the specific forms of $E(\omega)$ and $B(\omega)$ that apply to DMPC, and the 2-norms of the $E(\omega)$ matrix(*e.g.* the square root of the maximum eigenvalue of $E^T E$). Examination of Table 6.2 leads to the general conclusion (for all four types of MPM defined in section 6.2.1 to 6.2.4) that:

1. the robustness improves as the norm $\| M(\omega) \|$ decreases;

2. the robustness bound increases as the MPC is "detuned", *i.e.* smaller feedback gain K_{mpc}.

For example, as the norm of the controller gain $\| K_{mpc} \|$ decreases, the robustness bounds increase. The controller gain, K_{mpc}, is a function of MPC tuning parameters P, M and λ (see Equation (4.4)) . Hence, MPC can be tuned to provide the degree of robustness required for a given application.

Note however that:

- Increased robustness usually means decreased performance due to lower state feedback gain, K_{mpc}.

- The matrix inequalities that define robustness are generally conservative.

Table 6.2:2-Norm of Error Matrix and MPC Robustness Bound

	$\parallel E(\omega) \parallel$	$\mathcal{RB}$	Design Objective Minimize
Φ Uncertainty	$\parallel \Delta\phi \parallel$	$\frac{1}{\sqrt{2}\parallel M(\omega)\parallel}$	$\parallel M(\omega) \parallel$
θ Uncertainty	Θ	$\frac{1}{\parallel M(\omega)B(\omega)\parallel \parallel K_{mpc}\parallel}$	$\parallel M(\omega)B(\omega) \parallel, \parallel K_{mpc} \parallel$
k Uncertainty	Δk	$\frac{1}{\parallel M(\omega)B(\omega)\parallel}$	$\parallel M(\omega)B(\omega) \parallel$
d Uncertainty	Θ	$\frac{1}{\parallel M(\omega)B(\omega)\parallel \parallel K_{mpc}\parallel}$	$\parallel M(\omega)B(\omega) \parallel, \parallel K_{mpc} \parallel$

$\Theta = \sum_{i=1}^{n}(\delta S_i)^2$

6.3.4 Robustness Analysis: Simulation Results

The effect of the MPC tuning parameters λ, P and M on the robustness bounds of the process $1/(s+1), T_s = 0.3$, in the presence of different types of MPM, is calculated based on the definitions in Table 6.2.

The Nyquist frequency is one half of the sampling frequency, *i.e.* $f_N = f_s/2 = 1.667(rad/sec)$, which is used to normalize the real frequency in these figures. Therefore, in the normalized frequency range (0 to 1.0), the critical frequency of the process $f_c = 1(rad/sec)$ becomes $f_c/f_N = 0.6$.

1. **Control Weighting λ**

   With fixed values of P and M, $P = 8, M = 1$, the effects of the control weighting, λ, on the robustness bound are shown in Figure 6.1. The effect of λ on the state feedback gain K_{mpc} is shown in Figure 6.2.

   The effects can be summarized as:

   - More knowledge about the type of uncertainty and tighter bounds on the uncertainty result in increased robustness bounds. For example, the robustness bound for the *gain* uncertainty is much larger than the bound for the *general* case, *i.e.* 0.22 vs. 0.012.
   - Increasing the control weighting, λ, improves robustness bounds in the high frequency range more than those in the low frequency range. In other words, if the modelling error is a high frequency component (*e.g.* time-delay mismatches), then, the MPC controller is able to overcome the MPM by proper control weighting λ. However, on the other hand, if the model error happens at the low frequency range (*e.g.* gain and slow mode poles), the effect of λ is quite limited.
   - Increasing the control weighting λ decreases the feedback gain (performance) of the system(Figure 6.2).

Parameter Bounds: It is interesting to see how the frequency domain robustness bound can be used to calculate the parameter bounds for the process model

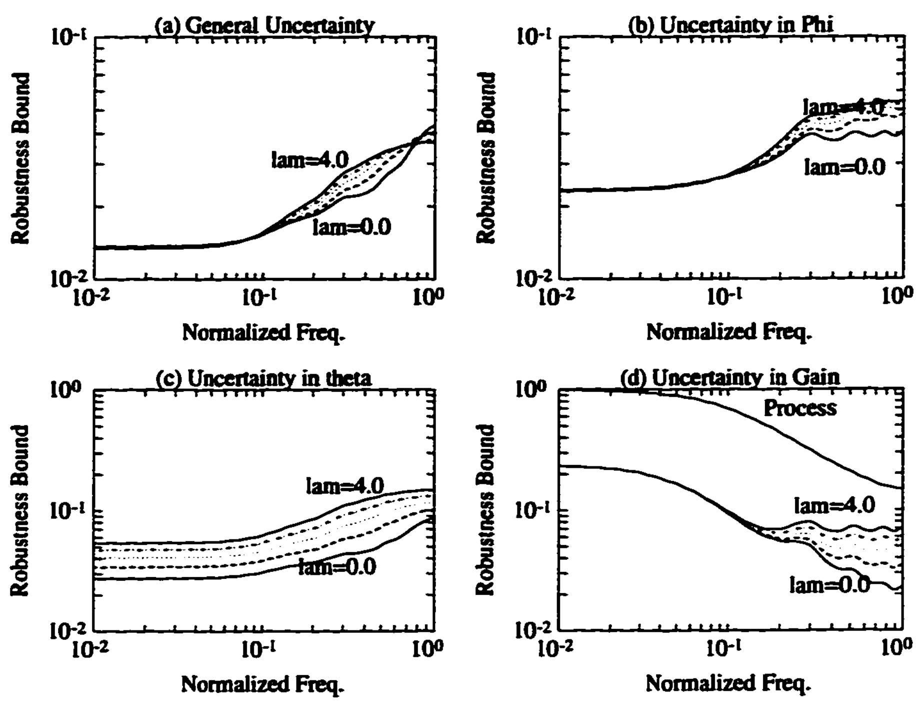

Figure 6.1: The Effect of Control Weighting λ in MPC on the Robustness Bounds

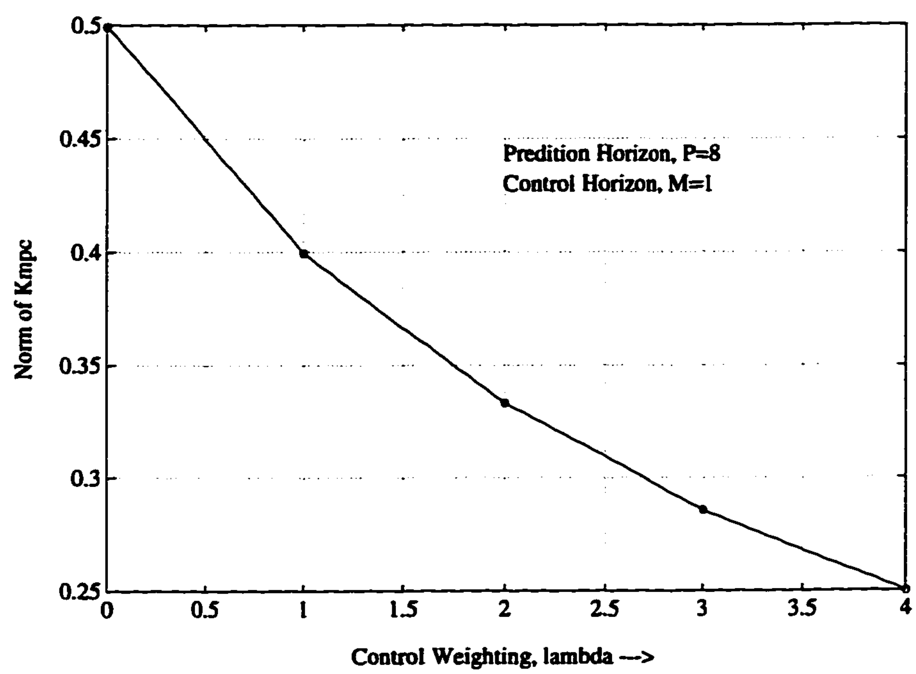

Figure 6.2: The Effect of Control Weighting λ in MPC on the Controller Gain

coefficients. As an interpretation of the robustness bound, for example, $\lambda = 1.0$ gives a minimum bound of 0.033 (Figure 6.1.c) for the $\Delta\theta$ case. According to the stability condition in Table 6.2 that,

$$\Theta = \sum_{i=1}^{10} (\delta S_i)^2 \leq 0.033$$

Assuming a maximum relative error in S_i as δs, then

$$\begin{aligned} \Theta &= \sum_{i=1}^{10} (\delta S_i)^2 \\ &= \sum_{i=1}^{10} [S_i(\frac{\bar{S}_i - S_i}{S_i})]^2 \\ &\leq (\delta s)^2 \sum_{i=1}^{10} S_i^2 \end{aligned}$$

For this example, the process model gives $\sum_{i=1}^{10} S_i^2 = 5.7814$. So that

$$5.7814 \times (\delta s)^2 \leq 0.033$$

which gives

$$\delta s \leq 7.56\%$$

That gives a quantitative result showing that the actual process step response coefficients can be allowed to vary in a neighbourhood of $\pm 7.56\%$, on average, about the real process coefficients without affecting the closed loop stability. On the other hand, if all other coefficients are exact, the one uncertain coefficient, say $\bar{S}_{10}$ is allowed to vary by δs from

$$S_{10}^2 \times (\delta s)^2 = 0.9029 \times (\delta s)^2 \leq 0.033$$

which gives $\delta s \leq 19.12\%$.

Now, consider another example for the $\Delta\Phi$-uncertainty case (Figure 6.1.b). The absolute value of the robustness bound appears smaller than in the previous example (0.023 *vs.* 0.033). But the robust stability condition for this case is

$$||\Delta\phi|| \leq \mathcal{RB}$$

For this simple example, $n_r = 1$ and $||\Delta\phi|| = 2(\Delta r)^2$ gives $|\Delta r| \leq 0.1073$. Therefore, the actual parameter of the slow mode AR model could vary in the range of (0.7408 ± 0.1073), *i.e.* 14.48%.

2. The Output Prediction Horizon P

With $\lambda = 0$ and $M = 1$, the effects of the prediction horizon, P, on the robustness bound and controller gain are very similar to that of the control weighting (Figure 6.3 and Figure 6.4). The low frequency part of the robustness bound can be increased by increasing the prediction horizon. For the case of gain uncertainty (Figure 6.3d), the difference between adjusting P or λ is quite obvious. It appears that the effect of the prediction horizon is very limited while control weighting can significantly increase the high frequency part of the robustness bound. One explanation could be that the gain mismatch has an influence over the whole prediction horizon. At the same time of increasing the prediction horizon, more gain mismatches are introduced via the process coefficients which counter affect the result of prediction horizon. Therefore, the prediction horizon is not a good tuning parameter for the gain uncertainty situations. But it is good for $\Delta\theta$ cases.

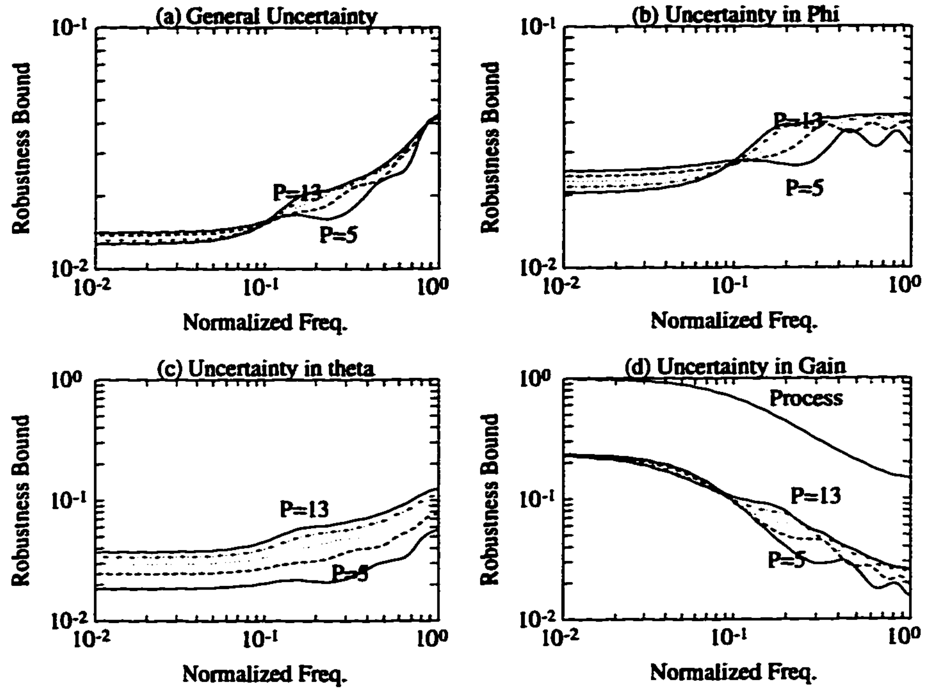

Figure 6.3: The Effect of MPC Prediction Horizon P on the Robustness Bounds

3. **The Input Control Horizon M**

   The control horizon, on the other hand, has a dramatic different effect on both the robustness bounds (Figure 6.5) and the controller gain (Figure 6.6) where $\lambda = 0$ and $P = 8$ are used. $M = 1$ is the best choice for all types of mismatch at almost every frequency in the sense that the robustness bounds are larger, *i.e.* that larger MPM can be tolerated without instability. These results also show that the robustness bound and the controller gain are very sensitive to changes in the control horizon. However, being integer numbers only, the control horizon can not be used to tune the controller smoothly. A new technology, the fractional control horizon, will be discussed in Chapter 8 to improve the smoothness of the tuning.

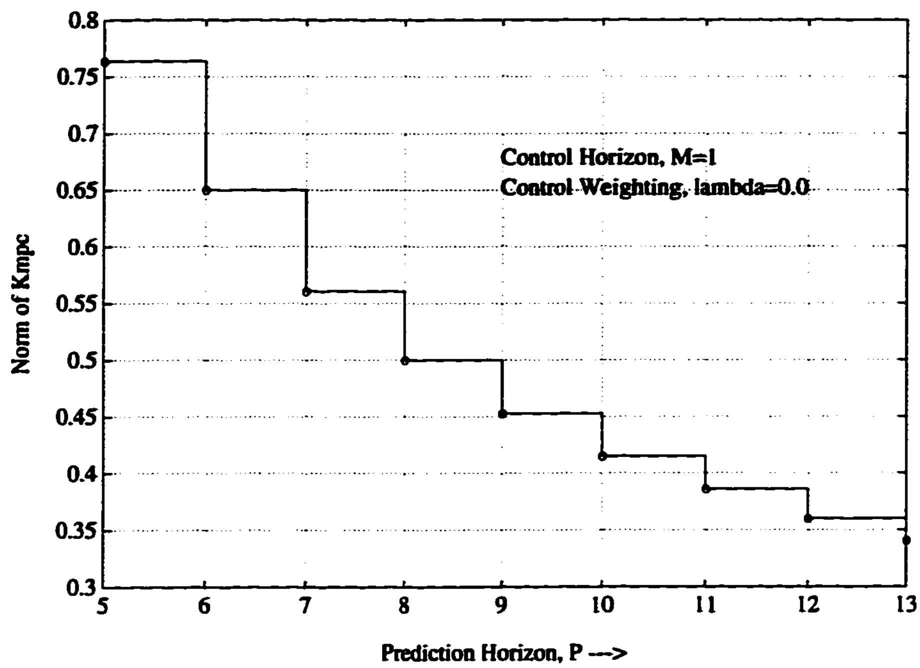

Figure 6.4: The Effect of Prediction Horizon P in MPC on the Controller Gain

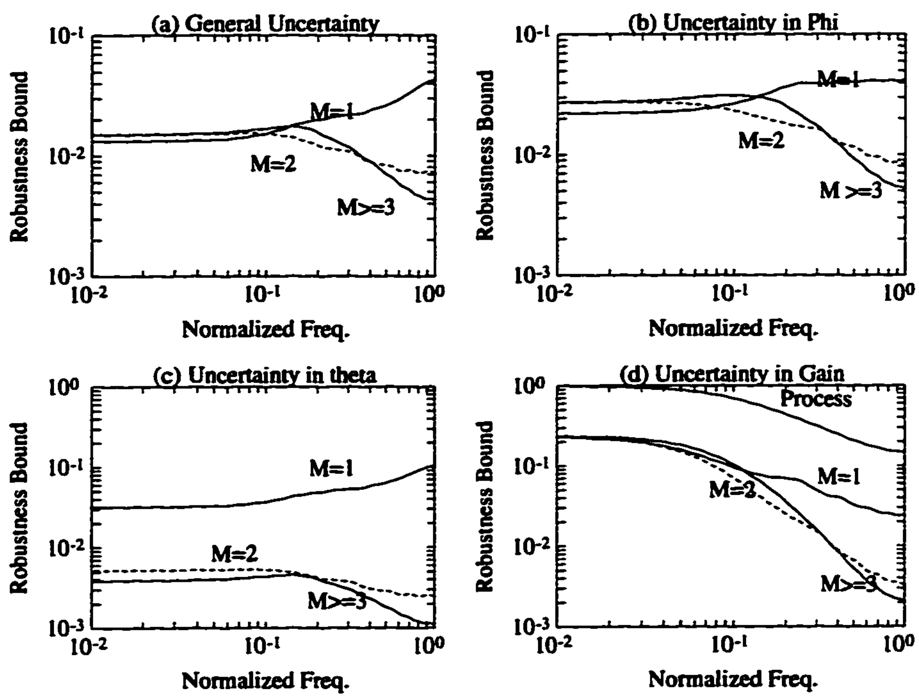

Figure 6.5: The Effect of MPC Control Horizon M on the Robustness Bounds

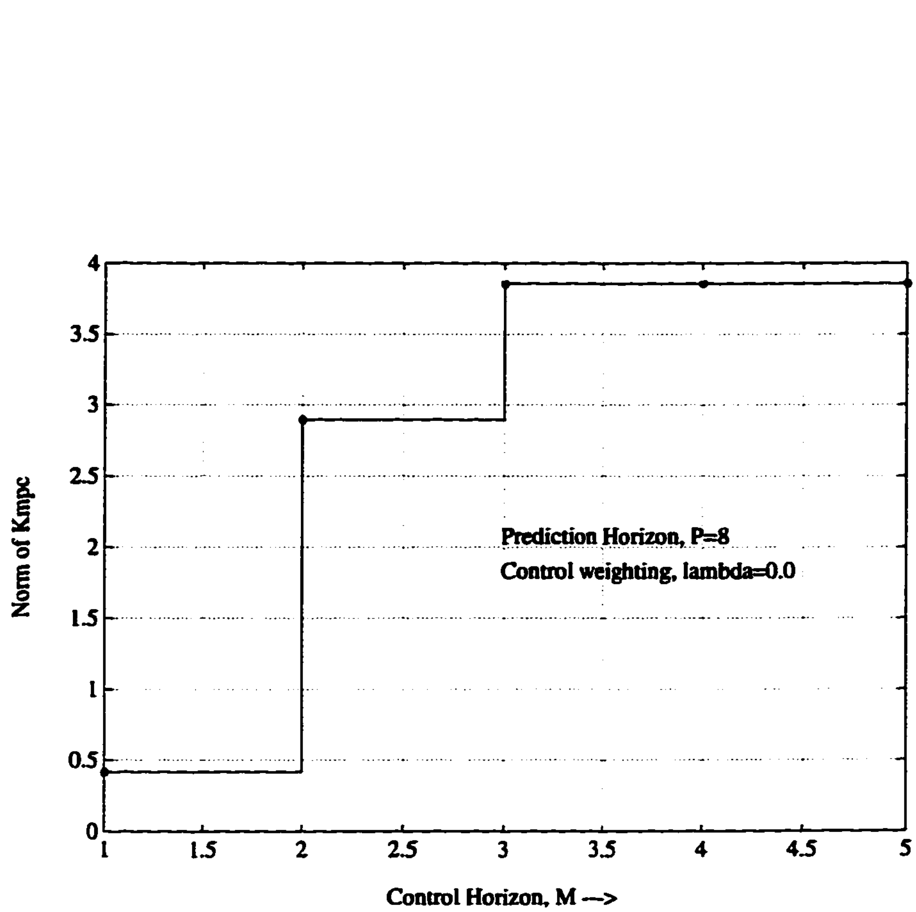

Figure 6.6: The Effect of Control Horizon M in MPC on the Controller Gain

4. **All of the results can be summarized by saying that, in general,** *increased robustness is obtained at the cost of poorer closed-loop performance.*

6.3.5 Sensitivity Analysis of The Robustness Bound

The robustness bound is a nonlinear, discontinuous function of all the tuning parameters and the process dynamics, *i.e.*

$$\mathcal{RB} = \mathcal{F}(\lambda, P, M)$$

The tuning parameters have different magnitudes and a different relative effect on robustness. However, there is no obvious way to generate relative or normalized effects. Hence, their effects are compared using the same controller gain which implies similar process performance. Then, the sensitivity of the robustness bound to tuning parameters can be analyzed as the partial derivative:

$$\mathcal{RB}_{sv} = \left[\frac{\partial(\mathcal{RB})}{\partial\zeta}\right]_{\|K_{mpc}\|=c}$$

where $\zeta = \lambda, M, or P$. Note that numerical approximations must be used to calculate the continuous derivative shown above.

Table 6.3 shows the sensitivity analysis result when $\|K_{mpc}\|$ is in the neighbourhood of 0.45 to 0.50. When a tuning parameter changes, the corresponding robustness

Table 6.3:Sensitivity Analysis of the Robustness Bound

		$f_0 \times 10^{-3}$	$f_m \times 10^{-3}$	$f_c \times 10^{-3}$
	λ	0	0.21	2.9
$\Delta\Phi$	P	0.6	1.6	0.05
	M	0.11	3.0	0.64
	λ	3.4	4.9	12.0
$\Delta\theta$	P	2.6	5.0	6.5
	M	0.54	0.83	1.4
	λ	0	0.65	4.7
Δk	P	0.44	5.0	0.15
	M	0	0.81	0.50
	λ	0	0.15	2.0
General	P	0.18	0.55	0.20
	M	0.04	0.12	0.45

bound changes as well. Its frequency components are represented by the low frequency part f_0, the medium frequency part $f_m = 0.17$ and the high frequency part $f_c = 0.6$ (which is the critical frequency of the overdamped process).

