REGULATORY LEVEL MODEL
PREDICTIVE CONTROL

A THESIS SUBMITTED TO THE UNIVERSITY OF MANCHESTER
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
IN THE FACULTY OF ENGINEERING AND PHYSICAL SCIENCES

2015

By
Yusuf Abubakar Sha’aban
School of Electrical and Electronic Engineering
Contents

List of Abbreviations 10
Declaration 15
Copyright 16
Dedication 17
Acknowledgements 18
About the Author 19

1 Introduction 20
   1.1 Motivation .................................................. 20
   1.2 Aim and Objectives .......................................... 24
   1.3 Contribution of thesis ...................................... 25
   1.4 Thesis Outline ............................................... 26
   1.5 Publications ................................................ 27

2 Literature Review 28
   2.1 Proportional Integral Derivative (PID) control .......... 28
   2.2 Model Predictive Control ................................... 32
       2.2.1 Decentralised/Distributed Model Predictive Control (MPC) 36
   2.3 System Identification ....................................... 38
   2.4 Conclusion .................................................. 42

3 Background Concepts 44
   3.1 System Identification ....................................... 44
       3.1.1 Ordinary Least Square (OLS) ......................... 45
3.1.2 Recursive Least Squares (RLS) ............................ 46
3.1.3 Model Structures ...................................... 48
3.1.4 Prediction error methods .............................. 50
3.1.5 Identifiability and Informative Data .................... 53
3.1.6 Rich and exciting signal ............................... 53
3.1.7 Closed-loop identifiability without external excitation . 55
3.2 Model Predictive Control ................................. 57
  3.2.1 State space model and linear velocity form .......... 57
  3.2.2 Prediction ........................................... 59
  3.2.3 Cost Function ...................................... 61
  3.2.4 Unconstrained MPC .................................. 62
  3.2.5 Constrained MPC .................................... 62
  3.2.6 Stability ........................................... 64
3.3 Conclusions ............................................... 66

4 Control of single-input single-output (SISO) Benchmark Processes 67
  4.1 Introduction .......................................... 67
  4.2 Problem formulation ................................... 68
  4.3 Complex processes ..................................... 69
  4.4 Dead-time Dominant Processes ......................... 70
    4.4.1 The effect on control performance ................. 71
    4.4.2 PID for time delayed processes .................... 73
    4.4.3 MPC for time delay process ......................... 76
    4.4.4 Numerical simulations ............................ 76
    4.4.5 Summary ....................................... 83
  4.5 Higher Order Systems .................................. 84
    4.5.1 Numerical Simulations ............................ 87
    4.5.2 Summary ....................................... 90
  4.6 System with fast and slow dynamics .................... 91
    4.6.1 Summary ....................................... 93
  4.7 Conclusions .......................................... 94

5 Industrial Case-Study: Brine Dechlorination 97
  5.1 Introduction .......................................... 97
  5.2 Problem Background ................................... 98
  5.3 Control Objectives ................................... 100
### List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Gain margin and phase cross-over frequency variation with $\kappa$</td>
<td>73</td>
</tr>
<tr>
<td>4.2</td>
<td>genetic algorithm (GA) parameter values</td>
<td>74</td>
</tr>
<tr>
<td>4.3</td>
<td>Gain margin and phase cross-over frequency variation with $n$ and $\alpha$</td>
<td>86</td>
</tr>
<tr>
<td>4.4</td>
<td>Controller Performance for $G_n$ (unconstrained)</td>
<td>88</td>
</tr>
<tr>
<td>4.5</td>
<td>Controller Performance for $G_n$ (constrained)</td>
<td>88</td>
</tr>
<tr>
<td>4.6</td>
<td>Controller Performance for $G_\alpha$ (unconstrained)</td>
<td>89</td>
</tr>
<tr>
<td>4.7</td>
<td>Controller Performance for $G_\alpha$ (constrained)</td>
<td>89</td>
</tr>
<tr>
<td>5.1</td>
<td>Variability metrics at different set-points</td>
<td>104</td>
</tr>
<tr>
<td>5.2</td>
<td>Controller Performance</td>
<td>108</td>
</tr>
<tr>
<td>5.3</td>
<td>MAE for controllers</td>
<td>119</td>
</tr>
<tr>
<td>6.1</td>
<td>Constraints on $n_k$ for proportional (P)/Proportional Integral (PI) controller and different process models</td>
<td>126</td>
</tr>
<tr>
<td>6.2</td>
<td>Constraints on $n_k$ for PID controller and different process models</td>
<td>126</td>
</tr>
<tr>
<td>6.3</td>
<td>SA parameter values</td>
<td>131</td>
</tr>
</tbody>
</table>
List of Figures

1.1 Traditional use of MPC [12] .................................................. 21
1.2 Emerging trend of MPC [12] .................................................. 22

2.1 Hierarchical and decentralized/distributed MPC [93] ..................... 36
2.2 Reaction Curve ................................................................. 41

3.1 Closed loop system ............................................................. 55

4.1 Feedback representation of control problem .............................. 68
4.2 Bode plot showing effect of dead-time ................................... 72
4.3 Plots of integral of absolute error (IAE) against $\frac{\theta}{\tau}$ for $G(s) = \frac{2e^{-\theta s}}{s+1}$ .................. 77
4.4 Plots of IAE against $\frac{\theta}{\tau}$ for $G(s) = \frac{2e^{-\theta s}}{s+1}$ .................. 78
4.5 Sample plots of manipulated and control variables for $G(s) = \frac{2e^{-\theta s}}{s+1}$ .................................. 79
4.6 Sample plots of manipulated and control variables for $G(s) = \frac{2e^{-\theta s}}{s+1}$ .................. 80
4.7 Plots of IAE against $\frac{\theta}{\tau}$ for $G(s) = \frac{e^{-\theta s}}{7s+1}$ .................. 81
4.8 Plots of IAE against $\frac{\theta}{\tau}$ for $G(s) = \frac{e^{-\theta s}}{7s+1}$ .................. 82
4.9 Plots of IAE against $\frac{\theta}{\tau}$ for $G(s) = \frac{2e^{-\theta s}}{(3s+1)(10s+1)}$ ............ 83
4.10 Plots of IAE against $\frac{\theta}{\tau}$ for $G(s) = \frac{2e^{-\theta s}}{(3s+1)(10s+1)}$ ............ 84
4.11 Bode plot of $G_n(s) = \frac{1}{(s+1)^n}$ for different values of $n$ .......... 85
4.12 Bode plot of $G_\alpha = \frac{1}{(s+1)(1+\alpha)(1+\alpha^2)(1+\alpha^3)}$ for different values of $\alpha$ .... 86
4.13 Open loop step response of $G_{FS}$ and $G_{fs}$ .............................. 92
4.14 Step and load disturbance response with Internal Model Control (IMC)-
tuned PI control ($\tau_c = 1s$) ................................................. 92
4.15 Bode plot of true system, $G_{FS}$ and its first order approximation $G_{fs}$ .... 93
4.16 ......................................................................................... 94

5.1 Block diagram showing chloro-alkali production stages ................ 98
5.2 Membrane cell ................................................................. 99
5.3 Process flow diagram(anolyte section) ..................................... 101
A.5 Plots of manipulated and control variables for $G_n(n = 4)$ . . . . . . . 172
A.6 Plots of manipulated and control variables for $G_n(n = 8)$ . . . . . . . 173
A.7 Plots of manipulated and control variables for $G_\alpha(\alpha = 0.1)$ . . . . . 174
A.8 Plots of manipulated and control variables for $G_\alpha(\alpha = 0.2)$ . . . . . 175
A.9 Plots of manipulated and control variables for $G_\alpha(\alpha = 0.5)$ . . . . . 176
A.10 Plots of manipulated and control variables for $G_\alpha(\alpha = 1)$ . . . . . 177

B.1 Plots of PH and MV for both MPC and PID due to Step disturbance . 179
B.2 Plots of PH and MV for both MPC and PID due to Step disturbance . 180
B.3 Plots of PH and MV for Step disturbance in MATLAB . . . . . . . . . . . 181
B.4 Plant test data at pH around 1.7 . . . . . . . . . . . . . . . . . . . . . . 182
B.5 Plant test data pH around 1.7 . . . . . . . . . . . . . . . . . . . . . . 183
B.6 MPC Horizons . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 184
List of Abbreviations

APC  advanced process control
ARMAX  Autoregressive Moving Average with exogenous input
ARX  Autoregressive with exogenous input
BJ  Box-Jenkings
Cl⁻  chloride ions
Cl₂  Chlorine Gas
CV  controlled variable
DCS  distributed control system
DMC  dynamic matrix control
dMPC  decentralised MPC
DMPC  distributed MPC
DV  disturbance variable
ENMSS-MPC  extended non-minimal state space model predictive control
FIR  Finite Impulse Response
FCCU  fluid catalytic control unit
FOPDT  first order plus dead-time
FPGA  field programmable gate arrays
GA  genetic algorithm
GC  general constraint
GBN  generalized binary noise
GPC  Generalised Predictive Control
H\(^+\)  hydrogen ions
H\(_2\)  hydrogen gas
H\(_2\)O  water
IAE  integral of absolute error
IMC  Internal Model Control
LCQ  linear quadratic controller
LQG  linear quadratic Gaussian
MAE  mean absolute error
MD  measured disturbance
MIMO  multi-input multi-output
MISO  multi-input single-output
MPC  Model Predictive Control
MV  manipulated variable
Na\(^+\)  sodium ions
NAOH  caustic soda
NMSS-MPC  non-minimal state space model predictive control
OE  Output Error
OH\(^-\)  hydroxyl ions
OLS  ordinary least squares
OPC  Open Connectivity
PFC  predictive functional control
P  proportional
PI  Proportional Integral
PID  Proportional Integral Derivative
PLC  programmable logic controller
rMPC  recentered barrier function MPC
PRBS  pseudo random binary sequence
RLS  recursive least squares
PE  prediction error
PV  process variable
QDF  quadratic difference form
QDMC  quadratic dynamic matrix control
RGA  relative gain array
RHC  receding horizon control
RLS  recursive least squares
RMPC  routine data MPC
SA  Simulated Annealing
SC  second constraint
SISO  single-input single-output
SOPDT  second order plus dead-time
SSE  sum of squared error
SVA  singular value analysis
TITO  two-input two-output
TV  total variation
UD  upper diagonal
The need to save energy, cut costs, and increase profit margin in process manufacture increases continually. There is also a global drive to reduce energy use and cut down CO₂ emission and combat climate change. These in turn have led to more stringent requirements on process control performance. Hence, the requirements for modern systems are often not achievable using classical control techniques. Therefore, advanced control strategies are often required to ensure optimal process performance. Despite these challenges, PID has continued to be the dominant industrial control scheme. However, for systems with complex dynamics and/or high performance requirements, PID control may not be sufficient. Therefore, a significant number of industrial control loops are not performing optimally and more advanced control than PID may be required in order to achieve optimal performance. MPC is one of the advanced control schemes that has had a significant impact in the industry. Despite the benefits associated with the implementation of MPC, the technology has remained a niche application in process manufacture. This thesis seeks to address these issues by developing ways that could lead to widespread application of MPC.

In the first part of this thesis, a study was carried out to understand the characteristics of processes that would benefit from the application of MPC at the regulatory control level even in the single-input single-output (SISO) case. This is a departure from the common practice in which MPC is applied at the supervisory control layer delivering set points to PID controllers at the regulatory control layer. Both numerical simulation and industrial studies were used to show and quantify benefits of MPC for SISO applications at the regulatory control layer.

Some issues that have led to the limited application of MPC include the cost and human efforts associated with modelling and controller design. And to achieve high process performance, accurate models are required. To address this issue, in the second part of this thesis, a novel technique for designing MPC from routine plant data – routine data MPC (RMPC) is proposed. The proposed technique was successfully implemented on process models. This technique would reduce the high human cost associated with MPC deployment, which could make it a widespread rather than niche application in the process manufacturing industry.
Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.
Copyright

i. The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the “Copyright”) and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

ii. Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made only in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.

iii. The ownership of certain Copyright, patents, designs, trade marks and other intellectual property (the “Intellectual Property”) and any reproductions of copyright works in the thesis, for example graphs and tables (“Reproductions”), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

iv. Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see http://documents.manchester.ac.uk/DocuInfo.aspx?DocID=487), in any relevant Thesis restriction declarations deposited in the University Library, The University Library’s regulations (see http://www.manchester.ac.uk/library/aboutus/regulations) and in The University’s policy on presentation of Theses.
Dedication

To my parents,
who brought me into this world and prepared me for life
Acknowledgements

All praise be to the Almighty, Lord of the Universe most gracious most merciful for giving me the opportunity to go through this process successfully.

My Sincere gratitude goes to my supervisor Prof. Barry Lennox for his guidance, encouragement and support during the course of this work. What I have learnt from your vast experience will remain invaluable to me.

To my parents, words cannot express my appreciation for your love, prayers and support. I am sure you will be proud of my accomplishments. To my dear wife Sa’adatu, thank you for being there at the most difficult and trying times. To all my siblings, relatives and friends: thank you for your prayers, well wishes and encouragements.

I will like to thank my colleagues at ABUDEE, particularly Prof. MB Mu’azu and my brother Abdullahi. I would also like to thank my colleagues at the Control systems centre, their encouragements, support made my stay in F1 more interesting and rewarding. I will like to particularly thank Osman, Razak and Ahmad Bayati for their advise and sharing of useful ideas. I will not forget to mention Eduardo, Awai, Awo, Mobolaji, Harun and all other members of the group. My sincere gratitude goes to the OSAIS family and other brothers, Khaled, Abdo, Abdulganiy (Farmer) and other friends and people I met during my studies in Manchester.

Finally, I would like to acknowledge the financial support of the Nigerian government through the Petroleum Technology Development Fund (PTDF). Thank you for believing in me and giving me this opportunity.
About the Author

Yusuf Sha’aban received a B.Eng degree in Electrical Engineering from Ahmadu Bello University, Zaria, Nigeria in 2007, and an MSc. in Control Systems from Imperial College, London, UK. in 2010.

Yusuf joined the services of Ahmadu Bello University, Zaria, Nigeria after his B.Eng in 2008 as an assistant lecturer. He was with university as a Lecturer II before proceeding for a PhD at the University of Manchester. His PhD is in the area of Model Predictive Control. His PhD was supported by the Nigerian government through the Petroleum Technology Development Fund (PTDF).

Yusuf is a Member of IET, IEEE, Nigerian Society of Engineers (NSE) and a corporate member of the Council for the Regulation of Engineering in Nigeria (COREN). Yusuf is happily married.
Chapter 1

Introduction

1.1 Motivation

Feedback control has revolutionised most areas of process operations, leading to improved safety, performance, profitability and environmental sustainability [1]. In the process industries, it is considered a mature technology. However, with continuous change in process economics, need to reduce variability in product quality and energy (fuel and electricity) consumption, increase in market competition and demand for lower cost, the required performance for feedback control increases continually [1]. In most cases, the form of feedback is of little importance and it is feedback in itself that matters [2]. But certain applications may be more suitable for control by specific feedback forms. Of all the feedback schemes available, PID control is the most popular, accounting for over 90% of all industrial control schemes [3,4]. Therefore, it is often considered as the backbone of industrial regulatory control.

Most processes can be adequately regulated by PID. It has a simple structure and has been extensively studied [2–10]. These and other favourable features of PID have contributed to its success in industry [10]. When implemented on suitable applications, PID can provide excellent performance. In many processes it can attain the performance achievable by more complex and advanced control schemes if properly tuned [2]. However, some industrial applications have complex dynamics, and performance requirements may also be high. In such cases, PID control may not be able to achieve the required performance. Additionally, the performance expectations of modern systems have become higher due to advances in computer technology and modelling techniques [11]. Therefore, considering more advanced controllers has become imperative.
MPC has been identified as one of the only advanced control techniques that had a significant impact in the process industry during the past three to four decades [12]. This can be credited to its ability to address certain practical issues such as constraints, ability to carry-out optimisation in the loop, improved performance, ability to handle multi-variable plants efficiently [13] and availability of supporting theory. Furthermore, advancements in semiconductors (processing power and memory) have led to applications in more areas, for example power trains [14]. With more stringent requirements on control loops, the future of MPC technology in the process and other industries seems promising. More deployment of the technology is anticipated in the near future [15].

A significant number of process industries already have MPC products installed with many others having the infrastructure to support it. These predictive controllers are mostly operating at the supervisory level providing set-points to PID controllers lower in the control hierarchy. This implementation shown in Figure 1.1 will be referred to as the traditional approach. The traditional approach of MPC has become well established as a niche application for process manufacturing.

![Diagram](image-url)

Figure 1.1: Traditional use of MPC [12]
This traditional approach which has not been widely established beyond the niche application areas is not the most optimal or profitable application [12], but has become the norm due to safety concerns as the lower level PID controllers serve as backup whenever there are stability issues with the supervisory MPC. A more profitable option will be to eliminate the PID controllers so that the actuators are directly below the predictive controller as depicted in Figure 1.2. This new approach is becoming common with technological advancements and increasing memory and computational speed [12,16].

![Plant-wide static set-point optimization (daily)](image1)

![Predictive Control (Logic, overrides, Decoupling, Exception handling)](image2)

![Actuators (Valve servos etc)](image3)

Figure 1.2: Emerging trend of MPC [12]

Amongst the challenges of MPC is the requirement for skilled manpower and the need for accurate process models. The requirement for high level expertise also limits the application of MPC in industry. Moreover, the more popular PID is also available in off-the-shelf hardware such as programmable logic controllers (PLCs) while MPC requires a dedicated software (typically running on a PC). A study by ADERSA\(^1\) revealed that with the required technical support, the predictive functional control

---

\(^1\)ADERSA (Association pour le Developpement de l’Enseignement et de la Recherche en Systematique Appliquee) is french company whose primary business is advanced process control
(PFC)\(^2\) found more acceptance over PID with the technical staff. PFC therefore, found application as an alternative to PID when made available on the PLC [17]. Over 1000 applications of PFC were reported by early 1996 [18]. MPC is therefore an ideal candidate whenever PID control cannot achieve the required performance specifications.

Another point worth noting is that in most cases, the full potential of PID is not fully harnessed as most practitioners only implement PI controllers due to the problem associated with noise. That not withstanding, even the PI controllers are often not properly tuned. Several studies have shown that a significant number of industrial PID loops are poorly tuned [3, 5]. In that regard, there has been a reluctance to implement other extensions of PID, for example the smith predictor [19], which significantly increases the complexity of controller tuning [20].

For applications already regulated using some form of feedback (PID for example), the replacement MPC should perform at least as well as the existing controller [21]. Therefore, a good starting point would be to develop an equivalent MPC controller around the steady state operation point – when constraints are not active. The MPC so developed will then naturally inherit some of the properties of the existing PID controller [22]. This approach could significantly reduce the challenges associated with tuning, and improving controller performance then becomes trivial since initial controller performs as well as existing control. This approach could potentially allow for embedding MPC in the control execution layer. However, before proceeding it is imperative to understand the characteristics of systems that will benefit from the developed MPC controller.

In this work, systematic studies are carried out to understand the characteristics of process systems that will benefit from the application of MPC even in the SISO case. Using well known benchmark process models, the benefits achieved by the replacement of a PID controller with an MPC (having fewer tuning parameters than traditional approaches) were quantified. A novel procedure for designing and tuning MPC controllers with similar performance as the existing PID controller was developed. This could open up more areas of MPC application, making it a widespread rather than niche technology. The aims and objectives of this work are now highlighted.

\(^2\)PFC is a 3rd generation predictive control product from ADERSA
1.2 Aim and Objectives

The main aim of this work is to develop an efficient algorithm for the replacement of a PID/PI controller by an equivalent MPC controller. Equivalent here means, the designed MPC will have a performance similar to the existing PID controller. The MPC can then be tuned to achieve improved performance. The proposed controller must be intuitive and straightforward to implement. It is also necessary for the controller to preserve the attractive properties of conventional MPC such as constraint handling, while ensuring safe operation. This approach can be programmed into the distributed control system (DCS) and could allow for more deployment of MPC in the process manufacturing industry. To achieve this aim, a number of objectives have been identified as follows:

1. To carry out studies to understand the characteristics of systems that will benefit from the implementation of controllers that are more advanced than PID. This involves quantifying and understanding benefits achievable through simulation with MPC considered as the advanced controller of choice.

2. To carry out an industrial case study to validate the findings of Objective 1. This involves developing expertise on implementing control algorithms on actual industrial processes, including all practical considerations for ensuring safe implementation of industrial controllers.

3. To propose an efficient modelling procedure suitable for the implementation of low level MPC. The procedure will be such that disruption of process operations is minimised. And with emphasis on reducing the cost and manpower requirements associated with traditional MPC design methods.

4. To investigate and propose ways of directly replacing existing PID controllers with equivalent MPC based on results from Objective 3.

5. To develop a detailed algorithm and procedure that can be followed to implement the controller proposed in Objective 4.

6. To test the developed algorithms by implementing on simulations and comparing with the replaced PID controller.
1.3 Contribution of thesis

- In Chapter 4 a systematic study was carried out to understand the characteristics of systems that would benefit from the implementation of MPC at the regulatory control level. To achieve this, benchmark process models were identified and numerical simulations used to arrive at the conclusions of the chapter. Part of this contribution was published in [23]. The work carried out in this chapter can be summarised as follows:

1. A review was carried out to understand the performance of industrial PID controllers, which revealed that a significant number of these controllers are poorly tuned and not performing optimally.

2. Some characteristics of systems that affect controller performance were studied. Models having the studied characteristics were analysed to understand effect on control complexity.