For each specific type of model uncertainties in the dual-model, the effects of tuning parameters can be summarized as:

1. For model uncertainties in the slow dynamics, *i.e.* $\Delta\Phi$, a combination of increasing the prediction horizon P for the low frequencies and increasing the control weighting λ for the medium and high frequencies could be used to improve the robustness.

2. For model uncertainties in the fast dynamics, *i.e.* $\Delta\theta$, all tuning parameters can be used to increase the robustness bounds although the control weighting λ has the greatest effect for this type of model uncertainty.

3. In case of process gain uncertainty, the control weighting λ increases the robustness at high frequencies but the prediction horizon, P, is the only tuning parameter that affects the robustness bounds at low frequencies.

4. For the general case, control weighting, λ, has the largest effect on robustness at high frequencies but the prediction horizon, P, has a stronger effect at medium and low frequencies. The effect of λ on the control performance (*e.g.* controller gain) is almost linear (Figure 6.2). Being a continuous adjustable parameter, it is the most commonly used on-line tuning parameter for MPC control systems.

   The control horizon M strongly affect both robustness and performance but in its original integer form, it gives very coarse tuning adjustments. The results also show that even with the same cost of control performance, the control horizon can not do a better job than P and λ.

Robustness analysis based on perturbation theory (*e.g.* Corollary 6.1 and 6.2) provides a direct and very general technique for analyzing the robust stability limits of MPC. However, the results can sometimes be very conservative. Tighter robustness bounds have been derived using:

- the dual-model formulation (2.13);
- specific types of MPM (Table 6.1);
- 2-norms of B, E and M matrices (Table 6.2).

However, the robustness results are still simply ***sufficient*** rather than ***necessary***. The next section looks at ***necessary and sufficient*** conditions for robust stability for some very simple, specific examples.

6.4 Robustness Analysis Using Root Loci

For SISO processes, the robust performance of the closed loop system can be easily analyzed using the root locus approach if there is only one unknown parameter. For gain mismatch or time-delay mismatch in MPC applications, the root locus approach can be successfully applied to calculate the maximum allowable gain or time-delay mismatch, *i.e.* the Δk or Δd that moves the eigenvalues of the system to the unit circle. The robustness bounds are necessary as well as sufficient and can be used

1. To evaluate the conservativeness of robustness bounds obtained from perturbation theorems.
2. To evaluate the effects of tuning parameters when model uncertainties are present.

The discussion in the preceding section and in previous studies based on the Small-Gain Theorem (Banerjee & Shah 1992) suggests that the slower the controller (*i.e.* large P, large λ, small M) the more robust the closed loop system. The following example shows this is not always true!

6.4.1 Effects of MPC Tuning Parameters on Gain Bounds

For a given process, the maximum allowable gain mismatch can be found by continuously increasing the gain mismatch term Δk until the stability limit is reached. By repeating the same procedure, the gain bounds K_m can be obtained for different combinations of controller parameters.

1. **Example: Root Locus Analysis**

   The effects of the prediction horizon P and control weighting λ are shown in Figure 6.7 where the process is $1/(s+1)$, $T_s = 0.3$, $M = 1$. Figure 6.8 shows the effects of the control horizon M and control weighting λ for the same process with fixed $P = 9$. Careful analysis of the results from several root locus plots leads to the following conclusions:

- Robustness increases as λ increases (at the expense of performance).
- The sign or direction of the effect of the prediction horizon P and control horizon M on robustness can reverse depending on the value of the control weighting λ (Figure 6.7 and 6.8).
  - For small λ, increasing P or decreasing M increases robustness.
  - For large λ, increasing P or decreasing M decreases robustness.
- Even with λ fixed, the effects of P and M on robustness (maximum allowable gain mismatch) are nonlinear.

These conclusions, although example dependent, are obviously much more specific than those based on perturbation theory. A similar root locus analysis of the second order discrete system $(1+0.5q^{-1})/(1-1.5q^{-1}+0.7q^{-2})$ (which has been used frequently by K. J. Aström) leads to similar conclusions.

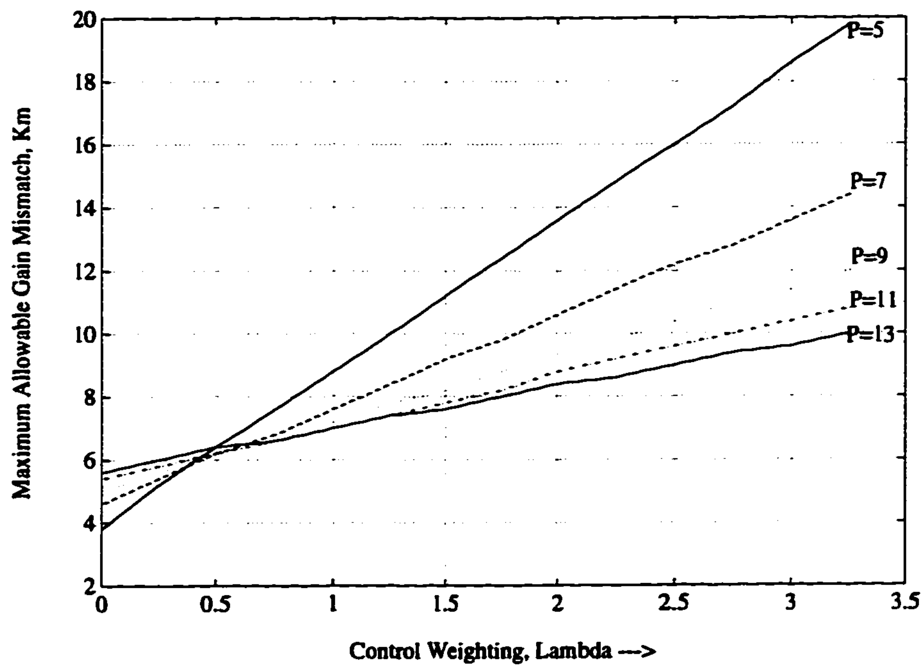

Figure 6.7:The Effect of Tuning Parameters, P and λ on Gain Bound

2. **Example: Time Domain Responses**

   The robustness analysis presented above for a first order system *provides tuning guidelines that in some cases are opposite to widely accepted guidelines for tuning MPC.* Consider the time domain responses presented in Figure 6.9 to Figure 6.12. using the following tuning parameters with $\Delta k = 10$:

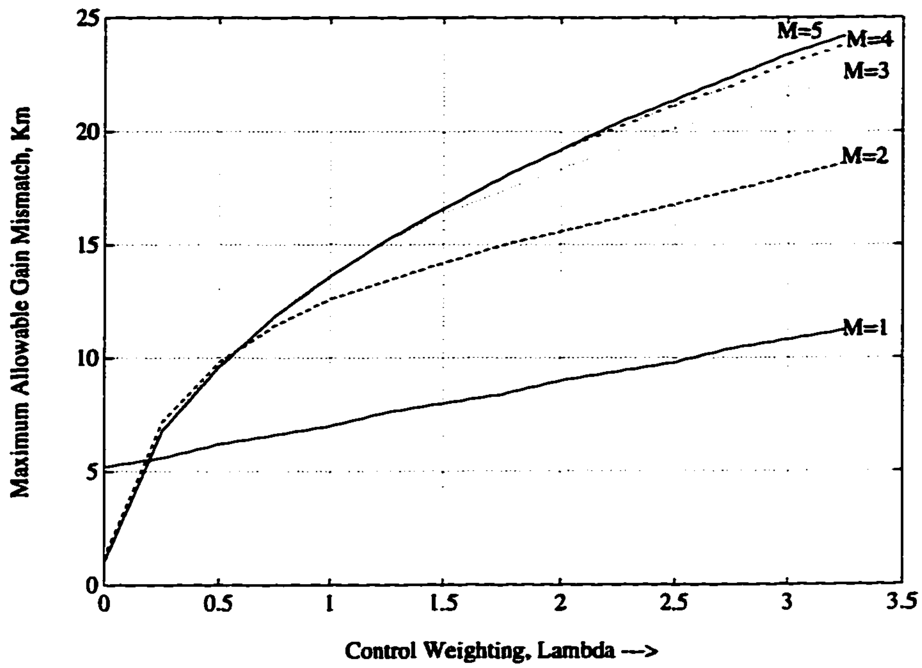

Figure 6.8:The Effect of Tuning Parameters, M and λ on Gain Bound

(a) $\lambda = 2.0,\ P = 7,\ M = 1$, the closed loop system is stable(Figure 6.9).

(b) $\lambda = 2.0,\ P = 9,\ M = 1$, the closed loop system is unstable(Figure 6.10), *i.e.* **increasing P de-stabilized the system.**

(c) $\lambda = 2.0,\ P = 9,\ M = 2$, the closed loop system is stable again(Figure 6.11). *i.e.* **increasing M stabilized the system.**

(d) $\lambda = 0.5,\ P = 9,\ M = 2$, the closed loop system is unstable(Figure 6.12), *i.e.* decreasing λ de-stabilized the system.

The results summarized in bold type are opposite to most widely used tuning guidelines and to intuitions.

3. **Example: Conservativeness of Perturbation Based Approach**

The conservativeness is quite obvious when comparing Figure 6.7 with Figure 6.1(d) (the one with gain uncertainty). With the same MPC tuning parameters ($\lambda = 0, P = 8, M = 1$), the robustness bound calculated from perturbation theory is very small ($\approx$ 0.026) while the true allowable gain mismatch can be up to 5.0. This comparison shows that *the conservativeness of robustness bounds derived based on existing perturbation theorems is large enough to produce misleading results about the effects of MPC tuning parameters on the system robustness.*

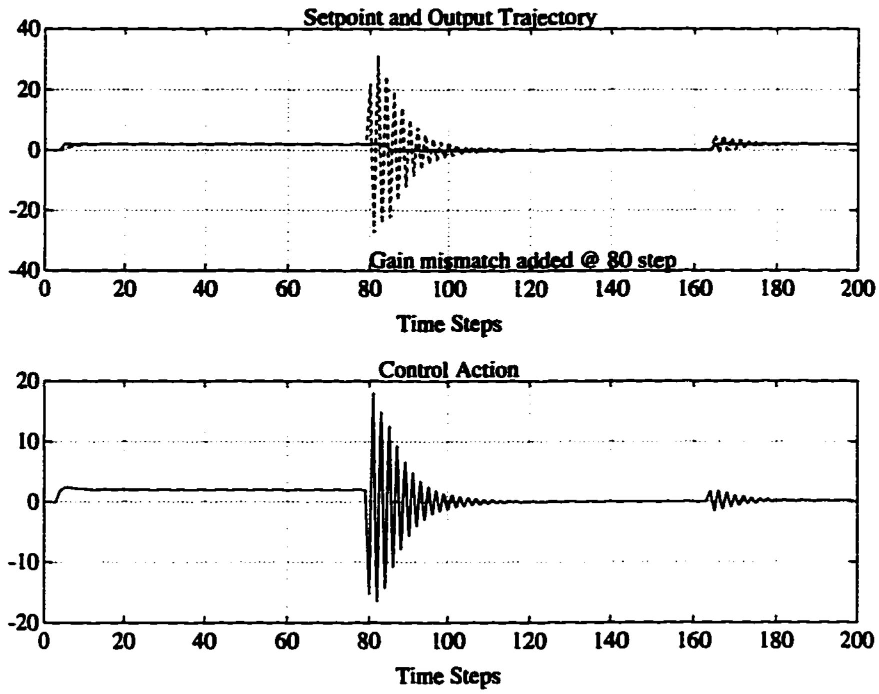

Figure 6.9:Stable MPC Response with $\Delta k = 10$ and $\lambda = 2.0, P = 7, M = 1$

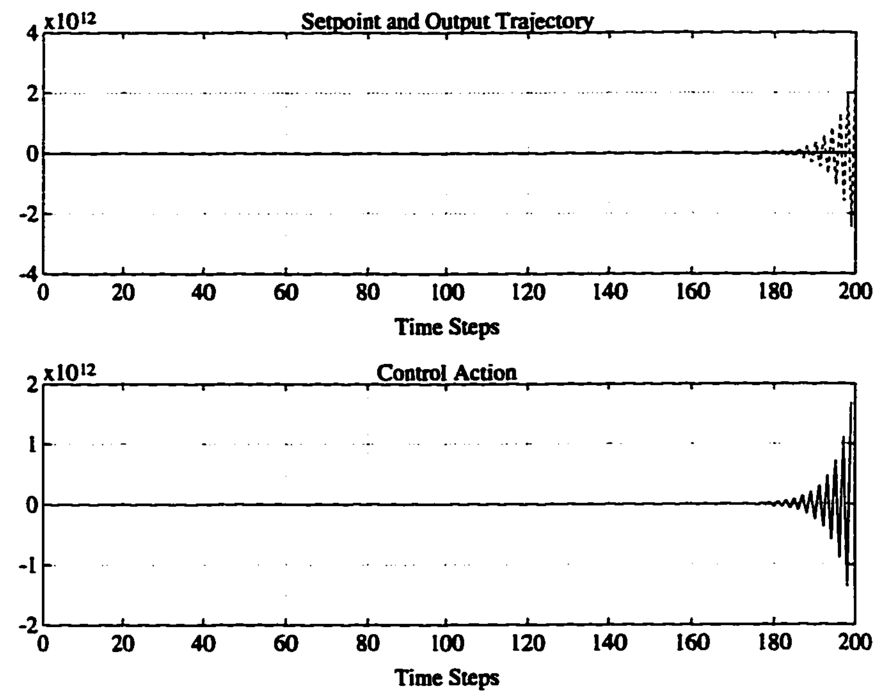

Figure 6.10: Unstable MPC Response with $\Delta k = 10$ and $\lambda = 2.0, P = 9, M = 1$

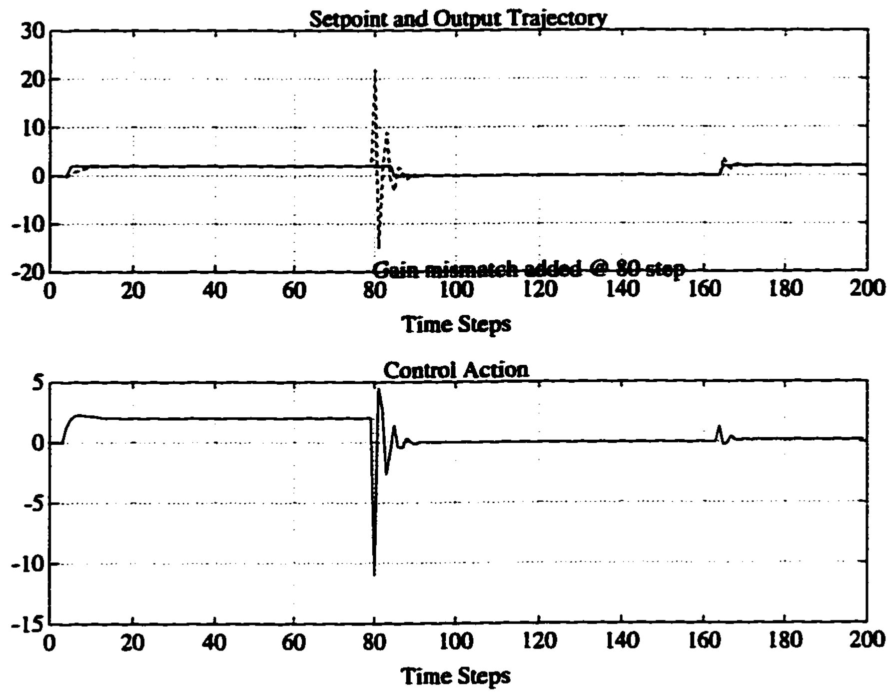

Figure 6.11:Stable MPC Response with $\Delta k = 10$ and $\lambda = 2.0, P = 9, M = 2$

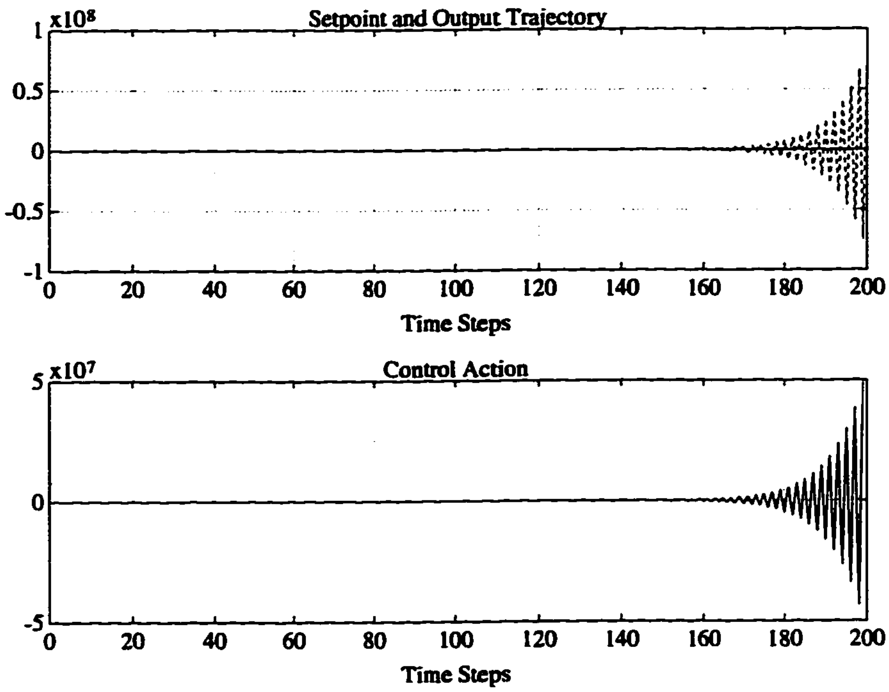

Figure 6.12: Unstable MPC Response with $\Delta k = 10$ and $\lambda = 0.5, P = 9, M = 2$

Table 6.4:The Effects of P and λ on Time-Delay Mismatch, $M = 1$

	Prediction Horizon (P)				
Weighting λ	5	7	9	11	13
0.0	3	3	4	4	4
$0.25 \leq \lambda \leq 1.0$	3	3	4	4	4
$1.25 \leq \lambda \leq 2.0$	3	4	4	4	4
$2.25 \leq \lambda \leq 3.0$	4	4	4	4	4

Table 6.5:The Effects of M and λ on Time-Delay Mismatch, $P = 10$

	Control Horizon (M)				
Weighting λ	1	2	3	4	5
0.0	4	1	1	1	1
0.25	4	2	2	2	2
0.50	4	3	3	2	2
0.75	4	3	3	3	3
1.0	4	4	3	3	3
1.25	4	4	4	3	3
$1.5 \leq \lambda \leq 2.5$	4	4	4	4	4
2.75	4	4	5	4	4
3.0	4	5	5	5	5
3.25	4	5	5	5	5

6.4.2 Effects of MPC Tuning Parameters on Time-Delay Bounds

The maximum allowable mismatch in the process time-delay, Δd_m, was obtained for different combinations of MPC tuning parameters using the same root locus approach described above. The time-delay mismatch was assumed to be an integer multiplier of the sampling interval, T_s.

1. **Example: MPM in the Time-delay**

   Using the same process ($1/(s+1)$, $T_s = 0.3$), Tables 6.4 and 6.5 show the maximum (integer) time-delay mismatch that can be handled using different sets of tuning parameters.

   The results can be summarized as:

   - The robustness to time-delay changes increases as λ increases (*i.e.* rows in Table 6.4 and 6.5).

- Increasing the prediction horizon P allows more time-delay mismatch (*i.e.* columns in Table 6.4).
- For small λ, decreasing M gives more robustness (*i.e.* rows 1 to 6 in Table 6.5).
- For large λ, increasing M tends to allow more time-delay mismatch (*i.e.* rows 9 and 10 in Table 6.5).

2. **Example: Time Domain Responses**

   The last conclusion is illustrated in the time domain, assuming the time-delay mismatch $\Delta d_m = 5$ and selecting the set of controller parameters as:

   (a) $\lambda = 3.0,\ P = 10,\ M = 1$, the closed loop system is unstable (Figure 6.13).

   (b) $\lambda = 3.0,\ P = 10,\ M = 2$, the closed loop system is stable(Figure 6.14), *i.e.* **increasing M stabilized the system**, which is contrary to most tuning guidelines.