3. Digital PID controllers were then applied to the identified process models and used as benchmark for comparison with MPC, a more advanced controller. The features accounting for control difficulty were varied and their effects on controller performance studied. Conclusions were drawn based on the observed performance of the two controllers.

- In Chapter 5, a case study was carried out to understand how industrial conditions affect the results obtained in Chapter 4. In the case study, an industrial process with complex dynamics regulated with a PI and lead-lag feed forward controller was used to study benefits achievable by replacing existing control with MPC. Even though a standard MPC algorithm was used, the nature of the problem and approach taken to achieve control makes the contribution significant. Variability analysis and a mechanistic model were used to solve control problem in a more efficient manner. A summary of work carried out is given below:

1. A study was carried out to understand complexity of problem. A non-linear mechanistic model of the process was used to carry out preliminary studies to understand benefits achievable with improved control – by replacing the existing PID with MPC.

2. Variability analysis was used to estimate achievable control improvements using stored plant data. The results of studies from mechanistic models and
variability analysis were used to study the viability of the control improvement project and to understand how much benefit was achievable.

3. **MPC** was implemented on the actual industrial process. Initial controller tuning was achieved using the mechanistic model of the process. To achieve improved control, steps were taken to address certain practical issues such as valve stiction. The control improvement was compared with predicted improvements and conclusions drawn. Improved control was achieved using the **MPC**.

- In Chapter 6, a novel method for designing **MPC** from routine plant data (RMPC) was proposed. Using the developed **RMPC** algorithm, an existing **PID/PI** can be replaced with an *equivalent** **MPC** controller without carrying out plant test. The developed controller is automatically tuned to emulate the performance of existing control. Therefore, it requires less human input when compared with conventional **MPC** tuning methods. It also allows for obtaining improved tuning in a straightforward manner. The work carried out in this chapter can be summarised as:

  1. A review was carried out to understand the requirements for closed loop plant data with noise only excitation to be informative. Based on the studied conditions, the requirements for **RMPC** were highlighted based on existing controller complexity and sampling time requirements.
  2. The method of designing **RMPC** was developed and a step by step algorithm for its implementation synthesized.
  3. The developed **RMPC** algorithm was applied to process models to replace existing **PI** controllers. The performance of **RMPC** was compared with **PI** on simulation models, steps required to improve controller performance demonstrated and conclusions were drawn.

### 1.4 Thesis Outline

This thesis begins with an introduction to the research carried out; the aims, objectives and contributions of the work were presented. Chapter 2 discusses the literature surveyed in order to achieve the objectives of the thesis. In Chapter 3, theory of background material necessary for the main chapters of thesis is presented.
CHAPTER 1. INTRODUCTION

The main chapters of the thesis can be grouped into two sections. The first section (Chapters 4 and 5) focused on understanding characteristics of systems that could benefit from replacement of existing regulatory layer controllers with MPC, while the second section (Chapter 6) deals with the development of a method for designing these MPC controllers.

Chapter 4 presents a study on the performance of PID and MPC controllers on systems with certain dynamics that affect control difficulty. Comparison of PID with MPC on selected process model benchmarks was carried. Through this study, processes that would benefit from MPC implementation are appreciated and achievable benefits quantified. Chapter 5 addresses the question addressed in Chapter 4 in a practical industrial problem. The industrial case study carried out involved the pH regulation of brine in a Chloro-alkali plant. This case study further validates the findings of Chapter 4 in an industrial setting.

In Chapter 6, a novel method for designing and tuning MPC from routine plant data is presented. The results in this chapter show that significant cost can be reduced by harnessing routine plant operation data.

Finally, the thesis is concluded in Chapter 7, where recommendations for future work are also presented.

1.5 Publications

Chapter 2

Literature Review

Process control plays a number of important roles in the process manufacturing industry. It ensures safe operations, profitability, efficiency, sustainability, reduced product variability, optimises process operations and minimises the effects of process operations on the environment [24]. Process control is a mature field and its evolution has largely been shaped by technological advancements. The process industry was particularly revolutionised by the advent of distributed computer systems and modern process information systems. Modern process control can generally be classified into three major classes; regulatory control like PID, conventional advanced control such as feed-forward and override control, and modern advanced control such as MPC [19]. Most industrial control loops are regulated using PID. However, industrial control problems mostly have some form of constraints. Hence, advanced control methodologies such as MPC are often required to systematically account for these constraints. The successful application of MPC and other model based control techniques require identification of reasonable models that capture plant dynamics.

In this chapter, a literature review of some key areas in process control is presented. In the next section, PID is discussed as the main industrial regulatory control scheme. Next, a review of MPC is presented in Section 2.2. A review on industrial system identification is presented in 2.3. And the chapter is concluded in Section 2.4.

2.1 PID control

PID is the most widely used form of feedback control. Its simple structure and ability to give acceptable performance in most cases has no doubt contributed to its success. Until the early 80s, the tuning methods of Ziegler and Nichols [25, 26] proposed about
four decades earlier was the state-of-the-art in industrial PID tuning. However, there
was a renewed interest in PID tuning in the 1980s [27,28]. And since then, PID tuning
has remained an area of active research [2]. However, despite the volume of research
activity on PID-tuning, a significant number of industrial control loops were still not
properly tuned [3].

A study of control loop performance in the 90s revealed that 95\% of loops operating
on a number of process plants were of PID form and that there wasn’t any standard
for implementing digital PID controllers [3, 29]. For example over 45 different PID
structures were recorded in [9]. It was therefore difficult to find any two identical PID
controllers in the market. More so, a lot of information about industrial tuning is buried
in propriety material [30]. Commonly used text book PID structures include the series
(cascade or interacting) and the ideal (parallel or non-interacting) form:

\[
\text{Series PID: } c(s) = K_c \left( \frac{\tau_I s + 1}{\tau_I s} \right) (\tau_D s + 1)
= \frac{K_c}{\tau_I s} \left( \tau_I \tau_D s^2 + (\tau_I + \tau_D) s + 1 \right)
\]

(2.1)

\[
\text{Parallel PID: } c(s) = K_c \left( 1 + \frac{1}{\tau_I s} + \tau_D s \right)
= \frac{K_c}{\tau_I s} \left( \tau_I \tau_D s^2 + \tau_I s + 1 \right)
\]

(2.2)

However, these text book versions are rarely if ever used in industry and discrete
versions of the controller are often used. It is also common to apply derivative action on
the controller output rather than the error to avoid derivative kick. Set-point weighting
is also used to improve controller set-point response [2]. A common representation of
this controller is given as:

\[
\begin{align*}
  u(t) &= k_p \left( y_{sp}(t) - y(t) \right) + k_i \int_0^t \left( y_{sp} - y \right) d\tau + k_d \left( \frac{dy(t)}{dt} \right) \\
\end{align*}
\]

(2.3)

A discrete difference form of (2.3) which uses a trapezium approximation at time step
k for a sampling time \( T_s \) is presented as:

\[
\Delta u(k) = k_p \left( [y_{sp}(k) - y(k)] - [y_{sp}(k-1) - y(k-1)] \right) +
  k_i e(k) T_s - \frac{k_d}{T_s} \left( y(k) - 2y(k-1) + y(k-2) \right)
\]

(2.4)
During the last 8 decades, numerous tuning rules have been proposed. Over 400 of such rules are documented in [9]. These rules were developed using various methods such as [31]: cycling, robust tuning, step response based, performance criteria optimisation and specified closed-loop response methods. A specified closed loop method, IMC based tuning [32] is now routinely used in industrial applications. This allows for tuning to achieve a desired closed-loop performance. For IMC-tuning, the closed loop time constant $\tau_c$ is the main tuning parameter. Different considerations for selecting $\tau_c$ based on open loop time constant $\tau$ and dead-time $\theta$ have been proposed [31].

Dead-time dominant systems are common in the process industries; all the 400 tuning rules presented in [9] are based on model representations with dead-time. While some researchers argue that derivative action is needed for systems with long dead-time [9], others are of the opinion that derivative action should not be used to compensate for dead-time [2, 33]. Another argument is that for processes with first order dynamics, the derivative action is not required [2]. Meanwhile, other researchers have developed PID tuning for first order process with derivative action in place [9, 31, 34]. Systems with oscillatory modes are relatively less common in industry [35]. As such not much is available on the tuning of systems with oscillatory response, particularly those with very low damping. For oscillatory systems, PI controllers can be used when performance requirements are not too high [2]. However, when high performance is required, then more advanced controllers may be necessary.

Multi-input multi-output (MIMO) systems are also common in the process industries. But the structure of PID was meant for SISO systems and practitioners are more comfortable with SISO design [36]. Multi-loop PID controllers are therefore tuned to form decentralised PID controllers. Methods of decentralised PID tuning in the literature include: detuning method [37, 38], sequential loop closing [39, 40], independent loop method [41, 42], and relay auto-tuning [28, 43–46]. Most of the fore mentioned decentralised PID techniques result in suboptimal or even poor performance [31]. In both the detuning and sequential loop tuning methods, the performance of the controller depends on the loop that is tuned first. While the relay auto tuning method in [44] requires iteration in loop design, which could be tedious.

Multivariable PID control design methods for MIMO systems are presented in [47, 48]. In multi-variable PID controllers, all manipulated variables (MVs) are adjusted to achieve control objectives in all controlled variables (CVs), unlike in decentralised PID. Multivariable controllers can give better results than SISO PID controllers [49].
However, due to difficulty in tuning, multi-variable PID controllers have not found widespread application in the industry. When compared with decentralised PID controllers, the number of parameters also increases significantly. For a two-input two-output (TITO) process for example, a 2 by 2 matrix of controllers is required.

Coupling is an inherent characteristic of process systems, resulting in loop interactions [31]. These interacting loops affect each other. Therefore, the control structure used has an effect on the decentralised controller performance and stability. Procedures for selecting MV – CV parings are therefore used to ensure best pairings are used. This is achieved using methods such as relative gain array (RGA) and singular value analysis (SVA) [50]. If loop interactions are strong, decouplers are then used to minimise the effects on other loops. Decoupling control [50, 51] is one of the earliest and common ways of reducing control loop interactions.

Recent decoupling techniques for TITO systems can be found in [36, 49, 52]. The decoupled controller technique offers the advantage of allowing for SISO design. Nevertheless, achieving both good set-point tracking and disturbance rejection has been a major challenge for PID controllers [53]. To this effect, there has been recent interest in developing alternatives to PID [53–57]. Possible alternatives to PID include: Discrete-time linear multi-input single-output (MISO) controllers, state feedback observers and MPC [2]. In [57], PID was identified as a simple, fast, easy to implement and easy to tune robust controller while the complications of MPC are justifiable only when there is significant economic benefit. A scalable SISO MPC, the linear quadratic controller (CLQ) was then proposed.

Despite the challenges associated with the implementation and tuning of PID to achieve high performance, PID has been the most successful controller in the industry so far. The controller is likely to remain relevant in the industry in the near future [2]. However, there is the need to understand when to use other controllers rather than PID. This can allow for the use of more appropriate controllers which can in turn lead to improvements in control and financial savings. And to also understand the benefits that can be attained by using such alternatives. In some cases, using other features tend to improve the performance of PID; for example smith predictor in dead-time dominant processes. However, studies have shown that practitioners are reluctant to implement these extensions [19]. In fact even the derivative action of PID is rarely used because of additional tuning requirements [3]. Amongst the alternatives for PID, MPC seems to be the most promising. It has since found wide acceptance in the oil and gas and process
industry and is capable of solving a number of the challenges faced by PID [2, 12].

2.2 Model Predictive Control

Over the past few decades MPC has continued to play an important role in the chemical, petroleum and related industries. The advancements in computing and memory technologies have opened up new areas of application for MPC [21]. The advent of multidisciplinary research is also seeing the penetration of MPC into newer fields of application such as: biomedical [58], autonomous vehicle control [59], traffic control, transportation, signal processing, power electronics [60], building automation [61], and hydroelectric applications [62]. The economic benefits associated with MPC and the presence of constraints in most control systems have also aided its popularity. Other valuable characteristics of MPC are relative transparency of tuning and ability to handle complex systems [11, 21]. At each time step in MPC, a finite horizon optimal control problem is solved and the first vector in the open-loop optimal sequence obtained is then selected as the current control input [63].

Various issues relating to the implementation of MPC are presented in the tutorial by Rawlings [64]; issues discussed include modelling, target calculation, feasibility [65], stability, optimality, state estimation and future research directions. MPC requires the open-loop optimal control problem to be solved online. This is achieved using the concept of receding horizon [12]; a finite horizon is used to ensure a solution is obtained in reasonable time relative to the plant dynamics. A survey of MPC technology in the industry was presented in [18], a detailed survey of commercially available MPC was presented while analysing the rapid development of the technology over a period of 25 years. Issues of interest to the academic community and practitioners were clearly identified. Initial MPC implementations did not guarantee stability, which was not a major concern to the practitioners [18, 21]. But as research continued, various methods of ensuring closed-loop stability were proposed. These methods use the MPC objective function as a Lyapunov function and a combination of the following ingredients: terminal cost, $F(\cdot)$, terminal constraint set, $X_f$ and local stabilising controller $\kappa_f(\cdot)$ [21, 63, 64, 66].

There has been significant interest in MPC tuning. It has been considered in most MPC texts [12, 67, 68]. The availability of many tuning variables have allowed for improved performance of MPC. However, this has made tuning more challenging
requiring systematic approaches [69]. Commonly used tuning parameters include; prediction horizon, control horizon, model horizon, weights on outputs, weights on rate of change of MV, weight on MV magnitudes, reference trajectory parameter and constraints. In [69], a review of available MPC tuning guidelines for various MPC formulations was presented. The availability of an appropriate model was identified as the first and key step to MPC tuning while bearing in mind the requirement for trade-off between robustness and performance. These tuning methods are generally classified into theoretical and industrial (heuristics) based tunings.

MPC can be classified into three major categories based on model types; Finite Impulse Response (FIR), transfer function and state space based MPC [70]. FIR and step response model based formulations are restricted to stable processes only. The model orders are typically high especially in the process industries where slow processes and dead times are common. Common formulations using FIR/step response include dynamic matrix control (DMC) [71] and quadratic dynamic matrix control (QDMC) [72]. Transfer function based formulations are applicable to a wider range of both stable and unstable processes. A well known transfer function formulation is Generalised Predictive Control (GPC) [73, 74]. Another formulation of MPC that has found acceptance is the PFC, which is a predictive control technique designed for SISO systems bearing in mind practitioners’ requirements for ease of understanding, installation and tuning [11].

Most recent literature is based on state space formulation of MPC [12, 75–78]. Owing to their relatively simpler design framework, ease of analysis and ease of extension of linear systems theory to the formulation, state space formulations have become the most popular in academia [12]. They are well suited for multi-variable processes. But, the use of state observers for state estimation in state space based MPC often lead to issues associated with convergence rate, robustness and non-linearity [70, 79].

A number of state space formulations exist [12, 68], amongst which are the non-minimal state space model predictive control (NMSS-MPC) [70] and the extended non-minimal state space model predictive control (ENMSS-MPC) [80]. These two formulations can be considered as state space implementations of transfer function formulation and therefore exhibit certain favourable characteristics of both. NMSS-MPC has been shown to provide improved performance over both GPC and standard linear quadratic Gaussian (LQG) control [81]. The need for a state observer is avoided as the states comprise of present and past measured plant outputs and inputs. The
problem of performance deterioration with constraints is solved by NMSS-MPC while maintaining the simplicity of other state space formulations [70, 78, 82]. However, the NMSS-MPC also suffers from performance deterioration due to plant-model mismatch. This problem is addressed in ENMSS-MPC by including the output errors as part of the state variables [80]. Both NMSS-MPC and ENMSS-MPC formulations result in state space matrices with higher dimensionality as compared to minimal state space realizations.

Various cost functions that are minimised within the MPC formulation are possible [12, 18, 67, 68]. In each of the formulations, either the optimal control input or its change are computed at each sample time. In the linear model velocity-form [82] for example, the change in control variable from the previous time step is computed. This approach has the advantage of having a straightforward implementation while embedding integral action, a requirement for offset free control.

Traditionally, MPC is applied at the supervisory control layer delivering set-points to PID controllers. This is still prevalent, especially for large and complex multi-variable systems. However, MPC is now being applied at the regulatory control layer. This is partly due to improvements in optimisation algorithms and increased computing speed. Another factor is increased market competition and tighter environmental regulations. Previous works on the implementation of MPC at the regulatory layer include [16, 83–85]. In [16], multi-variable feedback was used to regulate an edible oil refining plant; this was achieved using recentered barrier function MPC (rMPC). Increased production was achieved by operating very close to actuator limits. A PLC was used to interface the rMPC computer with sensors and other hardware.

In [17], an auto-tuned predictive controller was developed using minimal plant information; modelling was based on relatively basic but efficient process information. This MPC controller was then embedded on an industrial PLC. Simulation results showed that the controller achieved a performance similar to a well-tuned PID but with much less effort. These results were extended in [83] to accommodate constraints, it was also shown that MPC can be an alternative to conventional PID within the regulatory control loop. In [84], a constrained MPC with predefined complexity was developed. Parametric solution based on points (using regular shapes) was used to reduce online computational and storage requirement. This ensures that there is no increase in complexity with increase in state dimension. This suboptimal solution still retains stability properties of other MPC formulations. Implementation on a plant using PLC
confirmed that the benefits from simplicity and efficiency outweighed the resulting loss in performance.

Most existing process systems are already controlled using some form of feedback. Hence designing an MPC controller for such systems will involve replacing the existing PID controller. An approach can be to design MPC controllers that are equivalent to the existing controllers and then work to improve the developed MPC controller. Some works concerned with linking existing controller with MPC include [86–88]. In [88], a method of replicating the performance of an existing controller with MPC using reverse engineering was presented. The method finds an MPC cost function and state estimator that give same performance as a lower order continuous controller. The obtained continuous time results are then converted to discrete form for MPC implementation. This method was extended in [87] to handle issues of discretisation before reverse engineering and to deal with extra degrees of freedom when a low order controller is used. In [86] on the other hand, the tuning of MPC with similar behaviour as an existing linear state feedback controller was achieved by finding the optimal weight matrices, $Q$, $R$ and $P$. These formulations assumed that an appropriate plant model was available which is not always the case. In fact modelling is the most important aspect of MPC design and tuning [89].

Since most existing low level industrial controllers are of the PID type implemented in a decentralised fashion. A good starting point for the replacement of suitable loops with MPC at the low level is to consider developing the MPC in a decentralised or distributed manner. Using this approach, the MPC can be designed incrementally beginning with a single loops. This approach could minimise disruptions in plant operations often associated with traditional MPC implementation methods. Implementing such SISO based controllers using PLCs could also lead to more application of MPC at the low level.

In large scale and networked systems, the management of centralised MPC controllers may become too complex, not scalable and inflexible [90, 91]. Hence, this class of systems may have MPC implemented in a decentralised fashion, with each subsystem having a local MPC controller that computes its optimal control sequence.

---

$Q$ is the state weighting matrix, $R$ the input weighting matrix and $P$ terminal state weight.
2.2.1 Decentralised/Distributed MPC

An early treatment of decentralised control can be found in [92]. In decentralised MPC (dMPC), controllers compute their outputs irrespective of other controllers and there is no exchange of information during control computation. Hence, it is not affected by communication issues. This may suffice for weakly interacting systems, but in systems with strong interactions, it leads to deterioration of controller performance [93].

To cater for this performance degradation, distributed MPC (DMPC) schemes are used. In distributed MPC (DMPC), local controllers exchange information during controller computations. This information exchange caters for strong interactions but is affected by communication issues and requires iteration. To achieve performance equivalent to centralised MPC, DMPC needs to converge if the computational time allows [94].

A common architecture of DMPC is hierarchical as shown in Figure 2.1. Various approaches to DMPC are available based on both linear and non-linear models [93]. The various methods differ mainly in the manner of handling coupling. Of interest are the methods presented in [94–100]. In the method by Alessio [95, 96], decoupling is handled in prediction equations using the degree of decoupling as a tuning factor. In [97, 98], a model with decoupled inputs was used and neighbouring inputs were modelled as disturbances in predictions (communication-based MPC). In [94, 99, 100], negotiation amongst subsystems was used to achieve a cooperation-based MPC. In cooperation-based MPC, issues such as closed-loop stability, infeasibility, optimality and local estimation in DMPC framework are treated.

Figure 2.1: Hierarchical and decentralized/distributed MPC [93]
Quite a number of plants have decentralised PID in place. Hence, a number of works have been on how to implement DMPC on plants with local PIDs already installed. For example, the formulation in [99] ensures nominal closed loop stability using either state or output feedback. This cooperation-based DMPC, if allowed to iterate to convergence has an optimal performance equivalent to centralised MPC. Communication-based MPC, on the other hand, mostly converge to the Nash equilibrium which is often different from the Pareto-optimal solution [94]. Other works on cooperative DMPC include [101] which is further extended to reduce computational requirements in [102] and to cater for non-linear systems in [103].

In [104], non-cooperative DMPC [105] was proposed using dissipativity. Dynamic supply rates and storage functions in quadratic difference form (QDF) [106] were used to include future inputs and outputs making it suitable for an MPC framework. The local formulation of the optimization problem makes the approach simpler with less communication burden. However, if network interactions are strong, non-cooperative DMPC can result in poorer performance than decentralised MPC (dMPC), which could lead to instability [101].