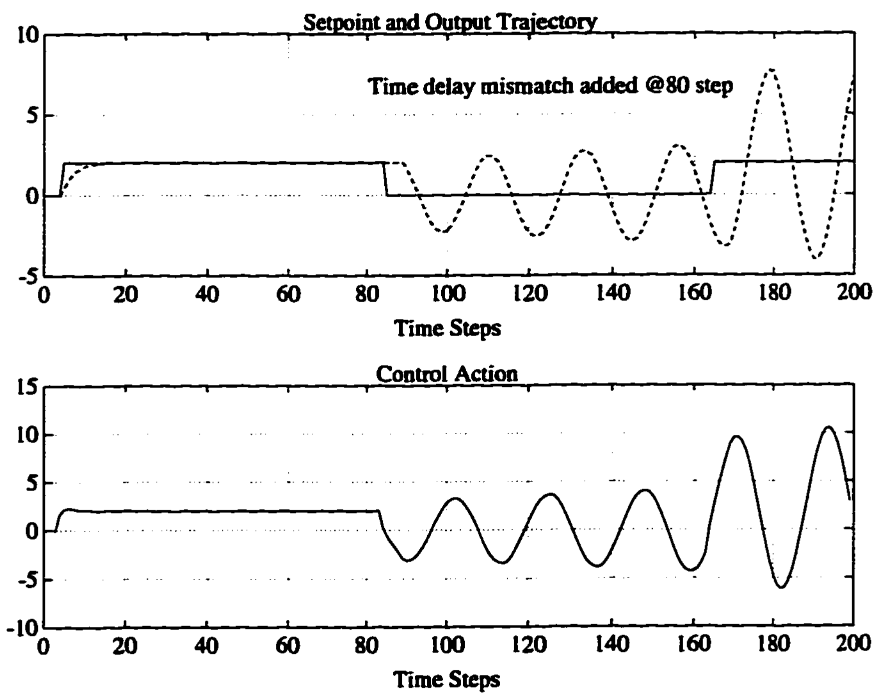

Figure 6.13: Unstable MPC Response with $\Delta d = 5$ and $\lambda = 3.0, P = 10, M = 1$

6.5 Conclusion

1. A dual-model, state space formulation for MPC is formulated and used as the basis for stability and robustness analysis. Specific, less conservative bounds

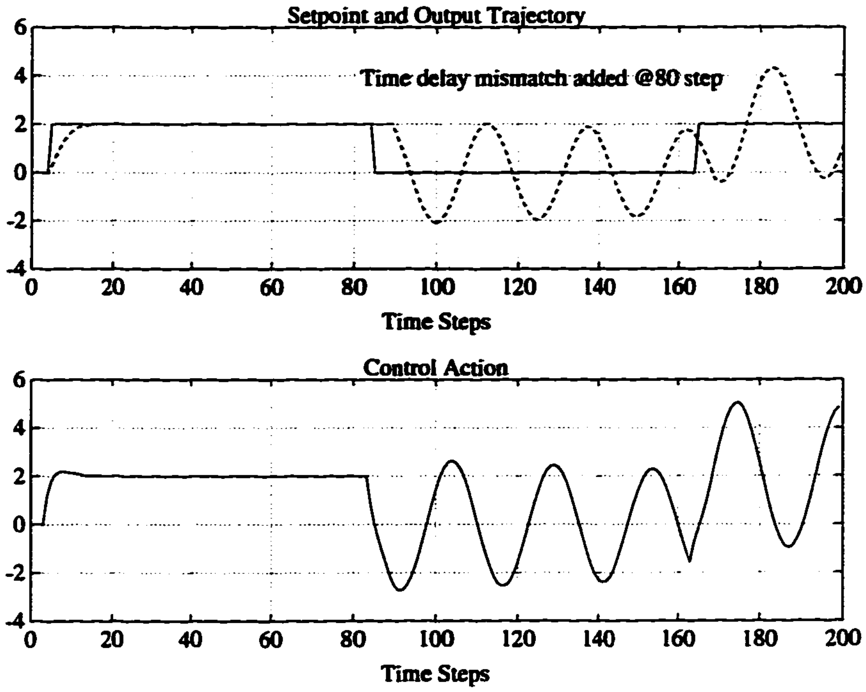

Figure 6.14:Stable MPC Response with $\Delta d = 5$ and $\lambda = 3.0, P = 10, M = 2$

for robust stability are derived for the cases of MPM in the slow modes of the process, the fast modes of the process, the process gain and the process time delay. Tuning guidelines for robust MPC tuning are developed and illustrated by simulations.

2. Root locus techniques used to determine necessary and sufficient bounds for robust stability of a SISO MPC system with MPM in the process gain or time-delay, showed that robust stability analysis based on matrix perturbation can be very conservative and sometimes misleading.

3. The robustness analysis showed that widely accepted MPC tuning guidelines are not always correct. For example, increasing P can de-stabilize and increasing M can stabilize an MPC system in the presence of MPM.

Chapter 7

Constraint Handling and Constrained Stability

7.1 Introduction

In this age of high competition, chemical processes, whether new built or existing, have to maintain production at high efficiencies to reduce operation costs. Production rates have to be pushed closer and closer to, or even over their design capacities so that, with the same capital investment, significant profit increases can be obtained. On the other hand, safety issues in terms of occupational as well as environmental protection have become more and more important for modern automated plants. These new challenges require multivariable and constraint dependent control strategies.

Obviously, the widely applied, SISO based PID control techniques which are the basis of most DCS systems can not be used to solve the interactions and constraints in multivariable systems. For input constraints, the conventional method is simply to cut-off, i.e. cut the absolute input at its limits (usually called control saturation). There is no traditional way to handle output constraints, which mostly relate to the final product quality and/or environmental emissions.

Model predictive control techniques have the ability to satisfy industrial requirements, especially because of their unique ability to handle hard constraints. MPC uses a process model to transform output constraints into input constraints. These output constraints, together with input constraints, form a feasible region for the future control action. Then, an optimization problem can be set up to find the optimal control action within the feasible region. Usually, process control becomes an optimization problem instead of a control problem. However, computational efforts have to be considered for real time control applications. Two effective methods have been used to reduce the computations:

1. Linear models are used to describe the processes and the optimization is also linear or quadratic with linear constraint equations. In this convex type constraint region, the optimal solution is always unique.

2. Supervisory control applications are recommended for use with large scale MPC

applications where numerous local control loops perform regulatory control. With this control structure, the control interval used for MPC is in the minute-domain while embedded local PID controllers run at very fast rates(seconds).

With the rapid developments in both powerful computer hardware and more efficient numerical software packages, the computation itself is not a big concern for the implementation of predictive controllers.

Two issues in the predictive control area need further investigation. The first major concern, especially to academic researchers, is the stability of the constrained predictive control system. It is especially important from the point view of predictive control theory completeness. Obtaining an optimal solution does not guarantee long term stable behaviour. In recent years, important results on constrained stability have been achieved for MPC with quadratic objective functions (Zafiriou 1991, Zafiriou & Marchal 1991, Oliveira & Biegler 1994) and MPC with modified objective functions (Rawlings & Muske 1993). Since these research results are very new to most control researchers, this issue is reinvestigated in this thesis using the dual model state space formulation. Detailed examples are used to clarify the concept and procedures.

For MPC with a linear objective function, the control solution of a well-posed problem is always one of the vertices of the constraint region. The constrained stability is even more important. This problem is discussed in detail in this chapter and a very simple stability criterion is obtained.

Since the constraints result in a set of linear algebraic equations containing the process model coefficients, past inputs, states and outputs, constrained MPC is very sensitive to model uncertainties and disturbances. These effects could produce inconsistent constraints where conventional predictive controllers could not find a solution and simply abort the control application. Practically, this is unacceptable because in those situations, the controller has to take whatever action is necessary (and feasible) to return the process back to normal as soon as possible. In order to avoid constraint inconsistency, some output constraints can be relaxed (Zafiriou & Hung-Wen 1993) or even ignored under some circumstances. This is also the major area of difference among commercial MPC control packages. In this Chapter, a new constraint handling strategy is proposed to deal with this problem.

7.2 State Space Formulation with Constraints

The dual model state equation in Equation (2.13) plus the output prediction in Equation (4.1) can be rewritten here as:

$$\begin{aligned} X(k+1) &= \Phi X(k) + \theta \Delta u(k) \\ &= \Phi X(k) + \theta C^T \Delta U(k) \end{aligned} \tag{7.1}$$

$$Y_p(k) = \Phi_p X(k) + A \Delta U(k)$$

Note that the current control action $\Delta u(k)$ has been dimensionally extended to match the profile $\Delta U(k)$ to be consistent with the output prediction equation $Y_p(k)$.

Usually three types of hard constraint sets are considered by the control algorithm — input limits, incremental or rate limits and output limits. Predictive control can extend these constraint sets to long range dimensions by using the future profiles for the input moves and output predictions. Therefore, constraints can be imposed on the input trajectory $U(k)$ and the output trajectory $Y_p(k)$ and expressed as:

$$\begin{array}{llll} \mathcal{C}_{\Delta u}: & L_{\Delta u} & \leq \Delta U(k) & \leq H_{\Delta u} \\ \mathcal{C}_u: & L_u & \leq U(k) & \leq H_u \\ \mathcal{C}_y: & L_y & \leq Y_p(k) & \leq H_y \end{array}$$

All constraint sets can be transformed into constraints on $\Delta U(k)$ by using the output prediction equation such that

$$\begin{array}{llll} \mathcal{C}_{\Delta u}: & L_{\Delta u} & \leq \Delta U(k) & \leq H_{\Delta u} \\ \mathcal{C}_u: & L_u & \leq L\Delta U(k) + u(k-1) & \leq H_u \\ \mathcal{C}_y:: & L_y & \leq A\Delta U(k) + \Phi_p X(k) & \leq H_y \end{array}$$

where A is the dynamic matrix and L is a unit lower triangle matrix.

REMARKS:

1. The boundaries of the constraint set are time variant because of the inclusion of the past control input $u(k-1)$ and states $X(k)$. Therefore, the on-line control optimization must consider a completely new constraint set every control interval. This increases the amount of time for the algorithm to find a solution on-line.

2. For almost all process control applications, the steady state, *i.e.* $\Delta u(k) = 0, u(k) = u_{ss}$ and $X(k) = X_{ss}$, must also satisfy the constraint equations. Therefore, the constraint boundaries must satisfy

$$\begin{array}{ll} L_{\Delta u} & \leq 0 \\ L_u & \leq u_{ss} \\ L_y & \leq X_{ss} \\ H_{\Delta u} & \geq 0 \\ H_u & \geq u_{ss} \\ H_y & \geq X_{ss} \end{array}$$

3. Theoretically, different limits can be applied to each future control move and each output prediction in the trajectories , *e.g.* $L_u = [l^1,\ l^2,\ l^3]^T$. To simplify the optimization problem, they can be chosen equal to avoid complicating the control algorithm. For example, the low limit of $u(k)$ can then be written as $L_u = l_u[1,\ 1,\ 1]^T$. Therefore, in this thesis, both analysis and simulations use the simplified form without further comment.

4. In a linear convex structure, the inconsistent constraint problem, *i.e.* there would be no common region for the constraint sets, can be detected by building a new linear programming problem (Solow 1984). Efficient algorithms such as the Simplex method for this linear programming problem can be used to find whether there is a inconsistency or not. Even though this topic is beyond the scope of this study, it is important to note that inconsistencies do occur in practical applications due to disturbances, model uncertainties and/or improper constraint limits.

7.3 Stability Analysis of Constrained, Quadratic MPC

7.3.1 Stability Analysis: A Combinatorial Problem

The implementation of constrained predictive control usually becomes an optimization problem which minimize the performance index

$$\begin{aligned} \mathcal{J} &= \|Y_{sp}(k) - \Phi_p X(k) - A\Delta U(k)\|_2 \\ s.t. \quad G\Delta U(k) &\geq b \end{aligned} \tag{7.2}$$

where

$$G = \begin{bmatrix} I \\ -I \\ L \\ -L \\ A \\ -A \end{bmatrix}, \quad b = \begin{bmatrix} L_{\Delta u} \\ -H_{\Delta u} \\ L_u - u(k-1) \\ -H_u + u(k-1) \\ L_y - \Phi_p X(k) \\ -H_y - \Phi_p X(k) \end{bmatrix}$$

Usually, numerical algorithms have to be applied to find the solution for this quadratic optimization problem except some simple cases, *e.g.* SISO control with $M = 1$. With boundary time-varying constraints, the controller structure remains linear but switches from one to another (Zafiriou 1990). This makes the stability analysis very difficult.

A general analysis of the constrained control system stability, by most researchers (Zafiriou 1990, Zafiriou 1991, Oliveira & Biegler 1994), usually starts from the assumption that a set of constraint equations is active, *i.e.*

$$G_a \Delta U(k) = b_a \tag{7.3}$$

where G_a, b_a are subsets, *i.e.* selected rows, of the general constraint matrix G and b respectively.

Note that the quadratic objective function plus a convex constraint set guarantees that the constrained solution must be on the boundaries (recall that a linear program always has its solution at one of the vertices). This property results in some elements of the control profile being determined by the constraint equations, which define the constraint boundaries, while others are not. In another words, the degrees of freedom of the controller are reduced. Because each constraint equation in G can be considered as a whole row element to build the active matrix G_a, mathematically, there are $(2^n - 1)$ possible structures of G_a if the number of constraint equations is n.

The active constraint equation is in a linear algebraic form and can be solved by many methods including numerical algorithms. Note that this algebraic equation is likely underdetermined, *i.e.* the rank of G_a is less than or equal to the number of columns which is the control horizon M. Otherwise, there would be no solution to satisfy all equations, *i.e.* an inconsistent constraint set. For example, a SISO problem with two unknown control variables (*i.e.* $M = 2$) has a maximum of only two active constraint equations. Graphically, one active constraint equation means the optimal solution must be on a single line so that only one control variable is left to be determined by the optimization. If two constraint equations are active, the solution is simply the intersection point of these two lines. Therefore, active constraint equations reduce the degrees-of-freedom in the optimization problem.

The analytic solution of the ill-structured algebraic equation should use the concept of range and null-space decomposition (Golub & Van-Loan 1989). Assume that the rank of G_a is r, as discussed above, and $r \leq r_n \leq M$, where r_n refers to the number of rows in G_a, the problem is to find a full rank sub-space of G_a such that a part of the algebraic equation can be solved to generate components of $U(k)$.

1. Constrained Solution and Stability Analysis

The following five steps define a procedure to perform constrained stability analysis.

- **STEP 1:** perform QR decomposition (which is an extension of the eigenvalue analysis to non-square matrix) on the matrix G_a^T as

$$G_a^T = \mathcal{Q}\mathcal{R}$$

where $\mathcal{Q}$ is an orthogonal matrix, *i.e.* $\mathcal{Q}^T\mathcal{Q} = I$;

- **STEP 2:** exchange the columns of the matrix $\mathcal{R}$ such that $\mathcal{R}$ becomes an upper triangle matrix, *i.e.*

$$G_a^T\mathcal{P} = \mathcal{Q}\mathcal{R}\mathcal{P} = \mathcal{Q}\mathcal{R}_{ut} = [\mathcal{Q}_1 \;\; \mathcal{Q}_2]\begin{bmatrix} \mathcal{R}_1 \\ 0 \end{bmatrix} \tag{7.4}$$

where $\mathcal{R}_1$ is a full rank upper triangle matrix and $\mathcal{P}$ is the permutation matrix. Since $\mathcal{P}$ is constructed by exchanging the columns of an identity matrix, it is also full rank and satisfies $\mathcal{P}^T\mathcal{P} = I$.

Obviously, by multiplying both sides of Equation (7.4) by $\mathcal{P}^T$, this relation can be rewritten as

$$G_a^T = [\mathcal{Q}_1 \;\; \mathcal{Q}_2]\begin{bmatrix} \mathcal{R}_1 \\ 0 \end{bmatrix}\mathcal{P}^T$$

$$G_a = \mathcal{P}\left[\mathcal{R}_1^T\ 0\right]\begin{bmatrix} \mathcal{Q}_1^T \\ \mathcal{Q}_2^T \end{bmatrix}$$

Therefore, replacing G_a in the active constraint equation (7.3), yields

$$\mathcal{P}\left[\mathcal{R}_1^T\ 0\right]\begin{bmatrix} \mathcal{Q}_1^T \\ \mathcal{Q}_2^T \end{bmatrix}\Delta U = b_a \tag{7.5}$$

- **STEP 3:** partition the full control vector ΔU into two components, the range component ΔU_1 and the null-space component ΔU_2,

$$\begin{aligned} \Delta U &= [\mathcal{Q}_1\ \mathcal{Q}_2]\begin{bmatrix} \Delta U_1 \\ \Delta U_2 \end{bmatrix} \\ &= \mathcal{Q}_1\Delta U_1 + \mathcal{Q}_2\Delta U_2 \end{aligned}$$

With this representation, the LHS of the active equation (7.5) becomes

$$\begin{aligned} LHS &= \mathcal{P}\left[\mathcal{R}_1^T\ 0\right]\begin{bmatrix} \mathcal{Q}_1^T \\ \mathcal{Q}_2^T \end{bmatrix}[\mathcal{Q}_1\ \mathcal{Q}_2]\begin{bmatrix} \Delta U_1 \\ \Delta U_2 \end{bmatrix} \\ &= \mathcal{P}\left[\mathcal{R}_1^T\ 0\right]\begin{bmatrix} \mathcal{Q}_1^T\mathcal{Q}_1 & 0 \\ 0 & \mathcal{Q}_2^T\mathcal{Q}_2 \end{bmatrix}\begin{bmatrix} \Delta U_1 \\ \Delta U_2 \end{bmatrix} \\ &= \mathcal{P}\mathcal{R}_1^T\Delta U_1 \end{aligned}$$

where the orthogonal property of $\mathcal{Q}$ is used such that

$$\mathcal{Q}_1^T\mathcal{Q}_1 = I, \mathcal{Q}_1^T\mathcal{Q}_2 = 0$$
$$\mathcal{Q}_2^T\mathcal{Q}_1 = 0, \mathcal{Q}_2^T\mathcal{Q}_2 = I$$

Since both matrix $\mathcal{P}$ and $\mathcal{R}_1$ are full rank, the range component ΔU_1 can be completely determined from the algebraic constraint equation as

$$\Delta U_1 = \mathcal{R}_1^{-T}\mathcal{P}^T b_a \tag{7.6}$$

- **STEP 4:** find the null-space control component ΔU_2 from the unconstrained optimization problem where the objective function becomes

$$\begin{aligned} \min \mathcal{J} &= \|Y_{sp}(k) - \Phi_p X(k) - A\Delta U(k)\| \\ &= \|Y_{sp}(k) - \Phi_p X(k) - A\mathcal{Q}_1\Delta U_1(k) - A\mathcal{Q}_2\Delta U_2(k)\| \end{aligned} \tag{7.7}$$

The unconstrained solution is therefore obtained as

$$\Delta U_2(k) = (A\mathcal{Q}_2)^*[Y_{sp}(k) - \Phi_p X(k)] - (A\mathcal{Q}_2)^* A\mathcal{Q}_1\Delta U_1(k) \tag{7.8}$$

Finally, the full control variable ΔU can be obtained by combining these two components from Equation (7.6) and Equation (7.8)

$$\begin{aligned} \Delta U &= \mathcal{Q}_1\Delta U_1 + \mathcal{Q}_2\Delta U_2 \\ &= \mathcal{Q}_2(A\mathcal{Q}_2)^*[Y_{sp}(k) - \Phi_p X(k)] + [I - \mathcal{Q}_2(A\mathcal{Q}_2)^* A]\mathcal{Q}_1\Delta U_1(k) \\ &= \mathcal{Q}_2(A\mathcal{Q}_2)^*[Y_{sp}(k) - \Phi_p X(k)] + [I - \mathcal{Q}_2(A\mathcal{Q}_2)^* A]\mathcal{Q}_1\mathcal{R}_1^{-T}\mathcal{P}^T b_a \end{aligned} \tag{7.9}$$

- **STEP 5:** find the constrained state feedback gain K^c_{MPC} using the calculated control moves as

$$K^c_{MPC}X(k) = C^T \Delta U(k)$$

Then, check the eigenvalues of the closed loop matrix $(\Phi - \theta K^c_{MPC})$.

It is interesting to note that whenever any constraint is active, the original (unconstrained) state feedback control law is changed. The effect of constraints on closed loop stability can be summarized as:

(a) If the range component ΔU_1 in Equation (7.6) is a function of the state variable (*i.e.* b_a contains the state variable $X(k)$), an extra state feedback path will be introduced into the closed loop system. The stability must be re-evaluated by re-calculation of the eigenvalues of the system matrix. For example, if any output constraint is active, the right hand side of the active equation b_a is state variable dependent which may change the closed loop stability properties.

(b) Even if ΔU_1 is not state dependent, the nominal unconstrained stability analysis may still change due to the fact that the dynamic matrix A now becomes AQ_2 in Equation (7.9).

(c) For the extreme case of $Q_2 = 0$, *i.e.* all control variables are decided by the constraint equation, the stability is totally independent of the unconstrained optimal solution and independent of the tuning parameters specified for conventional MPC. For example, control weighting and output weighting do not have any effect at all. The control horizon and output horizon would have a very limited effect. The control law simply becomes

$$\Delta U = Q_1 \Delta U_1 = G_a^{-1} b_a$$

Again, the effect of constraints on stability depends on whether b_a is state dependent or not. State dependent b_a changes the feedback structure of the system. State independent b_a does not give any state feedback so that the system would behave as an open-loop process.

(d) On the other hand, if $Q_2 = I$, there is no component determined by the active constraint equation.

$$I - Q_2(AQ_2)^* A = 0$$

The control variable is exactly equal to the unconstrained optimal solution.

2. Example

Let's consider the 3rd order SISO system in Equation (2.17), *i.e.*

$$G(s) = \frac{1}{(s+1)(3s+1)(5s+1)}$$

with the sampling interval $T_s = 1$ and the dual-model order $n = 8$. An MPC controller is designed using $P = 8, M = 3, \lambda = 0$ to obtain a stable closed loop system with the largest unconstrained poles at $(-0.0695 \pm 0.1398j)$.

Assuming that the hard constraints have the form of

$$\begin{array}{rcll} -0.5 & \leq & \Delta U(k)_{3\times 1} & \leq 0.5 \\ -2.0 & \leq & U(k)_{3\times 1} & \leq 2.0 \\ -1.5 & \leq & Y_p(k)_{8\times 1} & \leq 1.5 \end{array}$$

With these fourteen constraint equations, there are a total of $(2^{14} - 1 = 16383)$ combinations of possible active constraint equations. Then, for each combination, the above stability analysis has to be performed.