In [107], eight MPC controllers were compared; two centralised MPCs, a decentralised controller and four DMPC schemes were considered based on closed loop performance. The subsystems were only coupled through their inputs. Network communication issues and timing were not considered in the study. However, obtained results can be generalised because any system can be split into subsystems coupled only through their inputs [108]. All controllers considered in the study exhibited good closed loop properties in the presence of disturbance and plant-model mismatch. In general, the results showed that centralised MPC controllers gave the best performance. While dMPC gave the worst overall performance, this performance was improved by DMPC schemes. The improvement depends on controller formulation and design. However, for transient performance dMPC had the shortest settling time and best transient performance.

A major factor affecting performance in any MPC is the availability of models that adequately captures plant dynamics. And for fast closed-loop performance, high precision models are required [69]. The process of obtaining these models from plant data is known as system identification and is the most tardy and arduous task in MPC projects [89].
2.3 System Identification

Modern control systems are designed using some form of process knowledge or model. These models can be developed theoretically from first principles – grey-box models, empirically from process data – black-box models or a combination of the two – white-box models. Grey-box modelling is favoured in fields like mechanical, electrical and aerospace engineering. However, it is impractical for large complex processes [109]. Therefore, black-box modelling, known as system identification is favoured in the process industries [110]. System identification has become an important and integral aspect of model based control. In some situations, white box models are developed by deriving models from first principles and then fitting some of the model parameters to plant data [111].

Some key issues in system identification include plant tests, model structures, estimation methods and model validation. Plant tests play an important role in successful model identification. Traditional plant test methods involve the use of single variable and sequential open-loop step tests [112]. This method is transparent allowing for more insight into process dynamics in an intuitive manner. However, these single loop methods are characterized by long test durations and require high human commitment due to their manual nature [89], making plant tests very expensive. Plant tests typically account for up to 75 % of cost associated with MPC projects [113]. Open loop tests also disrupt normal plant operations and may have economic and even safety consequences.

The quality of data used for identification will have an impact on how good the identified model will be. Hence, to obtain more informative data, it is often necessary to excite a range of frequencies of interest. Typical excitation signals that ensure persistent excitation [114] include; pseudo random binary sequence (PRBS), generalized binary noise (GBN), filtered white noise and sum of sinusoids [115–117]. These signals can be generated automatically thereby reducing the human requirement of plant tests [115]. PRBS and GBN signals are more commonly used in the industry [115, 118], as they reduce the effect of test on plant operations and can be designed to ensure optimum test duration. However, the use of these signals in a sequential fashion still requires long plant test durations, and resulting data may not have good information about the multi-variable characteristics of MIMO systems [112]. Hence, exciting all or a subset of input variables concurrently which is now gaining acceptability in the industry offers the advantages of shorter test durations and better information about
plant characteristics [119].

In some cases, coupling, directional dependent dynamics and other peculiarities associated with process systems necessitate special considerations for successful model identification. A host of practical issues associated with the identification of process models were discussed in [115]; detailed project procedures were given based on industrial case studies while keeping the theory easy for practitioners. For example in coupled and directional systems, the use of partly correlated signals to identify multi-input models yielded better result [115, 120]. This is in contrast with the requirement for non-coupled systems which require excitation signals to be uncorrelated.

The data generated from plant tests is used to identify models using any of the available model structures. Model structures commonly found in identification literature include [114]: Autoregressive with exogenous input (ARX), Autoregressive Moving Average with exogenous input (ARMAX), Output Error (OE) and Box-Jenkins (BJ). Both ARX and FIR parameters are estimated using linear least squares which is numerically reliable, making them popular in industrial applications [121]. The other structures require numerical optimisation routines which make them susceptible to non-optimal results, but with the ability to model more complex disturbances [114]. Despite other model structure being more compact, FIR model found more acceptability because of the relative ease in model parameter selection as only a single parameter (the process settling time) needs to be specified [89]. FIR requires a large number of model parameters making it computationally expensive for slow and large systems. In slow systems, FIR model variance error is high and estimates may be biased due to truncation [89]. Therefore in such situations, ARX models are often preferred.

ARX model parameters are typically estimated using ordinary least squares (OLS) and some of its extensions [122]. However, recursive estimation methods are essential for real-time adaptive control applications. These techniques update model parameters as new data become available, making them suitable for both on-line and off-line estimation. A frequently used recursive technique is the recursive least squares (RLS) algorithm [123]. With the RLS algorithm, as new data set becomes available, the inverse of the updated covariance matrix is required. This often leads to ill-conditioning. Therefore, to prevent the covariance matrix from becoming ill conditioned, the upper diagonal (UD)-RLS method was developed [122]. The UD-RLS updates the square root of the covariance matrix. Because of its robustness and ability to cope with noise and collinearity, the UD-RLS algorithm has found use in non-adaptive industrial
applications [124, 125]. In these applications, the RLS algorithm is used to identify model parameters off-line and the identified parameters are not updated on-line.

Despite system identification being one of the most active areas of research in academic control community, the industry was reluctant to apply most of the results in industrial MPC [119]. However, due to increased competition and demand for better models, newer identification results are now being applied in industrial MPC. For example subspace identification [89, 114] which is numerically efficient and allows for estimation of state space models directly from input-output data is now applied in industrial MPC [126]. Over the past 10 to 13 years, there have been significant efforts to reduce the requirement for expertise and cost associated with modelling for MPC, resulting in improvements in certain aspects of modelling for MPC [112]. Some of these aspects include: use of automated multi-variable tests, use of closed-loop tests, use of parametric models and model validation using model quality grading [89, 127]. These new features are now finding acceptability amongst major industrial MPC vendors [112].

Recent advancements have also allowed for development of automatic modelling procedures for MPC. These automatic methods are aimed at reducing the high requirements for expertise and time associated with MPC modelling while improving performance. For example in [119], a semi automatic MPC was developed using the ASYM method [115, 128] which is based on the asymptomatic theory of system identification [129, 130]. A fully automated modelling technique for MPC was also given in [112]. This automatic MPC design method uses features such as closed-loop identification, frequency domain based model order selection and error bound based model validation to achieve its automatic operation. The model order selection method uses variance and bias errors over the frequency of interest. Model grading was used for model validation. Model grading was achieved using $3\sigma$ bound derived using asymptotic theory and only high grade models (grades A and B) are considered for automatic selection. Using the developed method on an industrial process decreased test duration from between 2 - 3 weeks to 4.5 days [112].

First order plus dead-time (FOPDT) models are still routinely used to approximate a wide range of real process systems. These models are obtained using step test measurements [31, 67], which is a popular procedure in the process industries. These FOPDT models also act as starting points for more advanced identification procedures. In the reaction curve method, the plant is subjected to a step change in manipulated
variable and the model parameters are obtained as depicted in Figure 2.2. Where $\theta$ is the process dead-time and $\tau$ the time constant. Also commonly used are second order plus dead-time (SOPDT) models. In some cases, models obtained using newer and more advanced identification techniques are converted to their first and second order approximations to give operators a better feel of the nature of response [89].

Some processes do not allow for open loop plant tests due to safety and/or economic reasons. Moreover, closed-loop identification offers certain advantages in retuning existing controllers and the design of new ones [109]. To obtain unbiased and consistent estimates, prediction error methods [114] are necessary as non parametric (e.g. FIR) and subspace methods are known to give biased estimates with closed loop data [89]. The prediction error identification framework requires that an identifiable model structure is used, and that the resulting covariance matrix (inverse of information matrix) is finite [109]. Hence, for a finite covariance matrix, the information matrix is required to be positive definite, which can be achieved using informative experiments.

Informativity of experiments is often achieved by using sufficiently rich signals. For
example in open-loop identification, the requirement is for the input signal to be persistently exciting [131]. For closed-loop identification, identifiability conditions can be given in terms of controller complexity. For example it has been shown that if the feedback controller has significant non-linearities or is of a higher order than the process, then successful identification is possible in closed loop [114]. Minimal requirements for closed-loop identification are also given in terms of trade-off between richness of external signal and complexity of feedback controller [109]. So that assuming a sufficiently complex controller with no closed-loop pole zero cancellations, informative data can be obtained from noise only excitation i.e. without any other external excitation other than the natural plant disturbances [131]. However, pole-zero cancellations cannot be ruled out in advanced control.

Developments in modern process information systems and falling prices of memory have allowed for the storage of vast amount of data from numerous measurements [132]. Harnessing the process information embedded in such data can reduce the costs associated with plant test. Techniques for identifying sufficiently excited routine operation data from historic data have been developed [133]. Identifiability conditions for noise only identification have also been extended to account for closed-loop pole-zero cancellations [134–136]. Hence, the recipe for MPC relevant identification from routine operation data is well developed. For example identifiability conditions for ARMAX model structure are given in [134]. These conditions which are based on relative order of model and controller are further extended to accommodate general prediction error structures and expressed in terms of sampling time requirements in [135].

2.4 Conclusion

PID has always been the dominant industrial control strategy. However, developments in computer technology and modelling techniques allowed for the development of model based techniques. These developments redefined the performance requirements of modern process systems. This coupled with tighter profit margins particularly in the oil refineries necessitated the development of MPC as an industrial solution. The industrial applicability of MPC, ease of tuning and ability to solve practical industrial concerns made it a very successful advanced control scheme.

Each of these two controllers has its weaknesses and strengths. They are therefore complimentary and to fully benefit from them, it is essential that the appropriate control
scheme is used for the problem at hand. However, this is not generally the case as PID is found in many processes that may not necessarily be well suited for its control. Therefore, there is the need to identify those processes that are not well suited for PID control and quantify benefits achievable by replacing such loops with more suitable control schemes such as MPC. Also for any control to be acceptable to the practitioners, ease of implementation and tuning are factors that cannot be overlooked. It is therefore imperative to develop easy and efficient ways of designing MPC to replace PID in non suitable processes.

System identification is an important factor in MPC and other model based control schemes. It accounts for significant cost associated with the implementation of MPC. The industry has not fully harnessed latest findings in system identification. However, there is a recent shift and results from the academic community are now being explored to reduce the cost associated with system identification. The vast amount of industrial process data can be of great benefit if latest results in closed-loop system identification are explored; significant research activity is already apparent in this area.
Chapter 3

Background Concepts

In this chapter, some necessary ingredients for process control and MPC are discussed. These consist mainly of background information from existing methods and results. Before any model based control is feasible, system identification techniques are used to obtain reasonable descriptions of plant dynamics. The first part of this chapter therefore gives some background on system identification. In the second part of the chapter, basic principles of MPC using state space formulation are introduced. State space formulation of MPC have become the standard in modern literature [12]. Other formulations can easily be converted to their state space equivalent. Issues such as prediction, constraint handling and stability are briefly presented.