Assume that the active constraint equation set is

$$\begin{bmatrix} 0.0435 & 0.0077 & 0 \\ 0.1064 & 0.0435 & 0.0077 \end{bmatrix} \Delta U(k) = \begin{bmatrix} -1.5 \\ 1.5 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & 0 & \cdots & 0 \end{bmatrix} X(k)$$

i.e. the low limit on the 1st output prediction and the high limit on the 2nd output prediction.

The active constraint matrix G_a can then be decomposed as

$$\begin{aligned} G_a^T &= \begin{bmatrix} 0.0435 & 0.1064 \\ 0.0077 & 0.0435 \\ 0 & 0.0077 \end{bmatrix} \\ &= \underbrace{\begin{bmatrix} -0.9847 & 0.1662 & | & 0.0527 \\ -0.1743 & -0.9387 & | & -0.2976 \\ 0 & -0.3022 & | & 0.9532 \end{bmatrix}}_{\mathcal{Q}_1 \quad \mathcal{Q}_2} \begin{bmatrix} -0.0442 & -0.1124 \\ 0 & -0.0255 \\ \hline 0 & 0 \end{bmatrix} \begin{matrix} \mathcal{R}_1 \\ \\ 0 \end{matrix} \end{aligned}$$

With a full rank $\mathcal{R}_1$, the range component ΔU_1 is

$$\begin{aligned} \Delta U_1 &= \mathcal{R}_1^{-1} b_a \\ &= \begin{bmatrix} -22.64 & 99.82 \\ 0 & -39.25 \end{bmatrix} \left(\begin{bmatrix} -1.5 \\ 1.5 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & 0 & \cdots & 0 \end{bmatrix} X(k) \right) \\ &= \begin{bmatrix} 183.67 \\ -58.87 \end{bmatrix} - \underbrace{\begin{bmatrix} 0 & 0 & -22.64 & 99.82 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -39.25 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{K_{mpc}^{c1}} X(k) \end{aligned}$$

Then, using the constrained calculation in Equation (7.9), the constrained state feedback gain can be obtained as:

$$\begin{aligned} K_{mpc}^{c} &= C^T \mathcal{Q}_2 (A\mathcal{Q}_2)^* \Phi_p - C^T [I - \mathcal{Q}_2 (A\mathcal{Q}_2)^* A] \mathcal{Q}_1 K_{mpc}^{c1} \\ &= [0 \;\; 0.0004 \;\; 0 \;\; 0 \;\; 0.0194 \;\; 0.0597 \;\; 0.1141 \;\; 0.1752 \;\; 0.2378] \end{aligned}$$

The closed loop stability can be determined from the eigenvalues of $(\Phi - \theta K^c_{mpc})$. Using the values calculated above, it turns out that the largest eigenvalues are $(0.7851 \pm 0.2091j)$. For this active constraint set, the control system is still stable. But remember that this is just one case of 16383 possible active constraint sets. For a complete stability analysis, all 16383 constraint combinations would have to be analyzed.

The above analysis shows that mathematically there is no problem determining the constrained stability of MPC. However, it is almost impossible to apply it to practical applications without simplification. Two direct applications of the above results are MPC control with a linear optimization objective function (MPC-LP) as discussed in Section 7.4 and MPC control with a control horizon $M = 1$.

7.3.2 SISO with $M = 1$, A Simplified Analysis

For the SISO process with $M = 1$, the constraint matrix G becomes a simple vector with the form

$$G = \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \\ s_1 \\ s_2 \\ \vdots \\ s_p \\ -s_1 \\ -s_2 \\ \vdots \\ -s_p \end{bmatrix} \qquad b = \begin{bmatrix} l_{\Delta u} \\ -h_{\Delta u} \\ l_u - u(k-1) \\ -h_u + u(k-1) \\ l_y - x_1(k) \\ l_y - x_2(k) \\ \vdots \\ l_y - x_p(k) \\ -h_y + x_1(k) \\ -h_y + x_2(k) \\ \vdots \\ -h_y + x_p(k) \end{bmatrix}$$

Obviously, the assumption of only one future control move, *i.e.* $M = 1$ does simplify the constraint set because all types of constraints can be converted into upper/lower limits on the control input. No complicated numerical procedures are involved. The optimal solution is found by simply reducing the unconstrained solution (if it is outside of the bounds) to the closest bound. Many applications, especially these involving control of sluggish processes, choose the control horizon $M = 1$ to avoid numerical computations. Even for some MIMO systems, *e.g.* 2×2, simplified solutions are also available (Mutha 1990).

The simplified constraint set obtained with $M = 1$ also makes it easy to analyze the stability of the constrained control system. Since only one constraint equation may be active at any time instant, the active constraint matrix is either $G_a = 1$ or $G_a = s_i$. There are $(P + 2)$ possible structures for G_a. Depending on what values are used in b_a, the closed loop stability can be discussed in two cases — input or output constraints.

1. **Input Constraints**

If only the constraints on ΔU or U are active, the constrained solution is

$$\Delta u(k) = \mathcal{F}(l_{\Delta u}, h_{\Delta u}, l_u, h_u, u(k-1))$$

Recall that the open loop state equation is

$$X(k+1) = \Phi X(k) + \theta \Delta u(k)$$

Since the control move $\Delta u(k)$ is not a function of the state variables, $X(k)$, and is bounded, the closed loop matrix is the same as the one in the open loop formulation. Even though the transient responses of the process states may change due to the limitations on the input energy, the asymptotical properties of the closed loop system are not different from the open loop process. Open loop stable processes are still closed loop stable when input constraints are active. But open loop unstable processes become unstable if input constraints are imposed. Conceptually, this is obvious since unstable processes require unbounded input energy to keep them under control.

The input constrained controller also finds application in many processes since it is very simple and intuitive. Simple Bang-Bang control is a typical example. Even though the controlled variable fluctuates, its average value is controlled to the desired point and there is no stability problem.

2. **Output Constraints**

Assume that one constraint equation (upper limit) is active,

$$G_a \Delta u(k) = b_a$$

where $G_a = s_i$ is a simple scalar and b_a is

$$\begin{aligned} b_a &= B[H_y - \Phi_p X(k)] \\ B &= [\underbrace{0 \cdots 0}_{i-1} \; 1 \; 0 \cdots 0] \end{aligned}$$

The constrained solution can then be obtained by solving the active constraint equation

$$\Delta u(k) = s_i^{-1}(BH_y - B\Phi_p X(k))$$

Clearly, the state variable $X(k)$ is fed back by the constrained control move. The closed loop system equation is obtained by substituting the control law back into the open loop state equation, *i.e.*

$$\begin{aligned} X(k+1) &= \Phi X(k) + \theta \Delta u(k) \\ &= (\Phi - s_i^{-1}\theta B \Phi_p) X(k) + s_i^{-1}\theta B H_y \end{aligned}$$

Since the system structure is still linear, closed loop stability can be determined from the eigenvalues of the system matrix $(\Phi - s_i^{-1}\theta B\Phi_p)$. The state feedback term does change the open loop elements in Φ. Since both B and Φ_p have specific structures, the state feedback can be written in an explicit form as

$$s_i^{-1}\theta B\Phi_p = \begin{bmatrix} 0 & \cdots & 0 & s_1/s_i & 0 & \cdots & 0 \\ 0 & \cdots & 0 & s_2/s_i & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & s_{i-1}/s_i & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & s_{i+1}/s_i & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & s_n/s_i & 0 & \cdots & 0 \end{bmatrix}$$

REMARKS:

- Since i can be any integer number from 1 to P, there are a total of P cases that must be evaluated to ensure closed loop stability.
- The extra state feedback affects the columns of the open loop dynamic matrix. Corresponding changes in the eigenvalues are expected.
- Traditional MPC tuning parameters, λ, γ_s, etc, do not have any direct impact on the closed loop performance. They do affect which constraint set becomes active though.

3. **Simulation Result**

Consider a first order plus time-delay SISO system with

$$G(s) = \frac{e^{-0.06}}{(s+1)}$$

The corresponding discrete model with $T_s = 0.1$ is

$$G(q^{-1}) = \frac{0.03921q^{-1} + 0.056q^{-2}}{1 - 0.9048q^{-1}}$$

The control horizon and prediction horizon are $M = 1, P = 5$ which results in seven possible active constraint sets. The first two are constraints on the input and the last five are constraint equations on the output predictions. The spectral radius of the corresponding closed loop system is shown in Figure 7.1, where the constraint set #0 refers to the spectral radius of the unconstrained system.

With the control weighting $\lambda = 0$, the *unconstrained* solution is stable with the largest pole located at 0.6430. Based on the spectral radius information in Figure 7.1, the active output constraint does however cause closed loop instability for this example. The active constraint on the 1st prediction of the output, $y(k+1|k)$, results in an unstable closed loop pole at 1.4292. To prevent this problem, this output constraint on $y(k+1|k)$ could be relaxed so that it is never active during the control calculations. This simple example shows that it is important to analyze the properties of the constrained system at the design stage such that reasonable and consistent constraint limits can be applied.

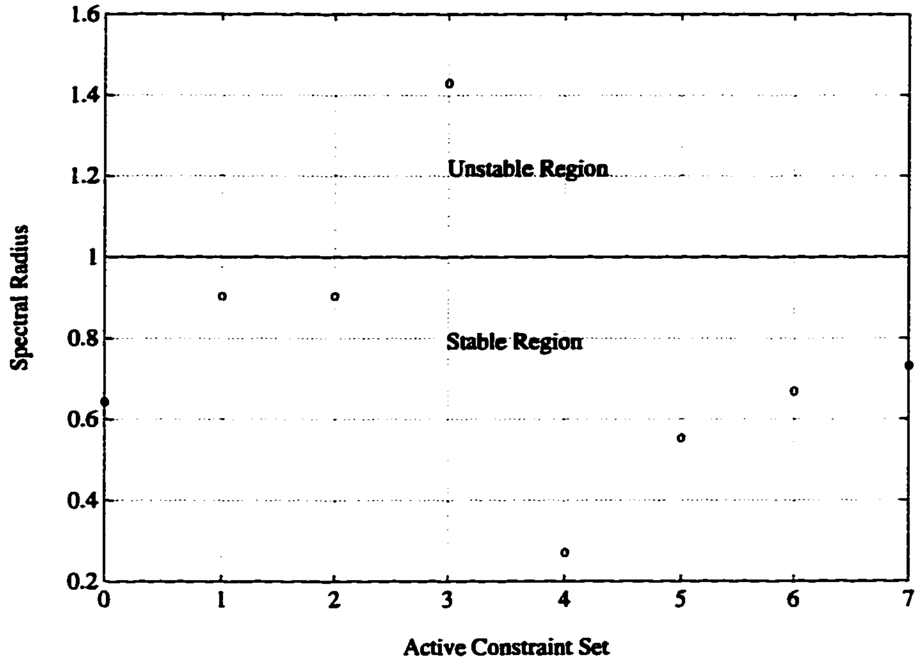

Figure 7.1:Closed Loop Spectral Radius for Active Constraints

7.3.3 Summary

Predictive controllers using a linear model description plus linear constraint equations become piece-wise linear, state feedback controllers when hard constraints are active. Closed loop stability can be still evaluated by examining the system eigenvalues, but there are many constraint combinations that must be considered. To guarantee closed loop system stability, all possible linear structures have to be stable even though some structures are never active during the control calculation.

Constraints on the process inputs do not change the open loop characteristics, but may result in problems for open loop unstable processes. Usually, input constraints are not a major concern in predictive control design since firstly, open loop unstable systems are rare in chemical processes, and secondly, most input constraints are only temporarily active.

As discussed earlier, output constraints are mapped into input constraints (using the process model) as part of the formulation of the constrained optimization problem. Thus it might appear at first that all constraints could be treated as input constraints. However, constraints on the process outputs (and/or states) introduce extra state feedback and therefore affect the closed loop stability significantly. Theoretically, their effects can be evaluated by considering every possible control structure. However, there are usually too many combinations even for a simple case such as $M = 1$. For the majority of multivariable applications, it is not practical to do a complete stability analysis if output constraints are specified.

The design procedure discussed above for constrained predictive controllers usually results in a conservative design. At the design stage, without the knowledge of which constraint set will be active, all possible constrained control structures must be stable even though some constraint sets will never be active. Also many constraints are only active momentarily so that their asymptotical effects, *i.e.* stability, do not have the chance to show up. However, for many constrained applications, the optimal solution (even at steady state) lies at the intersection of a set of active constraints. Obviously, the subset of constraints that could be active at steady state must be considered carefully.

7.4 MPC with Linear Programming (MPC-LP)

Using a linear objective function, the MPC formulation is the same as that in Equation (7.2) except that its optimization index becomes

$$\min \mathcal{J} = |Y_{sp}(k) - \Phi_p X(k) - A\Delta U(k)|$$

The MPC-LP control action has two distinct properties:

1. There is no unconstrained solution for MPC-LP;
2. The control solution of a well-posed problem is always at one vertex of the feasible constraint region.

Therefore, mathematically, all control components are determined by the full-rank, active linear constraint equations as:

$$\Delta U = G_a^{-1} b_a$$

For this case, there is no need to do any range and null-space decomposition. The solution can be taken into the dual-model state space formulation (7.1) as

$$\begin{aligned} X(k+1) &= \Phi X(k) + \theta C^T \Delta U(k) \\ &= \Phi X(k) + \theta C^T G_a^{-1} b_a \end{aligned}$$

The closed loop stability of the constrained MPC-LP depends totally on the open loop stability defined by Φ, and the active constraint equations in G_a and b_a.

There are two ways to handle the constrained stability problem of MPC-LP. One is to treat it as a special case of MPC-QP by observing that MPC-QP essentially considers all possible structures of G_a while MPC-LP considers only full ranked G_a structures. Then, the number of constrained control structures is reduced but is still too large for use in practical applications. Another approach is to relate MPC-LP to unconstrained MPC-QP which substantially simplifies the stability problem of MPC-LP. This approach is discussed in the following subsections in detail.

7.4.1 Input Constraints Active Only

If only input constraints (absolute and/or incremental) are active, *i.e.*

$$b_a = \mathcal{F}[L_{\Delta u},\ H_{\Delta u},\ L_u,\ H_u,\ u(k-1)]$$

The addition of the control actions to the state space formulation does not change the steady state properties since the input energy is limited. Therefore, the closed loop stability of the constrained MPC-LP is exactly the same as the open loop stability.

For example, with $M = 2$, if an upper limit on $\Delta u(k|k)$ and a lower limit on $u(k+1|k)$ are active, the constraint equation is

$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{pmatrix} \Delta u(k|k) \\ \Delta u(k+1|k) \end{pmatrix} = \begin{pmatrix} 0.5 \\ -0.5 - u(k-1) \end{pmatrix}$$

The control actions can then be solved as

$$\begin{pmatrix} \Delta u(k|k) \\ \Delta u(k+1|k) \end{pmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}^{-1} \begin{pmatrix} 0.5 \\ -0.5 - u(k-1) \end{pmatrix} = \begin{pmatrix} 0.5 \\ -1 - u(k-1) \end{pmatrix}$$

With this control action, the closed loop system becomes

$$\begin{aligned} X(k+1) &= \Phi X(k) + \theta [1,\ 0] \begin{pmatrix} 0.5 \\ -1 - u(k-1) \end{pmatrix} \\ &= \Phi X(k) + 0.5\theta \end{aligned}$$

Obviously, the open loop poles in Φ are not moved as a result of this constrained solution.

7.4.2 Output Constraints Active Only

If all active constraints are output variable related,

$$b_a = \mathcal{F}[L_y,\ H_y,\ X(k)]$$

The presence of the state variable vector $X(k)$ in the constraint equation and hence in the control calculation introduces a feedback term to the control system. Closed loop stability is therefore changed by the active constraints.

All output constraint equations have the form

$$A\Delta U(k) + \Phi_p X(k) \leq B_y \tag{7.10}$$

Note that B_y refers to the output constraint bound vector consisting of either lower bounds or upper bounds.

Since the active constraint matrix G_a is a subset of the dynamic matrix A in Equation (7.10), with the orthogonal permutation matrix $\mathcal{P}$,

$$\begin{aligned} \mathcal{P}A &= \begin{pmatrix} G_a \\ G_b \end{pmatrix} \\ \mathcal{P}\Phi_p &= \begin{pmatrix} \Phi_{p,a} \\ \Phi_{p,b} \end{pmatrix} \\ \mathcal{P}B_y &= \begin{pmatrix} B_{y,a} \\ B_{y,b} \end{pmatrix} \end{aligned}$$

Then, Equation (7.10) can be decomposed as two equation sets, an *active equality* set and an *inactive inequality* set, so that

$$G_a \Delta U(k) + \Phi_{p,a} X(k) = B_{y,a} \tag{7.11}$$

$$G_b \Delta U(k) + \Phi_{p,b} X(k) < B_{y,b} \tag{7.12}$$

The constrained solution obtained from Equation (7.11) is

$$\Delta U(k) = G_a^{-1} B_{y,a} - G_a^{-1} \Phi_{p,a} X(k) \tag{7.13}$$

Substituting $\Delta U(k)$ into Equation (7.12), yields

$$(-G_b G_a^{-1} \Phi_{p,a} + \Phi_{p,b}) X(k) < B_{y,b} - G_b G_a^{-1} B_{y,a} \tag{7.14}$$

The closed loop system becomes

$$\begin{aligned} X(k+1) &= (\Phi - \theta C^T G_a^{-1} \Phi_{p,a}) X(k) + \theta C^T G_a^{-1} B_{y,a} \\ &= (\Phi - \theta K_{mpc}^{LP}) X(k) + \theta C^T G_a^{-1} B_{y,a} \end{aligned}$$

Obviously, the open loop poles are shifted by the state feedback gain vector of MPC-LP, which is expressed as $K_{mpc}^{LP} = C^T G_a^{-1} \Phi_{p,a}$.

Again, the closed loop stability can be evaluated by the eigenvalues of $(\Phi - \theta K_{mpc}^{LP})$. If the stability analysis discussed in the last section is used, it becomes a combinatorial problem with too many possible structures of G_a to be considered.

A different approach is developed here. Note that the active constraint matrix G_a is a full-rank subset of the dynamic matrix A which is used for the unconstrained solution of MPC-QP. It is very interesting to examine the relationship between the unconstrained MPC-QP's state feedback K_{mpc} and the constrained MPC-LP's K_{mpc}^{LP} which are defined as

$$\begin{aligned} K_{mpc} &= C^T A^* \Phi_p \\ K_{mpc}^{LP} &= C^T G_a^{-1} \Phi_{p,a} \end{aligned}$$

An explicit explanation is that K_{mpc} is a *least squares* solution for all state equations and K_{mpc}^{LP} is the *exact* solution for some selected elements of the state variable. Mathematically, it is possible to find the relation between K_{mpc} and K_{mpc}^{LP} since

$$\begin{aligned} K_{mpc} &= C^T A^* \Phi_p \\ &= C^T (A^T A)^{-1} A^T \Phi_p \\ &= C^T (G_a^T G_a + G_b^T G_b)^{-1} (G_a^T \Phi_{p,a} + G_b^T \Phi_{p,b}) \end{aligned}$$

where the orthogonal property of the permutation matrix $\mathcal{P}$ is used.

Define matrices P_a and P_{a+b} as,

$$\begin{aligned} P_a &= (G_a^T G_a)^{-1} \\ P_{a+b} &= (G_a^T G_a + G_b^T G_b)^{-1} \end{aligned}$$

and use the well known Lemma:

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1}$$

so that

$$\begin{aligned} P_{a+b} &= (P_a^{-1} + G_b^T G_b)^{-1} \\ &= P_a - P_a G_b [I + G_b^T P_a G_b]^{-1} G_b^T P_a \end{aligned}$$

After some algebraic manipulation, the gain vector can be simplified to

$$\begin{aligned} K_{mpc} &= C^T P_{a+b} (G_a^T \Phi_{p,a} + G_b^T \Phi_{p,b}) \\ &= K_{mpc}^{LP} - C^T P_a G_b^T [I + G_b P_a G_b^T]^{-1} [-\Phi_{p,b} + G_b G_a^{-1} \Phi_{p,a}] \end{aligned} \tag{7.15}$$

REMARKS:

- The unconstrained MPC-QP's state feedback gain K_{mpc} is made up of the constrained MPC-LP's feedback gain K_{mpc}^{LP} and a correction term related to all output constraint equations.
- If all output constraints are active, *i.e.* $M = P, G_a = A, \Phi_{p,a} = \Phi_p$, the unconstrained K_{mpc} is exactly the constrained K_{mpc}^{LP}. The closed loop stability of MPC-LP is also equivalent to that of unconstrained MPC-QP.

An important result on the stability of MPC-LP can be obtained by further analysis of the state feedback gain relationship in Equation (7.15).

Theorem 7.1 *The output constrained MPC-LP system is closed loop stable if the corresponding MPC-QP system is unconstrained stable.*

Proof:

The closed loop system of MPC-LP is

$$\begin{aligned}
X(k+1) &= (\Phi - \theta K_{mpc}^{LP})X(k) \\
&= (\Phi - \theta K_{mpc})X(k) + C^T P_a G_b^T [I + G_b P_a G_b^T]^{-1} [\Phi_{p,b} - G_b G_a^{-1} \Phi_{p,a}] X(k) \\
&\quad (\textit{applying Equation } (7.14)) \\
&< (\Phi - \theta K_{mpc})X(k) + C^T P_a G_b^T [I + G_b P_a G_b^T]^{-1} [B_{y,b} - G_b G_a^{-1} B_{y,a}] \\
&= (\Phi - \theta K_{mpc})X(k) + Constant
\end{aligned}$$

With constant vector additions, the closed loop stability of MPC-LP is represented by the eigenvalues of $(\Phi - \theta K_{mpc})$ which is the closed loop matrix of the unconstrained MPC-QP control system. Stable unconstrained MPC-QP guarantees that MPC-LP is also stable. □

If the active constraint set has both input and output constraint equations, the control action ΔU can be partitioned as two parts, ΔU_i and ΔU_o, using QR decomposition procedures similar to these discussed above. Since only the output constraint relevant component ΔU_o introduces state feedback, the whole control system then becomes partially open-loop and partially closed loop. The stability result about MPC-LP can still applied to this case, as long as the control system is open loop stable. Detailed proofs are omitted here.