3.1 System Identification

A general discrete model structure for linear time invariant systems is given as:

\[ y(t) = G(q, \theta)u(t) + H(q, \theta)e(t) \]  \hspace{1cm} (3.1)

With rational transfer functions of plant and disturbance, \( G \) and \( H \) respectively, and independent zero mean random variable \( e(t) \) with variance \( \sigma^2 \). Parameter \( q \) denotes the shift operator: \( qy(t) = y(t+1) \) and \( q^{-1}y(t) = y(t-1) \). Using the prediction error framework, a predictor for (3.1) is [114]:

\[ \hat{y}(t|\theta) = G(q, \theta)u(t) + [1 - H^{-1}] [y(t) - G(q, \theta)u(t)] \]  \hspace{1cm} (3.2)
CHAPTER 3. BACKGROUND CONCEPTS

The transfer functions $G$ and $H$ are often parametrised using polynomials in the shift operator $q$. The parameters of these polynomials are obtained using estimation techniques such as OLS and RLS. This is achieved using an appropriate model structure. For a given model structure, the data used for identification is also required to be informative. In this section, discussions on OLS, RLS and model structures are presented. The requirement for informative data in a prediction error framework is also discussed.

3.1.1 Ordinary Least Square (OLS)

Consider a process model represented as:

$$y(t) = x_1(t)\theta_1 + x_2(t)\theta_2 + \ldots x_n(t)\theta_n y(t) = \phi^T \theta$$

(3.3)

where $\phi^T(t) = \left[ x_1(t) \quad x_2(t) \ldots \ x_n(t) \right]$ is the vector of instantaneous values of $n$ predictors at time $t$, $\theta = \left[ \theta_1 \quad \theta_2 \ldots \quad \theta_n \right]^T$ is a vector of unknown parameters and $y(t)$ is the output variable. Given $N$ measurements of $y(t)$, $Y(t) = \left[ y(1) \quad y(2) \ldots \quad y(N) \right]^T$ and $x_i(t): i = 1, 2, \ldots, n$ at time, $t = 1, 2, 3 \ldots N$. The regression model (3.3) can be written as:

$$Y(t) = \Phi \theta$$

(3.4)

where $\Phi = \begin{bmatrix} \phi^T(1) \\ \phi^T(2) \\ \vdots \\ \phi^T(N) \end{bmatrix}$ is the regression matrix. The nature of the solution $\theta$ depends on the number of observations $N$ relative to the number of independent variables $n$:

- If $n > N$, then there is an infinite number of solutions.
- If $n < N$, then there is no unique solution and $\theta$ should be selected according to the least squares criterion given in (3.10)
- If $n = N$, then there is a unique solution, $\hat{\theta} = \Phi^{-1} y$

Given the residual error, $\varepsilon = y(t) - \hat{y}(t) = y(t) - \phi^T \theta$. The least squares estimate
minimises the loss function:

\[ J_{LS} = \frac{1}{2} \sum_{t=1}^{N} (y(t) - \phi^T(t)\theta)^2 \]  
(3.5)

\[ = \frac{1}{2} \sum_{t=1}^{N} \epsilon(t)^2 = \frac{1}{2} \mathcal{E}^T \mathcal{E} \]  
(3.6)

\[ = \frac{1}{2} (Y - \Phi\theta)^T (Y - \Phi\theta) \]  
(3.7)

\[ = \frac{1}{2} (Y^T Y - 2Y^T \Phi\theta + \theta^T \Phi^T \Phi \theta) \]  
(3.8)

At the stationary points, the first order derivative of \( J_{LS}(\theta) \) with respect to \( \theta \) is zero.

\[
\left. \frac{\partial J_{LS}}{\partial \theta} \right|_{\theta = \hat{\theta}} = \frac{1}{2} (2\Phi^T \Phi \hat{\theta} - 2\Phi^T Y) = 0
\]  
(3.9)

\[ \hat{\theta} = (\Phi^T \Phi)^{-1} \Phi^T Y \]  
(3.10)

**Theorem 3.1. (Least-squares estimation) [123]:** The function of (3.6) is minimal for parameters \( \hat{\theta} \) such that

\[ \Phi^T \Phi \hat{\theta} = \Phi^T Y \]  
(3.11)

*If the matrix \( \Phi^T \Phi \) is nonsingular, the minimum is unique and is given by

\[ \hat{\theta} = (\Phi^T \Phi)^{-1} \Phi^T Y \]  
(3.12)

In real time and adaptive applications, as data matrices become large the correlation matrix also becomes large. Repeating the parameter estimation calculation using all available data can become intractable as it will require a lot of memory and computational resources. Moreover, the latest data set may give true information about the current dynamics of the process as dynamics change with time. Hence, for such applications, RLS is often used.

### 3.1.2 Recursive Least Squares (RLS)

In recursive least squares, the estimate from the current step \( \hat{\theta}(t = N) \) is updated using the latest measurement at sample time \( N + 1 \) to obtain \( \hat{\theta}(N + 1) \). The basic RLS algorithm is presented in Theorem 3.2
Theorem 3.2. Recursive least-squares estimation (RLS) [122]: Assume the regression matrix $\Phi$ has full rank so that $\Phi^T\Phi$ is nonsingular $\forall t \geq t_0$. Given $\hat{\theta}(t_0)$ and $(P(t_0) = \Phi^T(t_0)\Phi(t_0))^{-1}$, the least-squares estimate $\hat{\theta}$ satisfies the recursive equations:

$$\hat{\theta}(N+1) = \hat{\theta}(N) + K(N) \left( y(t) - \varphi^T(t)\hat{\theta}(N) \right) \quad (3.13)$$

$$K(N) = P(N+1)\varphi(N+1)$$

$$= P(N)\varphi(N+1) \left( I + \varphi^T(N+1)P(N)\varphi(N+1) \right)^{-1} \quad (3.14)$$

$$P(N+1) = P(N) - P(N)\varphi(N+1) \left( I + \varphi^T(t)P(N)\varphi(N) \right)^{-1} \varphi^T(t)P(t-1)$$

$$= \left( I - K(N)\varphi^T(N+1) \right) P(N) \quad (3.15)$$

The matrix $P$ is the inverse of the covariance matrix. When the loss function (3.6) is used, all data points are weighted equally. But, for time varying systems, historical data may give false information about the current state of a system. Therefore it is essential to discard old data. This can be achieved using exponential weighting [122]:

$$J_{LS}(\theta) = \sum_{k=1}^{N} \lambda^{N-k} \left( y(k) - \varphi^T(k)\theta \right) \quad (3.16)$$

where the forgetting factor $\lambda \leq 1$ determines how fast old data is forgotten. The least squares estimate with forgetting factor is:

$$\hat{\theta}(N+1) = \hat{\theta}(N) + K(N) \left( y(t) - \varphi^T(t)\hat{\theta}(N) \right)$$

$$K(N) = P(N)\varphi(N+1) \left( \lambda + \varphi^T(N+1)P(N)\varphi(N+1) \right)^{-1} \quad (3.17)$$

$$P(N+1) = \left( I - K(N)\varphi^T(N+1) \right) P(N) / \lambda$$

These equations are not numerically well-conditioned. Hence, the UD-RLS algorithm [122] can be used to update the square root of $P$ rather than $P$ as done in (3.17).

Before proceeding with parameter estimation using either OLS, RLS, their extensions or other available techniques, a suitable model structure is specified. The choice of model structure is influenced by a number of factors such as nature of disturbance and the presence of feedback.
3.1.3 Model Structures

Consider the general model structure of (3.1). Let the discrete time transfer functions, $G$ and $H$ be defined as:

$$G(q) = \frac{q^{-l}B(q^{-1})}{A(q^{-1})F(q^{-1})} \quad (3.18)$$

$$H(q) = \frac{C(q^{-1})}{D(q^{-1})A(q^{-1})} \quad (3.19)$$

With polynomials $A$ and $B$ of orders $n_A$ and $n_B$ respectively defined as:

$$A(q^{-1}) = 1 + \sum_{i=1}^{n_A} a_i q^{-i}$$

$$B(q^{-1}) = \sum_{i=1}^{n_B} b_i q^{-i}$$

Polynomials $C$, $D$ and $F$ of orders $n_C$, $n_D$ and $n_F$ respectively are also monic defined similarly to $A$. Commonly used model structures include:

1. Finite Impulse Response (FIR): $n_F = n_A = n_D = 0$

2. Autoregressive with exogenous input (ARX): $n_F = 0$, $n_D = 0$, $n_C = 0$,

3. Output Error (OE): $n_C = 0$, $n_D = 0$, $n_A = 0$,

4. Box-Jenkings (BJ): $n_A = 0$, and

5. Autoregressive Moving Average with exogenous input (ARMAX): $n_D = 0$, $n_F = 0$.

**Finite Impulse Response (FIR)**

The FIR is a special case of (3.3) written as follows:

$$y(t) = \varphi^T(t-1)\theta \quad (3.20)$$

where the process input vector, $\varphi^T(t-1) = [u(t-1) \ u(t-2) \ \ldots \ u(t-n)]$ and model parameters, $\theta^T = [b_1 \ b_2 \ \ldots \ b_n]$. FIR models are characterized by many parameters and do not require detailed understanding of the modelled system properties. Identification can be achieved using the least squares approach and the solution always converges to the global minimum. Higher order FIR models often cause computational
issues in model based control [114]. Other issues with high order FIR models include high variance error (due to high order) and bias (due to truncation) [119].

**Autoregressive with exogenous input (ARX)**

The most widely used parametric model structure is the ARX structure. This structure is:

\[
A(q^{-1})y(t) = B(q^{-1})u(t) + e(t)
\]

Given \( N \) input-output data samples, the unknown parameters can be estimated using OLS by formulating the problem as follows:

\[
Y = \Phi\theta + E
\]

where

\[
Y^T = \begin{bmatrix} y(1) & y(2) & \ldots & y(N) \end{bmatrix} \quad E^T = \begin{bmatrix} e(1) & e(2) & \ldots & e(N) \end{bmatrix}
\]

\[
\Phi = \begin{bmatrix} -y(n) & \ldots & -y(1) & u(n) & \ldots & u(1) \\ -y(n+2) & \ldots & -y(2) & u(n+1) & \ldots & u(2) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ -y(N-1) & \ldots & -y(N-n) & u(N-1) & \ldots & u(N-n) \end{bmatrix}
\]

\[
\theta = \begin{bmatrix} a_1 & a_2 & \ldots & a_n & b_1 & b_2 & \ldots & b_n \end{bmatrix}^T
\]

The least squares estimate seeks to minimise the loss function (3.6). The solution is given as:

\[
\hat{\theta} = \left[ \Phi^T \Phi \right]^{-1} \Phi^T y
\]

\[
= \left[ \sum_{t=n+1}^{N} \varphi^T(t)\varphi(t) \right]^{-1} \left[ \sum_{t=n+1}^{N} \varphi^T(t)y(t) \right]
\]

The bias in ARX models decreases with increasing model order, but high order models can lead to over fitting. Bias can also arise from non zero mean disturbances that are correlated with each other or with the regression matrix. Like FIR, the solution \( \hat{\theta} \)
converges to the global minimum [114].

For meaningful models to be obtained from any data set, the data needs to be informative with respect to the selected model structure. Prediction error methods for parameter estimation are applicable to arbitrary model parametrizations and structures. Hence, the concept of informative data and identifiability are discussed within the prediction error framework.