One direct application of Theorem (7.1) on MPC-LP is to use a linear optimization index in the MPC control system (Morshedi, Cutler & Skrovanek 1985, Lim 1988). As long as the *unconstrained* MPC-QP formulation is stable, the output constrained closed loop stability of MPC-LP can be guaranteed. To ensure at least one active output constraint equation, it is suggested that a end-condition type, output constraint be added to the original MPC-LP formulation, *e.g.* (Genceli & Nikolaou 1993). A relaxed endpoint constraint has also been used to guarantee the feasibility of constrained GPC, *e.g.* (Rossiter, Kouvaritakis & Gossner 1996).

The drawback of using MPC-LP is that there is no analytic solution even for the unconstrained case. MPC-LP control becomes a constraint driven optimization problem. Numerical searching must be used to find the solution.

Another application is to add a linear optimization index, $\mathcal{J}_2$, involving only critical output predictions (instead of all output predictions as in MPC-LP) whose active constraints would cause stability problem if a QP formulation were used (Oliveira & Biegler 1994). Using LP, the stability of the unconstrained system will not be affected. A more detailed discussion is given in Section 7.5.

7.5 Constraint Handling and Softening

Even though the predictive controller takes constraints into account in its calculations, the constraint violations can not be guaranteed at this stage. Disturbances as well as model uncertainties can still drive the controlled system to unexpected conditions which could be outside the constraint bounds. Therefore, constraint infeasiblities must be considered by any practical controller. The easy solution is to disable the control application and produce an alarm to operators. This is usually unacceptable. Two methods are recommended to solve this problem. One is to impose constraints carefully and only when it is necessary. Another is to relax certain constraints.

7.5.1 Constraint Window

Mathematically, it is not a problem to represent all kinds of hard constraints in an optimization format and to analyze their effects on the closed loop stability. However, it has been shown that the hard constraints requirements, especially the constraints on output variables and their predictions, are usually too restrictive for real applications. The original MPC algorithm applies output constraints on every point of the prediction horizon. The large number of output constraints would not only complicate the optimization but also increase the possibility of constraint inconsistency. Due to the unavoidable model uncertainties, it is admitted that it may be physically impossible to provide 100 percent constraint enforcement (Froisy 1994).

In practice, output constraints can be applied to only a part of the prediction horizon - *constraint window*. For example, for processes with NMP, the initial portion of the closed loop response can be ignored to improve the control performance (Garcia & Morshedi 1986). Theoretical results have also been reported which relax the constraints for a finite (initial) time j_1 (Rawlings & Muske 1993). Output constraints can also be applied only to the steady state prediction rather than the dynamic portion of the predictions. To consider the steady state constraints, one popular approach is to change the setpoint specification accordingly for dynamic constraint enforcement.

There are two kinds of constraints, hard constraints and soft constraints. Hard constraints are those do not allow any violation. However, temporary violations of soft constraints may be acceptable in some applications and hence constraint softening techniques can be applied. This can relax some constraint requirements for the predictive controllers. Since the output constraints may cause stability problems and require numerical searching, they should, if practical, be relaxed for the control stage.

7.5.2 Constraint Softening: An Introduction

Generally, the procedure used for softening constraints is to find an active constraint set first. Then, if the constraint can be relaxed, an extended objective function is used for the MPC calculation, *i.e.*

$$\mathcal{J} = \mathcal{J}_1 + \mathcal{J}_2$$

where $\mathcal{J}_1$ is the conventional MPC objective function, *i.e.*

$$\mathcal{J}_1 = \sum_{j=1}^{P}(y_{sp}(t+j) - y(t+j))^2$$

and $\mathcal{J}_2$ is a term to minimize the constraint violation.

$\mathcal{J}_2$ can be defined as $\beta\|\epsilon\|$ (Li & Biegler 1989, Zafiriou 1991, Feher & Erickson 1993) or $\|\epsilon\|^1, \|\epsilon\|^\infty$ (Oliveira & Biegler 1994). ϵ is called the relaxation factor. For example, let's assume the active output constraint is y_b such that $y = y_b$ is the bounded solution for the constrained MPC, then ϵ can be defined as

$$\begin{aligned} \epsilon &= y_b - y \\ &= y_b - y_m + a\Delta U \end{aligned}$$

An new optimization problem can then be solved which penalizes the constraint violation but allows temporary violations. Usually there are two problems in this kind of arrangement:

1. **Combinatorial Problem:**
   Similar to the constrained stability analysis, there are too many possible active constraint sets. Suppose that there are 10 output constraints, then, there are $2^{10} - 1 = 1023$ possible active output constraint sets. At the design stage, there is no way to know which one will be active. Therefore, the control design and analysis should consider every possible combination of active constraints. To prevent performance deterioration in the future, the controller tuning should be very conservative.

2. **Complicated Numerical Searching:**
   This new optimization is more complicated with mixed objective functions, since the $\mathcal{J}_2$ term also includes the future control moves. Efficient numerical algorithms for QP and/or LP can no longer applied to this problem.

3. **Stricter Stability Requirements:**
   If $\mathcal{J}_2$ uses a 2-norm definition, as in $\mathcal{J}_1$, these two terms can be combined together which is equivalent to changing the control weighting in the original MPC formulation (Oliveira & Biegler 1994). Since the weighting comes as a result of a combinatorial problem, it required that the original MPC system should be closed loop stable for a large range of control weighting.

A new strategy to handle output constraints is proposed in the next section.

7.5.3 Dynamic Constraint Softening using Prior Output Trajectory

With knowledge of the process model and past input/output behaviour, it is possible to make predictions of the process output. From the predicted output trajectory, it is possible to determine the extent of any output constraint violations. Therefore, an intuitive method can be used to put penalty weighting on any future output constraint violations.

Define

$$\mathcal{J} = \sum_{j=1}^{P}(\gamma_j + \psi_j)[y_{sp}(t+j) - y(t+j)]^2 \tag{7.16}$$

where γ_j is the standard output weighting terms discussed in Chapter 4 and ψ_j is defined as a function of the distance between the output constraint y_b and the future predicted output $y(t+j)$,

$$\begin{aligned}\psi_j &= \mathcal{F}\{y_b - y(t+j)\} \\ &= \mathcal{F}\{y_b - y_m(t+j) + A_j\Delta U\}\end{aligned} \tag{7.17}$$

Note that the unknown future control moves, ΔU, are included in the output weighting functions in Equation (7.17) which would make the control problem a nonlinear optimization problem (Feher & Erickson 1993). However, if only the model predictions, y_m, are applied, the weighting function can be simplified to

$$\psi_j = \mathcal{F}\{y_b - y_m(t+j)\} \tag{7.18}$$

with $\psi_j \to \infty$, if the j-th constraint is active. As shown in Figure 7.2, the weightings reflect the distances between the predicted outputs and their constraint limits.

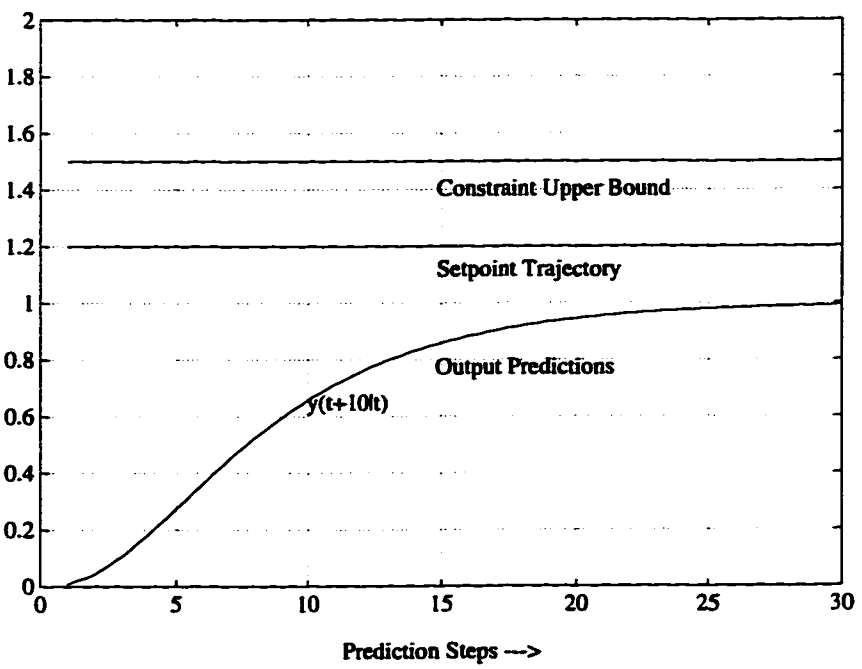

Figure 7.2:Process Predictions and Constraints

Three advantages of this new formulation are:

1. **Simple optimization**:

   The objective function is still quadratic so that many efficient numerical algorithms are available to find the control solution. Since only constraints on inputs are considered, an analytical solution may also be possible (Mutha 1990) which would significantly reduce the time for on-line optimization even further.

2. **Continuous control mode changes:**

   With the variable weighting, the predictive controller can still be represented as a state feedback controller but with a variable gain vector. If the ψ is defined as a continuous function of the distance, the gain elements are continuous functions as well. Therefore, the linear controller is smoothly switching from one structure to another instead of abrupt switching as in the original MPC formulation.

3. **Dynamic consideration of the constraints:**

   The new formulation considers the hard constraint boundaries as well as the distance from the current state to the constraint. The controller can therefore take early action to prevent future constraint violations. This would also reduce the possibility of infeasible solution or conflicting constraints.

The on-line implementation of the control optimization is the other major benefit from this new control formulation since it is the same as the unconstrained formulation. However, theoretically, there is a problem with the stability analysis. Because the output weighting are now a function of the state variables, they would introduce extra nonlinear state feedback into the controller. Generally speaking, it must be considered as a nonlinear controller for the closed loop stability analysis.

7.5.4 Stability Issue

Rewrite the (unconstrained) state feedback control gain for the MPC controller as

$$\begin{aligned} K_{mpc} &= \mathcal{F}(\lambda, M, P, \gamma, \psi) \\ &= C^T A^* \Phi_p \\ &= C^T [A^T(\gamma+\psi)A + \lambda I]^{-1} A^T(\gamma+\psi)\Phi_p \end{aligned} \tag{7.19}$$

Note that input constraints, whether on the incremental input or absolute input, do not change the open loop stability as discussed in the previous sections. Only the unconstrained solution, which is a state feedback controller, needs to be considered for the analysis of stability. Closed loop stability can be evaluated by the eigenvalues of the matrix $(\Phi - \theta K_{mpc})$.

REMARKS:

1. If the original *unconstrained* MPC system is stable for all $\gamma \in [0, \infty)$, then the variable weighting MPC is also stable. Obviously, it requires the MPC system to be tuned stable for a wide region of output weighting;

2. Assuming only one constraint is active, *i.e.* $\psi_j \rightarrow \infty$, the objective function becomes

$$\begin{aligned} \mathcal{J} &\rightarrow [y_{sp}(t+j) - y(t+j)]^2 \\ &= \max \|Y_{sp} - Y\|_2 \end{aligned}$$

which is similar to the "robust MPC" formulation (Campo & Morari 1987) and later extensions (Zheng & Morari 1995). The "Robust MPC" formulation does not try to minimize errors at all future horizons, but only the maximum error, *i.e.* worst-case optimization. Together with linear constraint equations, this formulation allows the integration of parameter uncertainties into the control calculation (Campo 1990). Therefore, it facilitates the robustness analysis of the closed loop control system.

A physical interpretation is when an output prediction approaches its constraint, the "robust MPC" controller would consider this output term only and make control moves to bring it back to the setpoint.

Use Ψ to represent the variable weighting matrix,

$$\Psi = \begin{bmatrix} \psi_1 & 0 & \cdots & 0 \\ 0 & \psi_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \psi_P \end{bmatrix}$$

with $\psi_j \in [0, \infty), j = 1, 2, \ldots, P$.

A **sufficient condition** for the closed loop stability of the variable weighting predictive control system is that all eigenvalues of $\Phi - \theta K_{mpc}(\psi = \infty)$ are within the unit circle for all individual ψ_j and their combinations.

For example, a typical 2-norm of the controller gain $\|K_{mpc}\|$ as a function of ψ is shown in Figure 7.3. Obviously, as the weighting increases, the controller gain increases and eventually converges to a constant. As a result, the original control system would become fast but could be destabilized (similar to reducing the prediction horizon to $P = 1$). If the stability of the closed loop system with $\psi = \infty$ can be ensured, there would not be any stability problem for the controller with low gain.

This sufficient condition provides an analytical tool to evaluate the stability at the control design stage. It is a combinatorial problem too but a much simpler one since output constraints are directly considered (instead of mapping into the input domain). In the output domain, all output constraints are considered as high/low limits only while in the input domain, they are linear functions. For example, a SISO system would have only one active output constraint at one time. That means if there are 10 output constraints, there are 10 possible structures of the Ψ matrix (reduced from 1023). This significantly reduces the computation effort required for the stability analysis.

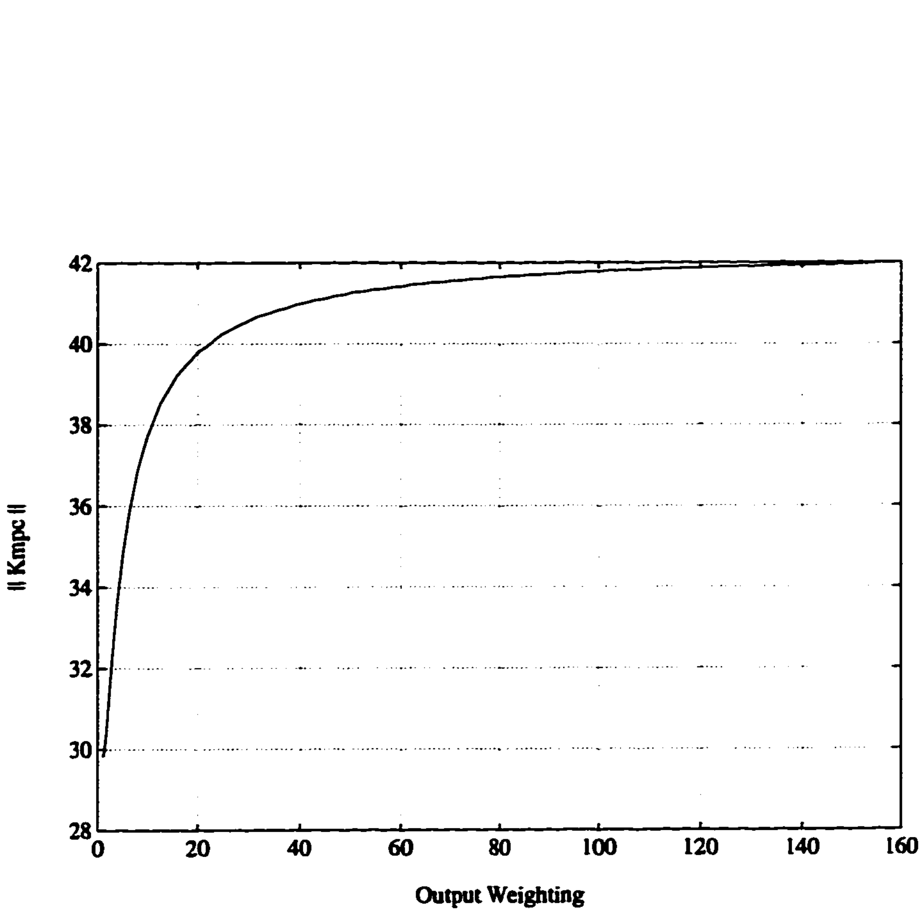

Figure 7.3:The Effect of Output Weighting on the State Feedback Gain

7.6 Conclusion

Model Predictive Control with hard constraints becomes a piece-wise linear state feedback controller so that the stability problem can be evaluated by the characteristics of the closed loop system matrix. Active *output* constraints introduce extra state feedback terms. A control system with active *input* constraints becomes an open loop system. Generally speaking, the stability analysis procedure is too complicated and time consuming to be carried out for practical applications.

A new stability result is developed for MPC with linear objective function and hard constraints. Its constrained stability can be guaranteed provided the corresponding MPC-QP formulation is unconstrained stable.

A new constraint handling method is proposed which softens output constraints and facilitates the stability analysis. The control calculation is also simplified by using the knowledge of the future output trajectory so that less numerical searching is required.

Chapter 8

Dual-Model Predictive Control with Dynamic Tuning

8.1 Introduction

The quadratic optimization objective function is widely used in modern control techniques ranging from LQC, GMV to Least Squares based identification algorithms. This is also true for model predictive controllers including DMC, IDCOM and GPC. One of the main reasons for the popularity of the quadratic objective function is that, for unconstrained situations, there exists an analytic solution which makes both analysis and computation very simple. In order to obtain good MPC performance, several parameters must be chosen properly. Among them are the prediction horizon(P), the control horizon(M), the control weighting(λ) and output weightings (including steady state weighting for open loop stable systems). It has been shown that the integer type parameters, P and M, have a large impact on the control performances (in Chapter 4) and robustness (in Chapter 6). Therefore, adjusting P and M is quite common for tuning MPC applications. There are, however, two problems associated with these two parameters:

1. Since they are integers, it is impossible to adjust the parameters smoothly and continuously. In particular, both the tracking performance and robustness are very sensitive to the control horizon(Qi & Fisher 1994). For example, in many applications, taking $M = 1$ makes the control system stable but sluggish, or conservative. On the other hand, using $M = 2$ results in fast response, and may cause instability or poor robustness.

2. When (auto) adjusting the control horizons, the structure of the dynamic matrix for the unconstrained solution must be changed and a matrix inversion (pseudo-inverse) is required. These commonly off-line calculation procedures need a great deal of computation especially for complex MIMO systems and may be impractical for on-line tuning.

The introduction of continuous (rather than integer) tuning parameters is desirable especially for on-line performance tuning. The commercial DMC package includes a set of 'equal concern parameters' for this purpose. In DMC, all prediction and control horizons are fixed and the 'equal concern parameters', (which appear to be equivalent to adjusting the output weightings), can be adjusted on-line by engineers or computer programs. The integer type control horizon can be simplified by assuming a linear relationship among control moves which continuously adjusts the controller. For example, the pole-placement based method is proposed to find the linear relationship (Peng, Fisher & Shah 1993). But it requires the solution of a nonlinear algebraic equation and is only suitable for designing *unconstrained* predictive controllers. In this chapter, an additional tuning parameter is defined which can be treated as a fractional horizon to replace the integer control horizon as a tuning parameter. The second problem, *i.e.* the computational load associated with modification and inversion of the dynamic matrix, can be solved by the idea of recursive least squares techniques.

Closed loop stability may become a problem if the controller parameters are changed on-line. It is well known that MPC is an optimal, performance-orientated control technology. Stability analysis is a post design procedure. With a variable linear control structure, *i.e.* on-line tuning, the stability of the closed loop system should be ensured before commissioning the control application.

The complete development of a new predictive control structure with an additional tuning parameter, the fractional control horizon, is discussed in this chapter. The effects of the fractional horizon on the nominal stability and the robustness are also proven. This new structure is then extended to more general situations as well as to *constrained* predictive control design.

8.2 Recursive Calculations for Increasing Horizons

The unconstrained MPC solution can be obtained by calculating the control trajectory, ΔU, that minimizes an appropriate performance index such as:

$$\mathcal{J} = \sum_{j=1}^{P}[y_{sp}(k+j|k) - y(k+j|k)]^2 + \sum_{j=1}^{M}\lambda[\Delta u(k+j-1)]^2 \tag{8.1}$$

This leads to the following least-squares control law:

$$(A^TA + \lambda I)\Delta U = A^TE \tag{8.2}$$

where the M-step future control action, ΔU, and P-element prediction error vector, E, are given by

$$\begin{aligned}\Delta U &= [\Delta u(k),\ \Delta u(k+1),\ \cdots,\ \Delta u(k+M-1)]^T \\ E &= Y_{sp} - \Phi_p X(k-1)\end{aligned}$$

The dynamic matrix A explicitly consists of the unit step response data as:

$$A = \begin{bmatrix} S_1 & 0 & 0 & \cdots & 0 \\ S_2 & S_1 & 0 & \cdots & 0 \\ S_3 & S_2 & S_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ S_M & S_{M-1} & S_{M-2} & \cdots & S_1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ S_P & S_{P-1} & S_{P-2} & \cdots & S_{P-M+1} \end{bmatrix}_{P\times M}$$

and

$$\Phi_p = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & \cdots & 0 \end{bmatrix}_{p\times(n+1)}$$

The calculation of future output predictions, $Y_m(k+i|k) = \Phi_p X(k)$ can be done by using process models in many different but equivalent ways (Li et al. 1989, Clarke & Mohtadi 1987, Qi & Fisher 1993), plus feedback/feedforward estimation techniques.

Obviously, the future control action is a *continuous* function of the parameter λ, but a *discrete* function of integer parameters, P and M.