### 3.1.4 Prediction error methods

Consider the general LTI model (3.1), the output of the process $y(t)$ has both a deterministic part, $u(t)$ and stochastic part, $e(t)$. Hence $y(t)$ is not a stationary signal.

**Definition 3.1.** [114]: A signal $s(t)$ is said to be quasi-stationary if it is subject to the following:

$$
E[s(t)] = m_s(t), \quad |m_s(t)| \leq C, \quad \forall t
$$

$$
E[s(t)s(r)] = R_s(t,r), \quad |R_s(t,r)| \leq C
$$

$$
\lim_{N \to \infty} \frac{1}{N} \sum_{t=1}^{N} R_s(t,t-\tau) = R_s(\tau), \quad \forall \tau.
$$

The expectation, $E$ is with respect to the stochastic component of the signal $s(t)$. The implication is that if the sequence $\{s(t)\}$ is deterministic, then the expectation has no effect and quasi-stationarity then means the sequence $\{s(t)\}$ is bounded if the limit (3.26) exists:

$$
R_s(\tau) = \lim_{N \to \infty} \frac{1}{N} \sum_{t=1}^{N} s(t)s(t-\tau)
$$

If on the other hand, $s(t)$ is a stationery stochastic signal, then $R_s(\tau)$ does not depend on $t$ and the conditions in (3.24) and (3.25) are satisfied. Two signals $s(t)$ and $w(t)$ are said to be jointly quasi-stationary if they are both quasi-stationary and if, in addition, the cross-covariance function $R_{sw}(\tau)$ exists.

$$
R_{sw}(\tau) = \bar{E}s(t)w(t-\tau)
$$

These jointly quasi-stationary signals are uncorrelated if their cross-covariance is identically zero. If the covariance function, $R_s(\tau)$ and cross covariance function, $R_{sw}(\tau)$
exists, then the power spectrum of \( s(t) \) and cross spectrum between \( s(t) \) and \( w(t) \) are defined in (3.28) and (3.29) respectively.

\[
\Phi_s(\omega) = \sum_{\tau=-\infty}^{\infty} R_s(\tau)e^{-i\tau\omega} \quad (3.28)
\]

\[
\Phi_{sw}(\omega) = \sum_{\tau=-\infty}^{\infty} R_{sw}(\tau)e^{-i\tau\omega} \quad (3.29)
\]

If these infinite sums exist, \( \Phi_s(\omega) \) is always real while \( \Phi_{sw}(\omega) \) is a complex-valued function of the spectrum. The real part of \( \Phi_{sw}(\omega) \) is called the cospectrum and the imaginary part called the phase spectrum.

Consider a linear time-invariant SISO process, let the true system be described by rational and proper transfer functions \( G_0(q) \) and \( H_0(q) \):

\[
S : y(t) = G_0(q)u(t) + H_0(q)e(t) \quad (3.30)
\]

Assume also that the true system is controlled with a stabilizing controller \( K(q) \) which is also proper and rational, such that:

\[
u(t) = K(q)[r(t) - y(t)] \quad (3.31)\]

Let the system be identified using the model structure \( M \) parametrized by \( \theta \in \mathbb{R}^d \):

\[
M(\theta) : y(t) = G(q, \theta)u(t) + H(q, \theta)e(t) \quad (3.32)
\]

Given all input/output data up to time \( t \), the one step ahead prediction of \( y(t) \) is uniquely defined by the prediction error framework, (3.2) which can be written as [114]:

\[
\hat{y}(t|t-1, \theta) = W_u(z, \theta)u(t) + W_y(z, \theta)y(t) \quad (3.33)
\]

Where

\[
W_u(z, \theta) = H^{-1}(z, \theta)G(z, \theta), \quad W_y(z, \theta) = [1 - H^{-1}(z, \theta)] \quad (3.34)
\]

Here, the notations of [109] are used:

\[
W(q, \theta) \equiv \begin{bmatrix} W_u(q, \theta) & W_y(q, \theta) \end{bmatrix}, \quad z(t) \equiv \begin{bmatrix} u(t) \\ y(t) \end{bmatrix} \quad (3.35)
\]
Assume also $z(t)$ is quasistationary so that the spectral density matrix, $\Phi_z(\omega)$ is well defined. Then the one step ahead prediction error $\epsilon$ is defined as:

$$
\epsilon(t, \theta) \equiv y(t) - \hat{y}(t|t-1, \theta) = y(t)W(z, \theta)z(t) = H^{-1}(z, \theta)[y(t) - G(z, \theta)u(t)]
$$

(3.36)

Therefore, given a set of $N$ input/output data, a sample estimate is obtained using the least squares prediction error criterion:

$$
\hat{\theta}_N = \arg \min_{\theta \in \Theta} \frac{1}{N} \sum_{t=1}^{N} \epsilon^2(t, \theta)
$$

(3.37)

If

$$
\bar{V}(\theta) \equiv \bar{E}[\epsilon^2(t, \theta)]
$$

(3.38)

where

$$
\bar{E}[f(t)] = \lim_{N \to \infty} \frac{1}{N} \sum_{t=1}^{N} E[f(t)]
$$

(3.39)

then under suitable conditions [109, 114],

$$
\hat{\theta}_N \xrightarrow{N \to \infty} \theta^* \equiv \arg \min_{\theta \in \Theta} \bar{V}(\theta)
$$

(3.40)

If the true system is contained in the model set and the parameter estimates converge asymptotically to the true values, then the parameter error converges to a Gaussian random variable:

$$
\sqrt{N}(\hat{\theta} - \theta_0) \xrightarrow{N \to \infty} N(0, P_0)
$$

(3.41)

where

$$
P_0 = [I(\theta)]^{-1}|_{\theta = \theta_0}
$$

(3.42)

$$
I(\theta) = \frac{1}{\sigma^2} \bar{E}[\psi(t, \theta)\psi(t, \theta)^T]
$$

(3.43)

$$
\psi(t, \theta) = -\frac{\partial \epsilon(t, \theta)}{\partial \theta} = \frac{\partial \hat{y}(t|t-1, \theta)}{\partial \theta}
$$

(3.44)
where $\nabla_\theta \equiv \frac{\partial W(z, \theta)}{\partial \theta}$. The matrix $I(\theta_0)$ is called the information matrix.

### 3.1.5 Identifiability and Informative Data

**Definition 3.2. (Identifiability) [114]**: A parametric model structure $M(\theta)$ is locally identifiable at a value $\theta_1$ if $\exists \delta > 0$ such that, $\forall \theta$ in $\|\theta - \theta_1\| \leq \delta$:

$$W(z, \theta) = W(z, \theta_1) \implies \theta = \theta_1$$

$M(\theta)$ is globally identifiable at $\theta_1$ if the condition holds for $\delta \to \infty$. And $M(\theta)$ is globally identifiable if it is globally identifiable at almost all $\theta_1$.

**Definition 3.3. (Informative Data) [114]**: A quasistationary data set $z(t)$ is called informative with respect to a parametric model set $\{M(\theta), \theta \in D_\theta\}$ if, for any two models $W(z, \theta_1)$ and $W(z, \theta_2)$ in that set,

$$\bar{E} \{[W(z, \theta_1) - W(z, \theta_2)]^2\} = 0 \quad \text{(3.46)}$$

implies

$$W(e^{j\omega}, \theta_1) = W(e^{j\omega}, \theta_2) \text{ for almost all } \omega. \quad \text{(3.47)}$$

### 3.1.6 Rich and exciting signal

**Definition 3.4. (Persistently Exciting regressor) [137]**: A quasi-stationary signal $\psi(t)$ is called persistently exciting (PE) if $\bar{E}[\psi(t)\psi^T(t)] > 0$.

**Definition 3.5. (Richness of a signal) [137]**: A quasi-stationary signal $u(t)$ is sufficiently rich of order $n$ (SRn) if the following regressor is PE:

$$\phi_{1,n}(t) \equiv \begin{bmatrix} u(t-1) \\ u(t-2) \\ \vdots \\ u(t-n) \end{bmatrix} = B_{1,n}(Z)u(t) \quad \text{(3.48)}$$
where
\[
\mathcal{B}_{k,n}(Z) \equiv \begin{bmatrix} z^{-k} & z^{-k-1} & \ldots & z^{-n} \end{bmatrix}^T, \quad \text{for} \quad k \leq n
\]

For open-loop identification, if \(\Delta W_x\) is defined as
\[
\Delta W_x \equiv W_x(z, \theta_1) - W_x(z, \theta_2),
\]
then from (3.35) and (3.30)
\[
[W(z, \theta_1) - W(z, \theta_2)] z(t) = [\Delta W_u + \Delta W_y G_0] u(t) + \Delta W_y H_0 e(t)
\] (3.49)

The smallest degree of richness of \(u\) for informative data can be obtained using (3.46). Bearing in mind the independence between \(u\) and \(e\), condition (3.46) can be expressed as:
\[
\bar{E} \{\Delta W_u + \Delta W_y G_0 u(t)\}^2 = 0
\] (3.50)
\[
E \Delta W_y H_0 e(t) = 0
\] (3.51)

From (3.51), \(\Delta W_y \equiv 0\). Hence finding the smallest degree of richness of \(u\) is equivalent to finding necessary and sufficient conditions on the richness of \(u\) such that:
\[
E[\Delta W_u(t)]^2 = 0 \quad \Longrightarrow \quad \Delta u \equiv 0
\] (3.52)

For closed-loop identification, from (3.35), (3.30) and (3.31),
\[
[W(z, \theta_1) - W(z, \theta_2)] z(t) = KS [\Delta W_u + \Delta W_y G_0] r(t) + H_0 [\Delta W_y - K \Delta W_u] e(t)
\] (3.53)

Here also because of the independence between \(r\) and \(e\), condition (3.46) is equivalent to the following set of conditions:
\[
\bar{E} \{KS [\Delta W_u + \Delta W_y G_0] r(t)\}^2 = 0
\] (3.54)
\[
\bar{E} \{H_0 S [\Delta W_y - K \Delta W_u] e(t)\}^2 = 0
\] (3.55)

Using \(S = (1 - KG_0)^{-1}\), the conditions become:
\[
\Delta W_y \equiv K \Delta W_u,
\] (3.56)
\[
\bar{E} \{K \Delta W_u r(t)\}^2 = 0,
\] (3.57)

Given (3.56), the second condition (3.57) can be written as:
\[
E \{ \Delta W_y r(t) \}^2 = 0
\] (3.58)
For identification from data without any external excitation except for natural disturbances of a process, closed-loop results are required. Hence, the extension of these results for the closed loop noise only excitation are presented.