8.2.1 The Effects of M on the Control Action

Assuming there is an extra control action Δu_{m+1} at step $m+1$, *i.e.* the control horizon $M = m+1$, the dynamic matrix A_{m+1} can be re-built by using the previous matrix A_m and a new coefficient vector x_{m+1} as:

$$A_{m+1} = [A_m, \;\; x_{m+1}]$$

where

$$x_{m+1} = [0, \;\cdots, \; 0, \; S_1, \;\cdots, \; S_{p-m}]^T_{(p\times 1)}$$

such that

$$\begin{aligned} A^T_{m+1}A_{m+1} &= \begin{bmatrix} A^T_m \\ x^T_{m+1} \end{bmatrix} \begin{bmatrix} A_m & x_{m+1} \end{bmatrix} \\ &= \begin{bmatrix} A^T_m A_m & A^T_m x_{m+1} \\ x^T_{m+1} A_m & x^T_{m+1} x_{m+1} \end{bmatrix} \\ A^T_{m+1}A_{m+1} + \Lambda &= \begin{bmatrix} A^T_m A_m + \lambda I & A^T_m x_{m+1} \\ x^T_{m+1} A_m & x^T_{m+1} x_{m+1} + \lambda_{m+1} \end{bmatrix} \end{aligned}$$

The least squares solution in Equation (8.2) can be expressed as:

$$\begin{bmatrix} A^T_m A_m + \lambda I & A^T_m x_{m+1} \\ x^T_{m+1} A_m & x^T_{m+1} x_{m+1} + \lambda_{m+1} \end{bmatrix} \begin{bmatrix} \Delta U_m \\ \Delta u_{m+1} \end{bmatrix} = \begin{bmatrix} A^T_m \\ x^T_{m+1} \end{bmatrix} E \qquad (8.3)$$

and after some algebraic manipulation, the solution can be obtained as:

$$
\begin{aligned}
\Delta U_m &= \Delta U_m^0 - G_1 x_{m+1}^T (E - A_m \Delta U_m^0) && (8.4)\\
\Delta u_{m+1} &= G_2 x_{m+1}^T (E - A_m \Delta U_m^0) && (8.5)\\
\textit{where} & \\
\Delta U_m^0 &= (A_m^T A_m + \lambda I)^{-1} A_m^T E \\
G_1 &= G_3 G_2 \\
G_2 &= (x_{m+1}^T x_{m+1} + \lambda_{m+1} - x_{m+1}^T A_m G_3)^{-1} \\
G_3 &= (A_m^T A_m + \lambda I)^{-1} A_m^T x_{m+1}
\end{aligned}
$$

Note that in both Equation (8.4) and (8.5) there is a term $(E - A_m \Delta U_0)$. Define $(E - A_m \Delta U_m^0)$ as the residual term after m-step future inputs calculated to minimize the prediction error E. The above expression can be interpreted as described in the following remarks.

REMARKS:

1. With an extra element in the control horizon, the new future control vector ΔU_m with $(M = m + 1)$, can be obtained by using the original $(M = m)$ control vector ΔU_m^0 plus a modification term (Equation (8.4)).

2. The effectiveness of increasing the control horizon can be evaluated by analyzing the residual term $(E - A_m \Delta U_m^0)$. From the point of view of system optimization, a large residual suggests increasing the control horizon until the residual becomes sufficiently small.

3. The recursive update of the future control vector involves only a new, lower dimensional matrix inverse, *i.e.* G_2, instead of an $(m+1)$ by $(m+1)$ matrix inverse. For SISO or MIMO systems, if only one control horizon is retuned from m to $m+1$, G_2 is a scalar. This is very useful for on-line adjustment of the controller parameters.

Note the following well known lemma about the block matrix inversion:

Assume the matrix

$$\Phi = \begin{bmatrix} A & B \\ B^T & D \end{bmatrix}$$

then

$$\Phi^{-1} = \begin{bmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \Phi_{22} \end{bmatrix}$$

where

$$
\begin{aligned}
\Phi_{11} &= A^{-1} + A^{-1}B(D - B^T A^{-1} B)^{-1} B^T A^{-1} \\
\Phi_{12} &= -A^{-1}B(D - B^T A^{-1} B)^{-1} \\
\Phi_{21} &= -(D - B^T A^{-1} B)^{-1} B^T A^{-1} \\
\Phi_{22} &= (D - B^T A^{-1} B)^{-1}
\end{aligned}
$$

a similar relationship between ΔU_m and ΔU_m^0 can also be obtained by defining

$$\begin{aligned} A &= A_m^T A_m + \lambda I \\ B &= A_m^T x_{m+1} \\ D &= x_{m+1}^T x_{m+1} + \lambda_{m+1} \end{aligned}$$

The detailed derivation is omitted here.

8.2.2 Fractional Horizons α

A linear interpolation can be calculated along the straight line $A-B$ in Figure 8.1 between the two control actions ΔU_m^0 (point A, with $M = m$) and ΔU_m (point B, with $M = m+1$). Defining a real parameter $\alpha, \alpha \in [0,1]$ and the control move as ΔU_m^α, leads to

$$\begin{aligned} \Delta U_m^\alpha &= \Delta U_m^0 + \alpha(\Delta U_m - \Delta U_m^0) \\ &= \Delta U_m^0 - \alpha K_m(E - A_m \Delta U_m^0) \end{aligned} \tag{8.6}$$

where $K_m = G_1 x_{m+1}^T$. Obviously, ΔU_m^α is vector of control moves that corresponds to the control horizon $M = m+\alpha$. Therefore, the continuously adjustable parameter, α, which has the same unit scale as the control horizon, can be used to improve control performance.

Note that a very useful extension of Equation (8.4) can be made to general predictive controllers with control horizons m and n as opposed to the special case of m and $m+1$. A linear interpolation between any two integer control horizons can be obtained with corresponds to a control horizon $m + \alpha(n-m)$ (line $A-C$ in Figure 8.1). The control move update would include a small sized matrix inversion instead of a scalar. All the stability results presented in the next section would still hold.

8.3 Stability and Robustness of MPC with A Fractional Control Horizon

The new control algorithm, expressed in terms of ΔU_m^α, is bounded by two conventional MPC controllers. Stability and robustness analysis can be extended to this algorithm too. The state feedback structure of this control algorithm is developed first as below.

8.3.1 State Feedback Control Form

In the state space formulation, the MPC controller can be represented by a state feedback controller. The controller gain vector K_{mpc} is a function of the tuning parameters λ, M, P and the output weighting terms in the control objective equation

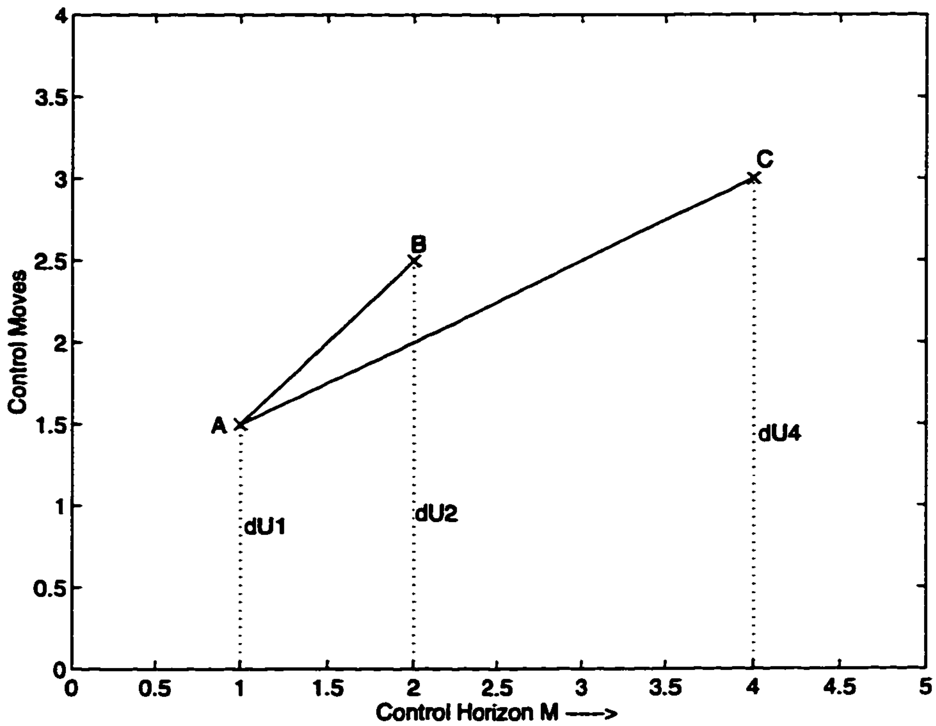

Figure 8.1:Control Actions and Interpolations

(8.1). The closed-loop stability of MPC systems can be analyzed by evaluating the eigenvalues of the closed loop matrix $\Phi - \theta K_{mpc}$.

Corresponding to the control calculation in Equation (8.6), it is easily verified that the controller gain becomes:

$$K^{\alpha}_{mpc} = K^{0}_{mpc} - \alpha(K^{1}_{mpc} - K^{0}_{mpc}) \tag{8.7}$$

where K^{0}_{mpc} is the controller gain with control horizon $M = m$, and K^{1}_{mpc} is the controller gain with control horizon $M = m + 1$.

Recalling the control gain changes in Figure 6.6, the norm of the control gain changes as a stair type function of the integer control horizon. The fractional horizon gives a smooth change to the control gain, *i.e.* like an interpolation line. Therefore, it is better suited for fine tuning the predictive controller. A simulation result is shown in Figure 8.2 where as α increases from 0 to 1, the output response moves from the $M = 1$ case to the $M = 2$ case, *i.e.* becomes faster.

8.3.2 Stability of MPC with A Fractional Horizon

In order to guarantee the closed-loop stability of the fractional horizon MPC controller, all eigenvalues of $(\Phi - \theta K^{\alpha}_{mpc})$ should be within the unit circle. If the value of α is given, K^{α}_{mpc} can be calculated and the closed loop stability can be easily evaluated. Further, a general closed-loop stability result of the fractional horizon MPC can be verified according to following lemma:

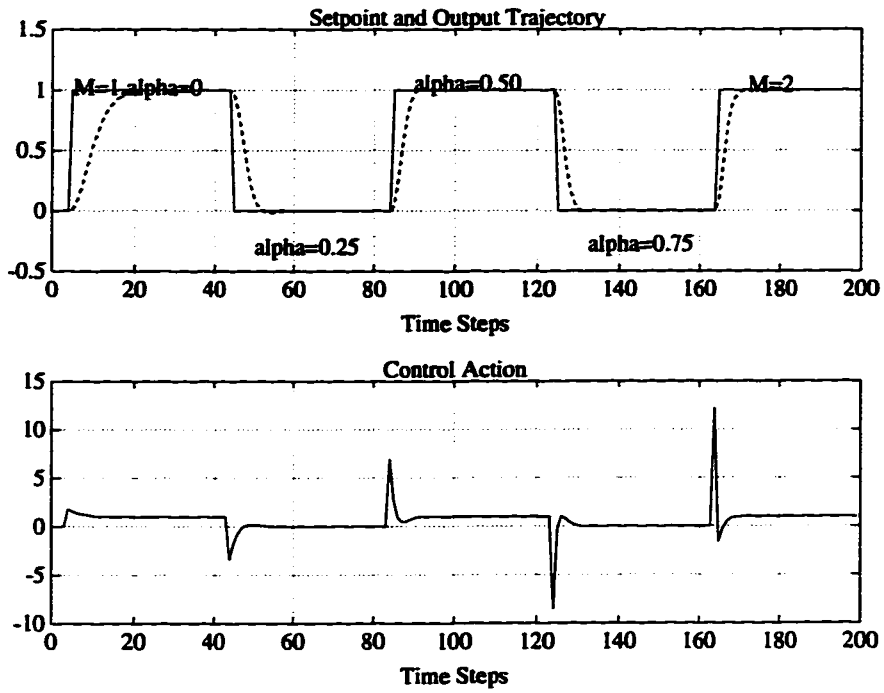

Figure 8.2:Control Response with α horizon

Lemma 8.1 *The closed-loop MPC system with fractional horizon $M+\alpha$ is stable for all $\alpha \in [0,1]$ if and only if the MPC systems are stable with (integer) control horizon M and $M+1$.*

Proof of necessity:

Since the parameter $\alpha \in [0,1]$, the necessity is trivial.

Proof of Sufficiency:

with horizon M, stable MPC $\longrightarrow \| \Phi - \theta K^0_{mpc} \|_2 < 1$.

with horizon $M+1$, stable MPC $\longrightarrow \| \Phi - \theta K^1_{mpc} \|_2 < 1$.

with horizon $M+\alpha$, the closed loop MPC

$$\begin{aligned} \| \Phi - \theta K^\alpha_{mpc} \|_2 &= \| \alpha(\Phi - \theta K^1_{mpc}) + (1-\alpha)(\Phi - \theta K^0_{mpc}) \|_2 \\ &\leq \alpha \| \Phi - \theta K^1_{mpc} \|_2 + (1-\alpha) \| (\Phi - \theta K^0_{mpc}) \|_2 \\ &< \alpha + (1-\alpha) \\ &< 1 \end{aligned}$$

□

If one of the predictive control structures is not stable, (usually the one with the larger control horizon), then a maximum allowable parameter α_{max} can be calculated by matrix perturbation methods.

Lemma 8.2 *If the MPC system is stable for $M = m$ but unstable for $M = m + 1$, then there exists an upper bound α_{max} such that the MPC system is stable for all $M = m + \alpha, \alpha \in [0, \alpha_{max}]$. Further, a sufficient condition gives α_{max} as:*

$$\alpha_{max} = \frac{1 - \| \Phi - \theta K^0_{mpc} \|_2}{\| \theta(K^1_{mpc} - K^0_{mpc}) \|_2}$$

Proof:

$$\begin{aligned} \| \Phi - \theta K^\alpha_{mpc} \|_2 &= \| (\Phi - \theta K^0_{mpc}) + \alpha\theta(K^1_{mpc} - K^0_{mpc})) \|_2 \\ &\leq \| \Phi - \theta K^0_{mpc} \|_2 + \alpha \| \theta(K^1_{mpc} - K^0_{mpc}) \|_2 \end{aligned}$$

with the requirement

$$\| \Phi - \theta K^\alpha_{mpc} \|_2 < 1$$

leads to

$$\alpha \leq \frac{1 - \| \Phi - \theta K^0_{mpc} \|_2}{\| \theta(K^1_{mpc} - K^0_{mpc}) \|_2}$$

therefore

$$\alpha_{max} = \frac{1 - \| \Phi - \theta K^0_{mpc} \|_2}{\| \theta(K^1_{mpc} - K^0_{mpc}) \|_2}$$

where by assumption, $K^1_{mpc} \neq K^0_{mpc}$. □

α_{max} can also be calculated by root locus methods which would give a sufficient and necessary condition. The knowledge of α_{max} can provide a guideline for on-line tuning the controller, *e.g.* stability margins.

8.3.3 Robust Stability of MPC with A Fractional Horizon

As discussed in Chapter 6, model uncertainties change both the system matrix Φ and the control matrix θ and hence the closed-loop system stability. The nominal system design should ensure that the actual closed loop system is stable against model uncertainties, $\Delta\Phi$ and $\Delta\theta$. In most applications, the smaller the control horizon, the more robust the system.

The parameter α can be used as an effective and convenient measure of compromise between performance and robustness. For example, for the choice of $M = 1$ and $M = 2$, an α-controller with $\alpha = 0.2$ gives much better robustness than $M = 2$ without the excessive decrease in control performance.

The effect of the fractional horizon α on the robustness of the control system can be evaluated the same way as the closed loop stability. A general result is given below without a detailed proof.

Lemma 8.3 *The closed-loop MPC system with fractional horizon $M + \alpha$ is* **robust** *stable for all $\alpha \in [0, 1]$ if and only if the MPC systems are* **robust** *stable with (integer) control horizons M and $M + 1$.*

8.4 Fractional horizon MPC with Hard Constraints

Even with hard constraints, the fractional horizon α concept can be used in the same way as before. However, there is no analytic relationship between the constrained solutions ΔU^0 and ΔU^1. The constrained optimization has to be solved twice to find constrained solutions, for ΔU^0 and for ΔU^1, within the same control interval. Then, an interpolation between these two control actions can be obtained using α. Since both ΔU^0 and ΔU^1 are constrained feasible, ΔU^α is located either inside the boundary (unconstrained) or on the boundary (constrained) due to the linear convex nature of the constraint equations. More complicated computations, *i.e.* solving constrained optimization twice, are required if the fractional horizon control is implemented in this way.

As discussed in Chapter 7, hard constraints would change the structure of the closed loop control system. In particular, output constraints may cause stability problems even though the nominal (unconstrained) control system is stable. The fractional control horizon, with good robustness and stability properties, can be used to adjust the control performance smoothly. However, it would certainly be a great advantage if the stability results could be extended to handle hard constraints.

The general idea is to change the controller structure by continuously adjusting the fractional horizon to avoid active constraints. Before further development, two presumptions should be pointed out followed by feasibility analysis.

ASSUMPTION 1:
Two stable controller structures can be obtained from the nominal unconstrained control design scheme which correspond to an aggressive (faster) and a slower control action respectively. The unconstrained controllers can therefore be referred to as state feedback controllers with gain vectors K^f_{mpc} and K^s_{mpc}.

ASSUMPTION 2:
One of the two controller structure, either the faster one or the slower one, has an unconstrained solution which satisfies all hard constraints, *i.e. constrained stable.*

Note that, the first assumption is trivial because of the unconstrained controller design. The second assumption, on the other hand, is not very obvious due to the complex nature of the time-varying boundary constraints.

Consider the control of open loop stable processes. Obviously, it is reasonable to assume that the original initial starting point is always at steady state and hence a feasible solution. Then, suppose no future control moves at all so that the process would stay at this operating point forever and there would be no violation of the constraints. That is to say, it is always a safe choice for one structure of the controller which is equivalent to choosing a control horizon equal to zero, *i.e.* $M = 0$, in the model predictive controller. Then, even during the transient period with receding horizon control calculations, an $M = 0$ controller is always safe and feasible because:

1. it is eventually stable because of the open loop stable processes;

2. it is always a feasible solution ensured by all previous control steps.

With those two selected **unconstrained** stable control structures, the fractional horizon controller can then be obtained which uses an interpolated control calculation in between, where

1. Without hard constraints, all control structures on the interpolation line are guaranteed (unconstrained) stable. This important line is then defined as Unconstrained Stable Line (USL).

2. With hard constraints, if one controller gives an unconstrained solution outside the feasible convex constraint region, USL can have one and only one intersection point with the constraint boundaries (for example, the point P in Figure 8.3).

3. Any control structure between the stable end and the intersection point is still on the USL and therefore guaranteed (constrained) stable;

4. The controller uses an unconstrained solution instead of constrained one which requires numerical searching. Note that a fixed structured controller would find the constrained solution at point C in Figure 8.3.

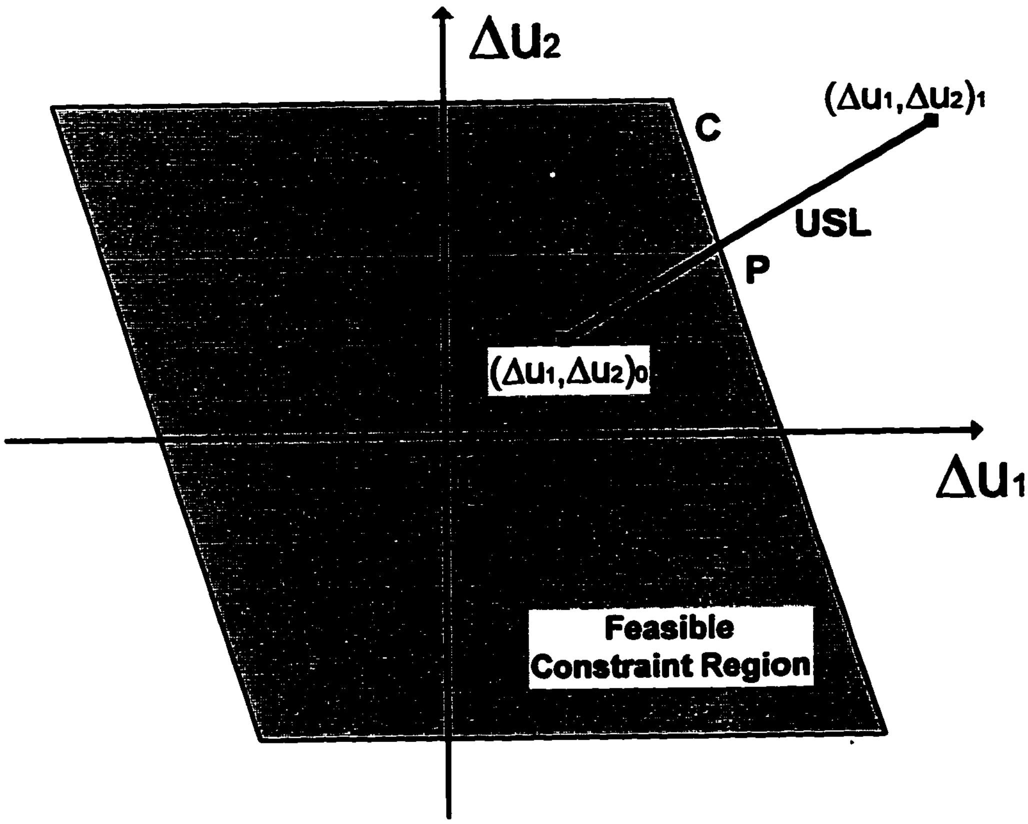

Figure 8.3:The Fractional Control Horizon and The Constraint Set

Therefore, stable control can be obtained that handles hard constraints by changing the control structure, as opposed to traditional fixed structure controllers. The new control calculation is simply an unconstrained MPC solution. The only other requirement is to find the intersection of the USL with the constraint boundaries which corresponds to solving a linear algebra equation set. As a result, at the design stage, the predictive controller can be chosen to achieve good nominal control properties without considering hard constraints. At the implementation stage, the on-line calculation of the adjustable control horizon can be used to handle constraints in a manner that keeps the stability property.

USL ALGORITHM:

- **Step 1:** Find two nominal unconstrained stable MPC controllers, K^s_{mpc} and K^f_{mpc};
- **Step 2:** Calculate the two (unconstrained) control actions, ΔU^0 and ΔU^1, corresponding to the two control structures;
- **Step 3:** Build the USL as

$$\Delta U^\alpha = \Delta U^0 + \alpha(\Delta U^1 - \Delta U^0)$$

- **Step 4:** If hard constraints are active, find α and ΔU^α by solving the USL and the constraint equations, *i.e.* $G\Delta U^\alpha = b$. Note that α can be chosen smaller than the value that satisfies the constraints.
- **Step 5:** Implement the calculated control move ΔU^α.

The USL algorithm is used at every control step. Even though two unconstrained MPC calculations and a linear algebraic equation must be solved on-line, it is much simpler than the numerical search algorithm of constrained quadratic optimization.

To illustrate the new constraint handling strategy, consider the following simple example. The process is

$$G(q^{-1}) = \frac{0.0014q^{-1} + 0.0054q^{-2}}{1 - 1.94q^{-1} + 0.9527q^{-2}}$$

Two unconstrained stable MPC controllers can be selected as:

- Controller 1: $P = 20, M = 1, \lambda = 0.0$ gives closed poles at $0.9288 \pm 0.0973j$;
- Controller 2: $P = 20, M = 3, \lambda = 0.0$ gives a closed pole at 0.2277;

The first controller is slow and the second one is much faster.

Now, add some constraints to this MPC control system.

$$|u_k| \leq 5, \quad and \quad |y_k| \leq 3$$

The unconstrained response from the slow controller in Figure 8.4 shows that the unconstrained system remains inside the feasible constraint region. But, the fast controller control action in Figure 8.7 (as 1-controller) violates the control constraint and therefore needs to be adjusted.

The fractional control horizon α is calculated such that the control move uses as much control action as possible for a fast control response but still remain within the constraint limits. Figure 8.6 shows the variations in α required to keep within the constraint limits. The control move varies between the $M = 1$ controller action, *i.e.* ($\alpha = 0$)-controller, and the $M = 3$ controller action, *i.e.* ($\alpha = 1$)-controller. This control system with a variable control horizon α gives a much fast response as shown in Figure 8.5. Figure 8.7 shows the changes of the constraint boundaries (the dashed lines for the upper/lower limits), the 0-controller moves (the dotted line) and the 1-controller moves (the dot-dashed line) for the control interval from 120 to 150. Obviously, the α-controller moves (the solid line) are switched from one to another. It uses smaller control action if it is close to the constraint and applies larger action if far away from the boundaries. It remains as a linear controller structure bounded by the stable 0-controller and the 1-controller so that the stability is ensured.

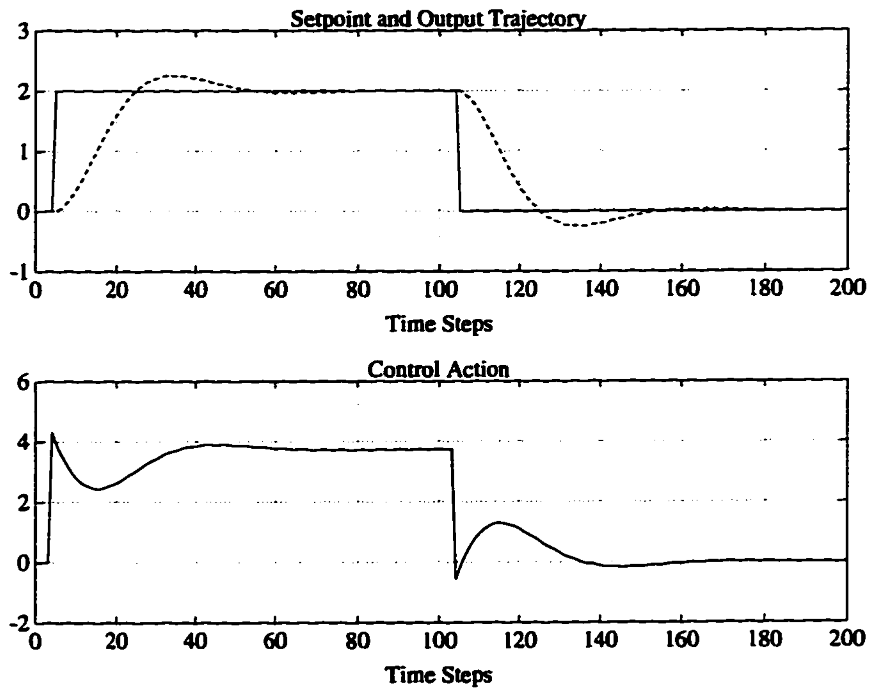

Figure 8.4:Unconstrained Control Response of $M = 1$ Controller

This example shows the great potential of using variable control structure to handle hard constraints. The obvious advantage comes from the guaranteed stability and the analytical solution even with hard constraints.

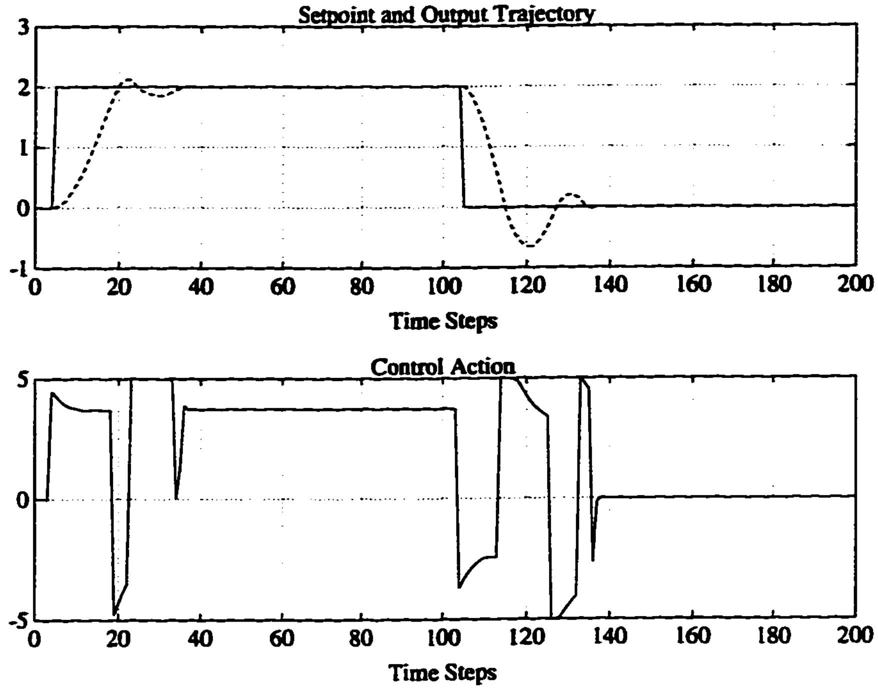

Figure 8.5:Constrained Control Response of Variable Horizon Controller

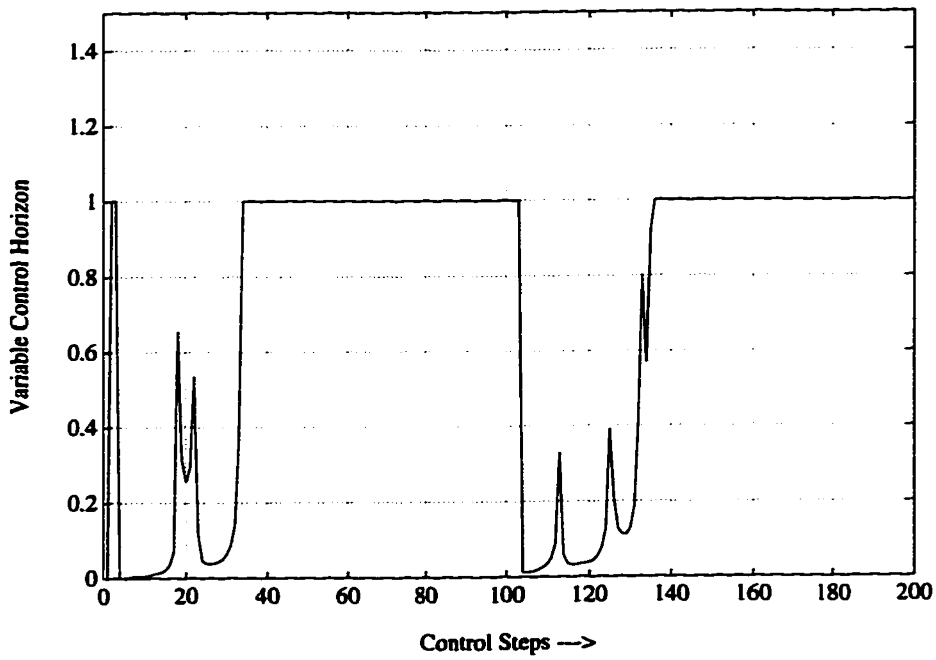

Figure 8.6:Variable Control Horizon α

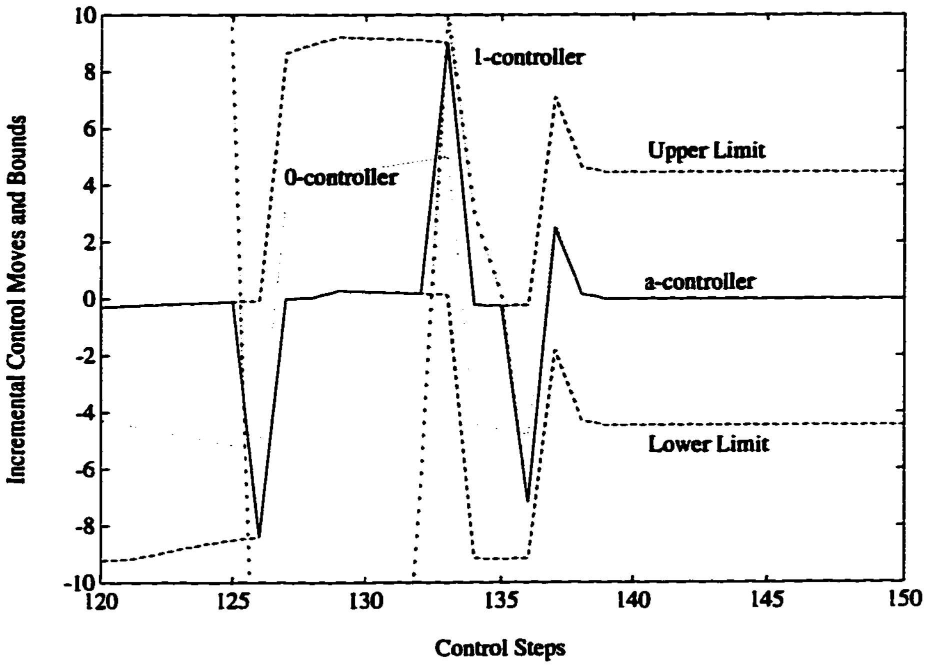

Figure 8.7:Constraint Limits and Control Moves

8.5 Dynamic Tuning Predictive Control: α-Controller

The tuning parameters of the structurally adjustable MPC controller can be changed based on information about model identification, and/or actual control performance. Examples of information are the residual between the model output and the actual process output, or some performance measures based on the actual overall closed loop control response.

8.5.1 Dynamic Tuning using A Variable Fractional Horizon

The fractional control horizon can be chosen a priori by the user at the same time as other control parameters. It can be also defined as a function of the system performance. For example, it can be automatically calculated depending on how good the output is tracking the desired trajectory, *i.e.* the predictive control structure can be changed dynamically.

Both nominal and robust performance analysis of predictive controllers have shown that they are very sensitive to the control horizon. For real applications, there is a trade-off between a fast response and a good robustness. The fractional control horizon technique can be used to adjust this kind of trade-off on-line.

Define the parameter α as a function of the difference between the model output y_k^m and the actual output measurement y_k over some period of time, *e.g.* the residual term:

$$d_k = y_k - y_k^m$$

The residual comes from the MPM, process disturbances and noise. Obviously, if a large residual exists, the controller should be tuned for good robustness (which usually means slow response). If the process can be described well by the model (*i.e.* small residual), fast control response can be achieved (with smaller robustness margin).

Therefore, to satisfy the two boundary conditions for the α,

$$\alpha_k|_{d_k=0} = 1, \quad \alpha_k|_{d_k=\pm\infty} = 0$$

For example, a nonlinear function such as the following can be used

$$\alpha_k = 1 - \frac{2}{\pi}\arctan|d_k^{\eta}| \tag{8.8}$$

The parameter $\eta, 0 \leq \eta \leq \infty$, is used to adjust the sensitivity of α to the residual term d_k as shown in Figure 8.8.

Obviously, this arrangement for α gives:

- a large control horizon, *i.e.* $\alpha \rightarrow 1$, if there is little noise/disturbance, model-plant-mismatch, etc;
- a small control horizon, *i.e.* $\alpha \rightarrow 0$, if there is large noise/disturbance, model-plant-mismatch, etc;
- an adjustable response speed for rejecting the existing residual d_k. For example, with $\eta = 0.5$, the value of α decreases rapidly for even a small value of d_k, while with $\eta = 5$, there is a dead-zone effect for small residuals.

Alternatively, a weighted sum of the residuals over some user-specified period of past time could be used and setpoint changes could be accommodated by considering deviations from a practical desired setpoint trajectory.

It is important to note that the "dynamic tuning" using Equation (8.8), or any other means of determining α, results in absolute stability of the predictive control if both boundary cases are stable.

To illustrate the advantages of dynamic tuning, consider the following example with gain mismatch added (from 1.0 to 3.0) at step 124. The variable fractional horizon is reduced (where $\eta = 0.2$) as soon as there is residual caused by the MPM (Figure 8.9). As a result, robust control performance is obtained as shown in Figure 8.10.

8.5.2 Closed Loop Performance Based Method

Using the fractional control horizon α, an interesting result can be obtained in terms of the dominant closed loop poles.

Lemma 8.4 *If $\mathcal{SR}^{\alpha}$ is the spectral radius of the closed-loop MPC system with fractional horizon $M + \alpha$, then*

$$\mathcal{SR}^{\alpha} \leq \mathcal{SR}^{1} + \alpha(\mathcal{SR}^{0} - \mathcal{SR}^{1}), \quad \alpha \in [0, 1]$$

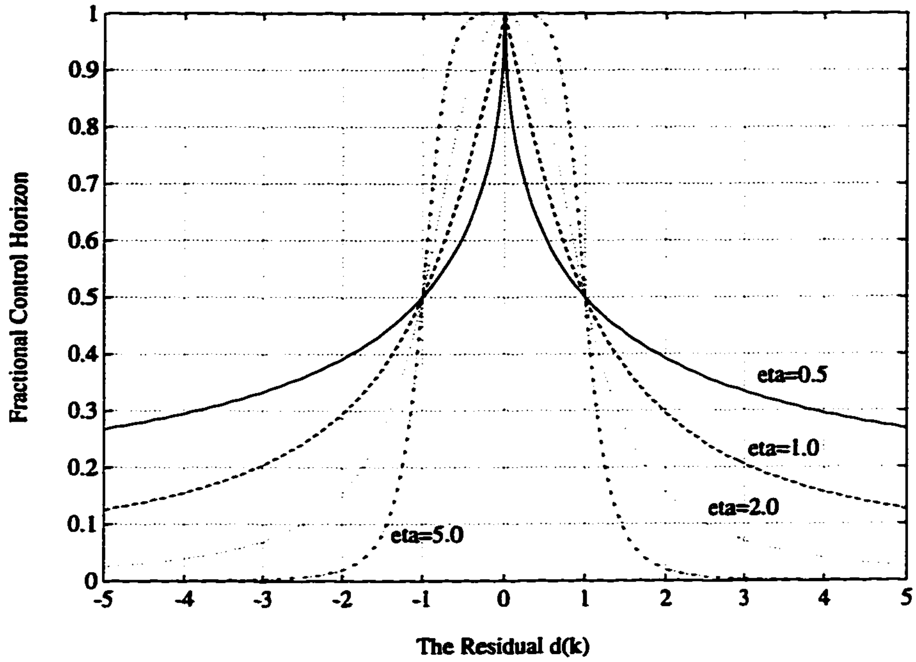

Figure 8.8:The Fractional Horizon *vs* Residual

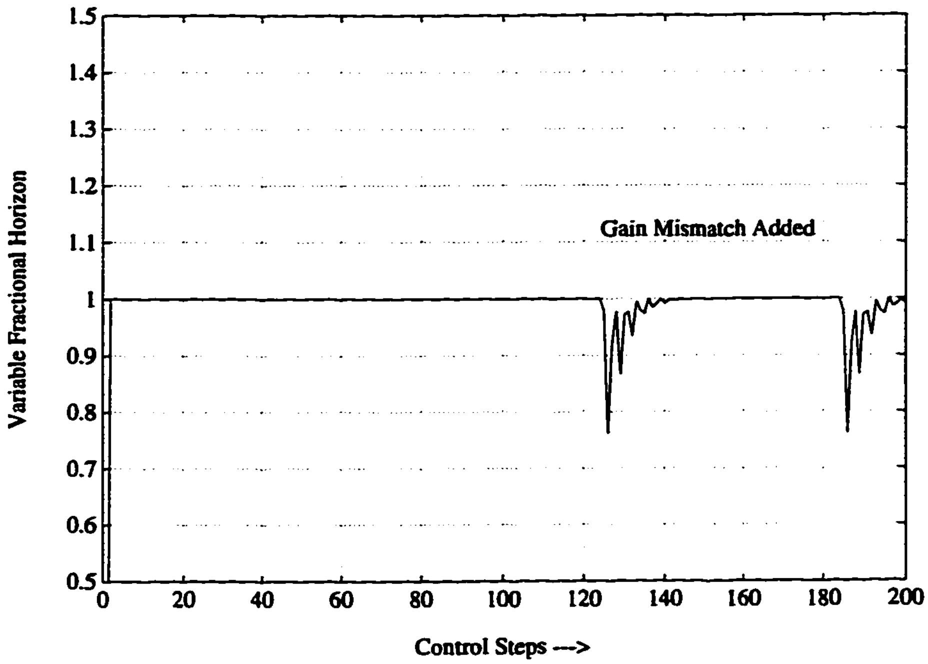

Figure 8.9:The Fractional Horizon Adjustment

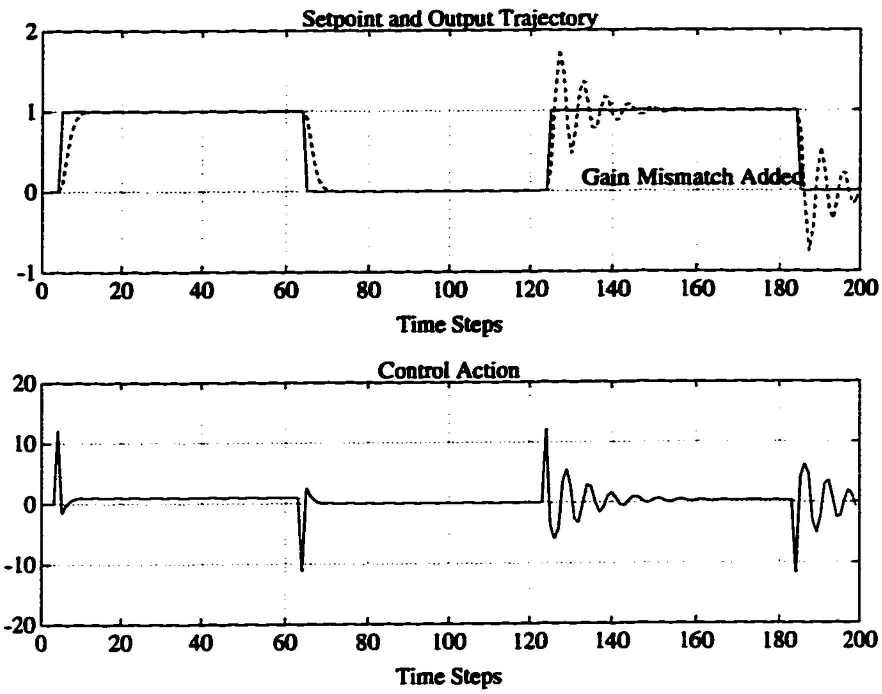

Figure 8.10:The Control Response Using Variable Fractional Horizon

Proof:

With horizon $M + \alpha$, the closed loop MPC

$$
\begin{aligned}
\mathcal{SR}^{\alpha} &= \| \Phi - K^{\alpha}_{mpc}\theta \|_2 \\
&= \| \alpha(\Phi - K^{1}_{mpc}\theta) + (1-\alpha)(\Phi - K^{0}_{mpc}\theta) \|_2 \\
&\leq \alpha \| \Phi - K^{1}_{mpc}\theta \|_2 + (1-\alpha) \| (\Phi - K^{0}_{mpc}\theta) \|_2 \\
&\leq \alpha \mathcal{SR}^0 + (1-\alpha)\mathcal{SR}^1 \\
&= \mathcal{SR}^1 + \alpha(\mathcal{SR}^0 - \mathcal{SR}^1)
\end{aligned}
$$

□

For example, if the fast controller has dominant pole at $p = 0.2$ and the slower controller has $p = 0.9$, then fractional horizon MPC with $\alpha = 0.5$ would have a dominant pole around $p = 0.2 + 0.5(0.9 - 0.2) = 0.55$.

A useful application of this Lemma is for on-line tuning of MPC control performance. After the MPC action is implemented on the real process, the closed loop control performance can be measured in terms of step response, poles or model parameters. Then, after two or more trials, α can be used to direct the new controller toward better performance. Both interpolation and extrapolation can be used for this purpose. Even though traditional MPC parameters such as λ can be used here, their effect on the closed loop performance (*i.e.* pole locations) is not as obvious as with α.

Note that even though the α-controller was originally developed as a fractional control horizon, it is not limited to the MPC controllers with the same λ, P only. It *can be extended to more general controller formats. The two bounding controllers can use MPC control structures with totally different tuning parameters. As long as the resulting controllers are stable, the concept of α-tuning can be applied. More generally, the two controllers can be any type of linear controller such as deadbeat, mean-level, PID control algorithms.*

8.6 Conclusions

A recursive calculation formula for control action was developed which simplifies the on-line control calculation. A continuous (real rather than integer) tuning parameter, α, is introduced to fine tune the predictive control such that best trade-off between performance and robustness can be achieved. The new predictive controller with a fractional horizon includes good stability, robustness and/or performance properties.

Another significant application of the fractional horizon control is to use a variable structure controller to handle hard constraints, or for automatic, on-line adjustment of the control performance. With the new control design, not only do the control properties such as closed loop stability remain valid, but the constrained solution is also simplified.

As an extension, an α-controller structure can be further used to handle more general situations. The new controller gives the flexibility required for tuning the control system on-line. The two bounding controller forms can be any stable linear control algorithms.

Chapter 9

Conclusions

The new predictive control scheme, DMPC, facilitates practical applications by integrating (1) state space recursive prediction, (2) predictive control-relevant parameter estimation, (3) robust disturbance handling via feedforward and feedback, and (4) dynamic tuning. Design and analysis techniques based on state space theory result in straightforward, effective tuning and robust control performance. Major results have been obtained in five areas: process modelling and identification, nominal control design, robustness analysis, constrained stability and dynamic control tuning.

1. **Process Modelling and Identification**

   The dual-model formulation developed in this thesis combines the conventional FSR and DARMA models into a compact state space format and has the advantages of both models, *i.e.* it is applicable to general open loop stable and unstable processes and the high dimension associated with FSR model (*e.g.* DMC) is reduced significantly. State space control theory has been applied directly to facilitate and improve predictive control design:

   - Controllability analysis shows that future output predictions used in the optimization index of predictive control are not equal weighted. Some predictions such as the longest prediction are heavily emphasized while, on the other hand, some linear combinations of the predictions are close to being out-of-control.
   - The parameter identification algorithm based on extended Kalman filter theory gives unbiased, optimal parameter estimation even with colored disturbance dynamics. Simultaneously estimating both states and parameters, the identification scheme results in a set of parameters which is tailored to the requirements of predictive control, *i.e.* predictive control-relevant identification. Model uncertainties in a parametric form are also obtained which are useful for the robustness analysis of model predictive control.
   - The expansion of the state space structure to handle multi-inputs and multi-output processes is straightforward. Therefore, the state space procedures used in this thesis including prediction, feedback observer, model

parameter identification, analysis, control tuning etc. are applicable to MIMO system control with almost no change.

2. **Nominal Control Design**

   After obtaining the process model coefficients, a nominal controller is designed by selecting the tuning parameters inherent in Model Predictive Control. Both the servo and regulatory performance of the predictive controller strongly depend on the tuning parameters, but are treated differently in this thesis.

   - **MPC Servo Controller**

     The effects of conventional MPC tuning parameters are determined and then used to improve the condition of the dynamic matrix, which consequently improves the robustness of the control system. For example, reducing the control horizon and/or adding a control move penalty results in better conditioning of the dynamic matrix. A new method based on matrix decomposition is developed to determine values for these intuitive but non-unique integer type tuning parameters such as the control horizon. The best combination of tuning parameters is then chosen to implement the predictive controller.

   - **MPC Predictive Feedforward Controller**

     A predictive feedforward control scheme is developed to improve the disturbance rejection performance. Similar to the future control profile concept, a disturbance profile is introduced in order to eliminate the effect of disturbances on the future output variables. Ideally, the disturbance can be rejected perfectly by feedforward control only but feedback is necessary in practice. Alternative methods for the controller design are also developed.

   - **Output Feedback Observer**

     Optimal state estimation (*i.e.* output prediction) is obtained by applying observer algorithms such as the Kalman Filter method to the dual-model formulation. Instead of solving a high dimensional Riccati equation, a much simpler, pole placement method can be used for designing the observer because of the particular structure of the dual-model formulation. Then, the effects of feedback modification techniques, the feedback horizon and the rotating factor β, on the state convergence are discussed. The proposed β-observer which is optimal for some specific but popular disturbance structures, provides a convenient, independent second degree of freedom or tuning parameter to adjust regulatory performance.

3. **Nominal and Robust Performance Analysis**

   The stability requirements for a predictive controller based on a nominal model and model uncertainties are determined in the state space domain where the predictive controller is equivalent to a type of state feedback controller. A general robust stability criterion for DMPC is obtained using matrix perturbation theory. With the special parameterization of the dual model representation, the conservativeness of the robust controller design is reduced. The effects of MPC tuning parameters are investigated and guidelines are obtained for robust predictive control design. It is also shown that the conservativeness of the sufficient criterion is sometimes big enough to give misleading results. For example, despite conventional tuning guidelines which say the opposite, increasing the output prediction horizon can de-stabilize and increasing the control horizon can stabilize an MPC system in the presence of model-plant-mismatch.

4. **Constrained Stability Analysis and Design**

   The conventional way to handle hard constraints in predictive control results in a numerical solution and undetermined stability for active output constraints. The stability analysis of constrained MPC is reformulated to show the effects and difficulties caused by active constraints. For MPC with a conventional quadratic objective function (MPC-QP), active input constraints make the control system become an open loop system. Active output constraints change the state feedback structure of the control system and therefore may cause instability. The constrained stability analysis method is mathematically sound but becomes a combinatorial problem. Since there are too many possible active constraint sets, the constrained stability analysis is impractical for most real control applications.

   For MPC with a linear objective function (MPC-LP), constrained stability results have also been developed in this thesis. Rather than evaluating every possible constrained control structure, a very simple stability criterion is developed which shows that active output constraints do not cause stability problems if the corresponding *unconstrained* MPC-QP formulation is stable. Since unconstrained stability can be evaluated easily within the state feedback control framework, this constrained stability criterion for MPC-LP is very convenient and useful for practical applications.

   Two improvements to handle hard constraints are suggested in this thesis. One uses a 'soft constraint' concept to replace the hard constraints on process outputs. With time-varying output penalty terms in the optimization index, the optimal solution is obtained analytically. Possible non-feasibility problems due to disturbances and MPM are also avoided. This proposed method also simplifies the stability analysis. Another alternative using the proposed α-controller is very promising for practical applications. The α-controller is adjusted automatically such that hard constraint violations are always avoided and the constrained control stability is absolutely guaranteed.

5. **Dynamic Tuning**

   For smooth on-line tuning, two tuning knobs, α and β, are introduced into the proposed implementation of predictive control. The α-controller is able to fine tune the servo control performance for the nominal as well as possible model-plant-mismatch situations. The single parameter α adjusted directly based on on-line available information can guarantee closed loop control stability. The

β-observer is used to tune the feedback design. Based on the quality of process measurements, the parameter β is used to adjust the speed of state estimation and disturbance rejection.

The biggest advantage of using these two parameters come from the fact that they are explicitly related to the pole locations. As shown in this thesis, α linearly shifts the dominant pole location of the controller and β shifts the pole locations of the state observer. The combined function of α and β tuning results in a user-determined tradeoff between robustness and servo/regulatory performances which can be implemented on-line using current information.

Even though the results in this thesis are obtained using a dual model representation, most of them are also applicable to other predictive control algorithms. For example, the α-controller and β-observer concepts can also be applied to the popular DMC and GPC schemes. Overall, the new predictive control scheme, DMPC, provides enhanced functionality and more flexibility for practical industrial applications.

Future extensions of this research work could include

- **DMEKF Applications in MIMO Systems**

  In principle, the DMEKF algorithm can be extended to MIMO process identification without technical problems. As shown by a simple SISO process in this thesis, this algorithm gives predictive control-relevant estimation of model parameters, and an integrated way of balancing the requirements of state estimation and parameter estimation. Applications of this algorithm to handle high order SISO and MIMO control of real industrial processes should be performed. In particular, the choice of the weighting matrix in the extended Kalman filter algorithm should be examined further.

- **Less Conservative Robustness Bounds for MIMO Systems**

  The use of the dual-model description helps obtain simpler and less conservative robust stability conditions than 'general robustness analyses'. Because of the state space form used in this thesis, the robustness analysis can be easily extended to MIMO processes. However, the sufficient conditions are still too conservative. To further reduce the conservativeness, new theoretical developments in matrix perturbation theory (*e.g.* internal matrices theorems) are required.

- **Constrained Robust Stability**

  The stability of the nominal MPC system is affected by both model uncertainties and (active) hard constraints. While the individual effect of either model uncertainties or constraints on the system stability is examined in Chapter 6 or Chapter 7, the closed loop stability under both model uncertainties and constraints is a much more complicated theoretical problem. At this moment, there does not appear to be a practical, theoretical approach to solve this problem. However, constraint avoidance methods using the soft constraint concept or the constrained α-controller structure are promising for practical applications. For example, adjusting the controller structure (instead of using fixed structure conventional MPC) via the α parameter results in excellent stability properties. More efficient numerical procedures would improve the practicality of this approach.

- **On-line Tuning α and β for MIMO Processes**

  For MIMO processes, the multivariable controller can be tuned on-line using α and β. The simplest method is to use only one α (or β) to modify the state feedback gain matrix K_{mpc} (or the observer matrix K). For example, the controller gain matrix becomes

$$K_{mpc}^{\alpha} = K_{mpc}^{0} + \alpha(K_{mpc}^{1} - K_{mpc}^{0})$$

  When only one parameter is adjusted, all theoretical results developed in this thesis can be directly applied. But, the dynamic performance of each input/output pair can not be tuned independently using only one tuning parameter. Therefore, during the nominal MPC design, conventional MPC parameters should be selected carefully to determine K_{mpc}^{0} and K_{mpc}^{1}.

  Another method is to use different tuning parameters (α or β) for different I/O pairs. For example, for a 2×2 process, the on-line α-tuning parameter becomes

$$\alpha = \begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{bmatrix}$$

  Each channel could be tuned independently by using the appropriate α_{ij}. The properties of α-controller, *e.g.* stability, robust stability *etc.*, may need to be extended for this more general formulation.

Bibliography

Asbjornsen, O. A. (1988), Stability and robustness of the DMC algorithm as compared with traditional ARMA model, *in* 'Proceedings of 1988 American Control Conference', Atlanta, U.S.A., pp. 278 – 283.

Balchen, J. G., Ljungquist, D. & Strand, S. (1992), 'State-space predictive control', *Chemical Engineering Science* **47**(4), 787–807.

Banerjee, P. & Shah, S. L. (1992), Tuning guidelines for robust generalized predictive control, *in* '1992 IEEE CDC', pp. 3233–3234.

Banerjee, P. & Shah, S. L. (1995), 'The role of signal processing methods in the robust design of predictive control', *Automatica* **31**(5), 681–695.

Bitmead, R. R., Gevers, M. & Wertz, V. (1990), *Adaptive Optimal Control: the thinking man's GPC*, Prentice Hall.

Campo, P. J. (1990), Studies in Robust Control of Systems Subject to Constraints, PhD thesis, California Institute of Technology.

Campo, P. J. & Morari, M. (1987), Robust model predictive control problems, *in* 'Proc. 1987 ACC, Minneapolis, MN', pp. 1021–1025.

Clarke, D. W. (1991), Adaptive generalized predictive control, *in* 'Chemical Process Control IV', pp. 419–444.

Clarke, D. W. & Mohtadi, C. (1987), 'Generalized predictive control – part I. the basic algorithm. part II. extensions and interpretations', *Automatica* **23**(2), 137 – 160.

Cutler, C. R. & Ramaker, B. L. (1980), Dynamic matrix control — a computer control algorithm, *in* 'Proceedings of the 1980 Joint Automatic Control Conference', San Francisco, pp. WP5–B.

Cutler, C. R. & Yocum, F. H. (1991), Experience with the DMC inverse for identification, *in* A. Y. & W. H. Ray, eds, 'Chemical Process Control - CPC IV, Fourth International Conference on Chemical Process Control', Elsevier, Amsterdam, pp. 297–317.

Dickman, A. (1987), 'On the robustness of multivariable linear feedback state-space representation', *IEEE Trans. Automatic Control* **32**(5), 407–410.

Feher, J. D. & Erickson, K. T. (1993), Solving the model predictive control problem with soft constraints, *in* 'Proceedings of the American Control Conference', San Francisco, California, pp. 377–378.

Finn, C. K., Wahlberg, B. & Ydstie, B. E. (1993), 'Constrained predictive control using orthogonal expansions', *AICHE Journal* **39**(11), 1810–1826.

Friedland, B. (1986), *Control System Design: an introduction to state-space methods*, McGraw-Hill, New York.

Froisy, J. B. (1994), 'Model predictive control: Past, present and future', *ISA Trans.* **33**, 235–243.

Garcia, C. E. & Morshedi, A. M. (1986), 'Quadratic programming solution of dynamic matrix control (QDMC)', *Chem. Eng. Comm.* **46**, 73–87.

Genceli, H. & Nikolaou, M. (1993), 'Robust stability analysis of constrained l_1-norm model predictive control', *AICHE Journal* **39**(12), 1954–1965.

Golub, G. H. & Van-Loan, C. F. (1989), *Matrix Computations*, 2nd edn, The Johns Hopkins University Press, Baltimore, MD.

Gudi, R. D., Shah, S. L. & Gray, M. R. (1994), 'Multirate state and parameter estimation in an antibiotic fermentation with delayed measurements', *Biotechnology and Bioengineering* **44**(11), 1271–1278.

Guidorzi, R. P. (1981), 'Invariant and canonical forms for systems structural and parameter identification', *Automatica* **17**(1), 117–133.

Han, H. S. & Lee, J. G. (1994), 'Necessary and sufficient conditions for stability of time-varying discrete interval matrices', *Int J Control* **59**(4), 1021–1029.

Ishikawa, A., Baba, T., Miki, T., Ochi, H. & Minter, B. J. (1995), Large scale multivariable controllers for an ammonia plant, *in* 'AICHE Ammonia Safety Symp.', Tuscon, AZ, USA, Sept. 18-20.

Juang, Y. T., Kuo, T. S. & Hsu, C. F. (1986), Stability robustness analysis for state space models, *in* 'Proc. 1986 CDC', Athens, Greece, pp. 745–750.

Keel, L. H. & Bhattacharyya, S. P. (1995), 'Robust stability of interval matrices: A computational approach', *Int. J. Control* **62**(6), 1491–1506.

Kolla, S. R., Yedavalli, R. K. & Farison, J. B. (1989), 'Robust stability bounds on time-varying perturbations for state-space models of linear discrete-time systems', *Int. J. Control* **50**(1), 151–159.

Kwok, K.-Y. & Shah, S. L. (1994), 'Long-range predictive control with a terminal matching condition', *Chemical Engineering Science* **49**(9), 1287–1300.

Larimore, W. E. (1990), Canonical variate analysis in identification, filtering, and adaptive control, *in* 'Proceedings of the 29th Conference on Decision and Control', Honolulu, Hawaii, pp. 596–604.

Lee, J. H., Morari, M. & Garcia, C. E. (1993), 'State-space interpretation of model predictive control', *Automatica* **30**(4), 707–717.

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

Li, W. C. & Biegler, L. T. (1989), 'Multistep, newton-type control strategies for constrained nonlinear processes', *Chem. Eng. Res. Des.* **67**(6), 562 – 577.

Lim, K. Y. (1988), Multivariable Optimal Constrained Control Algorithm (MOCCA), Master's thesis, University of Alberta.

Ljung, L. (1979), 'Asymptotic behaviour of the extended Kalman filter as a parameter estimator for linear systems', *IEEE Trans. on Automatic Control* **AC-24**(1), 36–50.

Matlab (1989), *386-MATLAB for 80386 MS-DOS Personal Computer: Users's Guide*, The Math Works, Inc.

McIntosh, A. R., Shah, S. L. & Fisher, D. G. (1991), 'Analysis and tuning of adaptive generalized predictive control', *Can. J. of Chem. Engng.* **69**, 97–109.

Morari, M. & Lee, J. H. (1991), Model predictive control: the good, the bad, and the ugly, *in* 'Chemical Process Control IV', pp. 419–444.

Morari, M., Garcia, C. E. & Prett, D. M. (1989), Model predictive control: Theory and practice, *in* 'Proc. IFAC Workshop on Model Based Process Control', Pergamon Press, Oxford. A later version of this paper can be also found in Automatica Vol.23 No.5 pp 335-348.

Morshedi, A. M., Cutler, C. R. & Skrovanek, T. A. (1985), Optimal solution of dynamic matrix control with linear programming techniques (LDMC), *in* 'American Control Conference', pp. 199–208.

Muske, K. R. & Rawlings, J. B. (1993), 'Model predictive control with linear models', *AICHE Journal* **39**(2), 262–287.

Mutha, R. K. (1990), Constrained long range predictive control, Master's thesis, University of Alberta.

Navratil, J. P., Lim, K. Y. & Fisher, D. G. (1988), Feedback prediction options in model predictive control systems, *in* 'Proc. IFAC Int. Workshop on Model Based Process Control', p. 6.

Niu, S. & Fisher, D. G. (1994), Multiple model least squares estimation method, *in* 'Proc. 1994 American Control Conference', Maryland, USA, pp. 2231–2235.

Niu, S., Fisher, D. G. & Xiao, D. (1992*a*), 'An augmented UD identification algorithm', *Int J Control* **56**(1), 193–211.

Niu, X., Abreu-Garcia, J. A. D. & Yaz, E. (1992*b*), 'Improved bounds for linear discrete-time systems with structured perturbations', *IEEE Trans. Automatic Control* **37**(8), 1170–1173.

Oliveira, N. M. C. & Biegler, L. T. (1994), 'Constraint handling and stability properties of model-predictive control', *AICHE Journal* **40**(7), 1138–1155.

Patel, R. V., Todd, M. & Sridhar, B. (1977), 'Robustness of linear quadratic state feedback design', *IEEE Trans. Automatic Control* **AC-22**, 945–949.

Peng, L., Fisher, D. G. & Shah, S. L. (1993), On-line tuning of GPC using a pole placement criterion, *in* 'Proc. 1993 American Control Conference', San Francisco, California, USA, pp. 791–795.

Qi, K. Z. & Fisher, D. G. (1993), Model predictive control for open-loop unstable processes, *in* 'Proc. 1993 American Control Conference', San Francisco, California, USA, pp. 796–800.

Qi, K. Z. & Fisher, D. G. (1994), Robust stability analysis of model predictive control, *in* 'Proc. 1994 American Control Conference', Maryland, USA, pp. 3258–3262.

Qin, J. S. & Badgwell, T. A. (1996), An overview of industrial model predictive control technology, *in* 'Proc. Chemical Process Control V', Athens, Greece.

Qiu, L. & Davison, E. L. (1986), New perturbation bounds for the robust stability of linear state space models, *in* 'Proc. 25th Conf. On Decision and Control', Athens, Greece, pp. 751–755.

Qu, Z. & Dorsey, J. F. (1990), Stability robustness of discrete systems with perturbations in state equation, *in* 'Proc. 1990 American Control Conference', San Diego, California, USA, pp. 3054–3057.

Rawlings, J. B. & Muske, K. R. (1993), 'The stability of constrained receding horizon control', *IEEE Trans. Auto. Cont.* **38**(10), 1512–1516.

Ricker, N. L. (1991), Model-predictive control: state of the art, *in* 'Chemical Process Control IV', pp. P–67.

Rivera, D. E. & Morari, M. (1986), 'Internal Model Control: 4. PID controller design', *Ind. Eng. Chem. Process Des. Dev.* **25**, 252–265.

Rossiter, J. A., Kouvaritakis, B. & Gossner, J. R. (1996), 'Guaranteeing feasibility in constrained stable generalized predictive control', *IEE Proc.-Control Theory Appl.* **143**(5), 463–469.

Saudagar, M. (1995), Unified Model Predictive Control, PhD thesis, University of Alberta.

Seborg, D. E., Edgar, T. F. & Mellichamp, D. A. (1989), *Process Dynamics and Control*, John Wiley & Sons, Inc.

Shah, S. L. & Cluett, W. R. (1991), 'Recursive least-squares based estimation schemes for self-tuning control', *The Canadian Journal of Chemical Engineering* **69**, 89 – 96.

Shook, D. S., Mohtadi, C. & Shah, S. L. (1991), 'Identification for long range predictive control', *IEE Proc. D* **138**(1), 75 – 84.

Solow, D. (1984), *Linear Programming: an introduction to finite improvement algorithms*, North-Holland, New York.

Treiber, S. (1984), 'Multivariable control of non-square systems', *Ind. Eng. Chem. Process Des. Dev.* **23**, 854–857.

Wilkinson, D. J., Morris, A. J. & Tham, M. T. (1994), 'Multivariable constrained predictive control (with application to high performance distillation)', *Int J Control* **59**(3), 841–862.

Wood, R. K. & Berry, M. W. (1973), 'Terminal composition control of a binary distillation column', *Chem. Eng. Sci.* **28**, 1707–1717.

Yedavalli, R. K. (1985), 'Perturbation bounds for robust stability in linear state space models', *Int. J. Control* **42**(6), 1507–1517.

Zafiriou, E. (1990), 'Robust model predictive control of processes with hard constraints', *Computers Chem. Engng* **14**(4/5), 359–371.

Zafiriou, E. (1991), On the closed-loop stability of constrained QDMC, *in* '1991 American Control Conference', pp. 2367–2372.

Zafiriou, E. & Hung-Wen, C. (1993), Output constraint softening for SISO model predictive control, *in* 'Proceedings 1993 American Control Conference', San Francisco, California, USA, pp. 372–376.

Zafiriou, E. & Marchal, A. L. (1991), 'Stability of SISO quadratic dynamic matrix control with hard output constraints', *AICHE Journal* **37**(10), 1550–1560.

Zheng, A. & Morari, M. (1995), 'Stability of model predictive control with mixed constraints', *IEEE Trans. Auto. Cont.* **40**, 1818–1823.