### 3.1.7 Closed-loop identifiability without external excitation

Consider the closed loop process depicted in Figure 3.1:

$$y(t) = \frac{G_c G}{1 + G_c G} r + \frac{H}{1 + G_c G} e$$  \hspace{1cm} (3.59)

$$u(t) = \frac{G_c G}{1 + G_c G} r - \frac{CH}{1 + G_c H} e$$  \hspace{1cm} (3.60)

The discrete time transfer functions, $G$ and $H$ are defined as in (3.18) and (3.19) and $C$ is defined as follows:

$$G_c(z) = \frac{X(z^{-1})}{Y(z^{-1})},$$  \hspace{1cm} (3.61)

with polynomials $X$ and $Y$ of orders $n_X$ and $n_Y$ respectively defined as:

![Figure 3.1: Closed loop system](image-url)
\[ X(z^{-1}) = x_0 + \sum_{i=1}^{n_x} x_i z^{-i} \]

\[ Y(z^{-1}) = 1 + \sum_{i=1}^{n_y} y_i z^{-i} \]

For closed-loop data without external excitation, \( r(t) = 0 \). Therefore, only condition (3.56) is required. The expressions for \( y(t) \) and \( u(t) \) become:

\[ y(t) = \frac{H}{1 + G_c G} e(t) \quad (3.62) \]

\[ u(t) = -\frac{G_c H}{1 + G_c H} e(t) \quad (3.63) \]

Substituting the transfer function polynomials,

\[ W_y = 1 - H^{-1} = \frac{C - AD}{C} \quad (3.64) \]

\[ W_u = H^{-1} G = \frac{DB}{CF} z^{n_k} \quad (3.65) \]

And condition (3.56) gives the following condition:

\[ \frac{(F_1 A_1 Y + z^{-n_k} B_1 X) D_1}{C_1 F_1 Y} = \frac{(F_2 A_2 Y + z^{-n_k} B_2 X) D_2}{C_2 F_2 Y} \quad (3.66) \]

Taking into account possible pole-zero cancellations, lead to (3.67) of Theorem 3.3.

**Theorem 3.3.** [135]: Assume the general prediction error model with discrete models defined in (3.32) and discrete controller defined in (3.61), then the system can be identified without any external excitation (except for the noise \( e(t) \)) if the following relationship holds among orders of the polynomials:

\[ \max(n_x + n_k - n_F, n_Y - n_B) \]

\[ \geq n_D + \min(n_C + n_F + n_Y, n_A + n_F + n_Y, n_B + n_X) \quad (3.67) \]

This gives condition on identifiability based on controller and process model parameter orders.
CHAPTER 3. BACKGROUND CONCEPTS

3.2 Model Predictive Control

This section presents model predictive control using the state space velocity formulation. First some definitions from discrete state space systems theory are presented before discussing MPC.

3.2.1 State space model and linear velocity form

Consider a system described by the discrete state space model:

\[
\begin{align*}
x(k + 1) &= Ax(k) + Bu(k) \\
y(k) &= Cx(k) + Du(k)
\end{align*}
\]  
(3.68)

**Definition 3.6. (Reachability) [122]:** The matrix pair \((A, B)\) is reachable if it is possible to drive the system (3.68) to any state from any other state in finite time.

The implication here is that, the eigenvalues of \((A + BK)\) can be assigned arbitrarily using a suitable choice of \(K\). To deal with only the unstable part of the system reachability conditions can be relaxed.

**Definition 3.7. (Stabilisability) [122]:** The matrix pair \((A, B)\) is stabilisable if there exists a matrix \(K\) such that \((A + BK)\) is stable.

**Definition 3.8. (Observability) [122]:** The matrix pair \((C, A)\) is observable if it is possible to determine the initial system state from a finite sequence of measurements.

The implication here is that, the eigenvalues of \((A + LC)\) can be assigned arbitrarily using a suitable choice of \(L\). Similarly, a weaker condition to observability is detectability.

**Definition 3.9. (Detectability) [122]:** The matrix pair is detectable if there exists a matrix \(L\) such that \((A + LC)\) is stable.

Stabilisability and detectability do not guarantee that the eigenvalues of \((A + BK)\) or \((A + LC)\) can be arbitrarily assigned respectively. In this thesis, unless otherwise stated, it is assumed that the state space models used are both detectable and stabilisable.

Consider a plant with \(m\) inputs and \(p\) outputs modelled using the linear time invariant
discrete time state space model:

\[
\begin{align*}
x_p(k+1) &= A_p x_p(k) + B_p u(k) + \omega(k) \\
y(k) &= C_p x_p(k) + \eta(k) \\
z_p(k) &= H_p x_p(k) + \eta_h(k)
\end{align*}
\] (3.69)

where \(x_p \in \mathbb{R}^{n_p}\) is the state variable, \(u \in \mathbb{R}^m\) is the manipulated variable, \(y_p \in \mathbb{R}^{p}\) is the output variable, and \(z_p \in \mathbb{R}^l\) is the controlled variable. \(\omega \in \mathbb{R}^{n_p}\) is the disturbance and \(\eta \in \mathbb{R}^{p}\) is the measurement noise associated with the output while \(\eta_h \in \mathbb{R}^l\) is the measurement noise associated with the controlled variable. Assume the output variable is same as the controlled variable, such that \(C_p = H_p\) and \(\eta = \eta_h\). The state space model can be obtained directly using subspace identification, first principle modelling or by converting from other model types.

The velocity form of the model which uses a change in the manipulated variable can be obtained by defining \(\Delta x_p(k) = x_p(k) - x_p(k-1)\) and a new state variable vector, \(x(k) = \begin{bmatrix} \Delta x_p(k)^T & y_p(k)^T \end{bmatrix}^T\). From (3.69), taking the difference operator:

\[
\begin{align*}
\Delta x_p(k+1) &= A_p \Delta x_p(k) + B_p \Delta u(k) + \Delta \omega(k) \\
y(k+1) - y(k) &= C_p \Delta x_p(k+1) + \Delta \eta(k+1) \\
&= C_p A_p \Delta x_p(k) + C_p B_p \Delta u(k) + C_p \Delta \omega(k) + \Delta \eta(k+1)
\end{align*}
\] (3.70)

which can be written in augmented state space format as:

\[
\begin{bmatrix} \Delta x_p(k+1) \\
y_p(k+1) \end{bmatrix} = \begin{bmatrix} A_p & 0_x \\
C_p A_p & I_x \\ 
\end{bmatrix} \begin{bmatrix} \Delta x_p(k) \\
y_p(k) \end{bmatrix} + \begin{bmatrix} B_p \\
C_p B_p \end{bmatrix} \Delta u(k) \\
+ \begin{bmatrix} \Delta \omega(k) \\
0_n \end{bmatrix} \\
y(k) = \begin{bmatrix} 0_y & I_x \end{bmatrix} \begin{bmatrix} \Delta x_p(k) \\
y_p(k) \end{bmatrix} + \eta(k)
\] (3.71)

where the matrices \(0_x, 0_n\) and \(0_y\) are zero matrices of appropriate dimensions and \(I_x\) is an identity matrix of appropriate dimension. For simplicity in the formulation of prediction equations that follow, the disturbance matrices were not included. However, obtaining these disturbance matrices is straightforward as they have the same format as
matrices $\Phi_1$ and $\Delta U$ in (3.80). The resulting state space equations are given as:

$$ x_{k+1} = Ax_k + B\Delta u_k $$
$$ y_k = Cx_k $$  \hspace{1cm} (3.72) $$

where $x_k \equiv x(k)$, the vectors $x_k \in \mathbb{R}^n$, $y_k \in \mathbb{R}^p$ and $\Delta u_k \in \mathbb{R}^m$ are the state, measured output and manipulated variable vectors respectively. Matrices $A$, $B$, and $C$ are defined in (3.71)

The state space model of (3.72) inherits the properties of (3.69). Two of such properties that are of interest in this work are [82]:

1. If the plant model is detectable and stabilisable, then the augmented velocity model is also and detectable and stabilisable. This is apparent when we consider the transfer function of the velocity model:

$$ C(zI - A)^{-1}B = \frac{z}{z-1}C_p(zI - A_p)^{-1}B_p $$  \hspace{1cm} (3.73) $$

2. The eigenvalues of the augmented model are those of the plant model with $p$ eigenvalues on the unit circle. The eigenvalues are given as:

$$ p(\lambda) = \begin{bmatrix} \lambda I - A_p & 0 \\ -C_p A_p & (\lambda - 1)I_p \end{bmatrix} $$
$$ = (\lambda - 1)^pgdet(\lambda I - A_p) $$  \hspace{1cm} (3.74) $$

The state space model equations can be used to obtain system response prediction over a given horizon.

### 3.2.2 Prediction

Let $x_i := x(k+i)$ be the state prediction vector given the current state $x(k = 0) := x_0$, $y_i := y(k+i)$ and the input vector $\Delta u_i := \Delta u(k+i)$. Then for a horizon, $N_p$, the prediction equations can be obtained by considering the evolution of the state equations:
\[ x_1 = A x_0 + B \Delta u_0 \] (3.75)
\[ x_2 = A (A x_0 + B \Delta u_0) + B \Delta u_1 \]
\[ = A^2 x_0 + A B \Delta u_0 + B \Delta u_1 \] (3.76)
\[ x_3 = A^3 x_0 + A^2 B \Delta u_0 + A B \Delta u_1 + B \Delta u_2 \] (3.77)
\[ \vdots \]
\[ x_{N_p} = A^{N_p} x_0 + A^{N_p-1} B \Delta u_0 + \cdots + B \Delta u_{(N_p-1)} \] (3.78)

written in matrix form as:

\[
\begin{bmatrix}
  x_1 \\
  x_2 \\
  \vdots \\
  x_{N_p}
\end{bmatrix}
= 
\begin{bmatrix}
  A & B & 0 & 0 & \cdots & 0 \\
  A^2 & AB & B & 0 & \cdots & 0 \\
  A^3 & A^2 B & AB & B & \cdots & 0 \\
  \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
  A^{N_p} & A^{N_p-1} B & A^{N_p-2} B & A^{N_p-3} B & \cdots & B
\end{bmatrix}
\begin{bmatrix}
  \Delta u_k \\
  \Delta u_1 \\
  \Delta u_2 \\
  \vdots \\
  \Delta u_{(N_p-1)}
\end{bmatrix}
\] (3.79)

\[ X = \Gamma_1 x_0 + \Phi_1 \Delta U \] (3.80)

If the vector of controlled variables, \( Y \) is defined as:

\[ Y = \begin{bmatrix}
  y_1 \\
  y_2 \\
  \vdots \\
  y_{N_p}
\end{bmatrix} = C X = C (\Gamma_1 x_0 + \Phi_1 \Delta U) = \Gamma x_0 + \Phi \Delta U \] (3.81)

Prediction equations (3.80) and (3.82) predict the evolution of the state and measured output respectively. To obtain an optimal sequence of control moves, these prediction equations are used to form an optimisation problem using a cost function.
3.2.3 Cost Function

In MPC, a cost function is used to optimise a performance index while penalising certain behaviour. Because of advancements, existence of analytical solution (unconstrained case) and convergence properties of quadratic programming, quadratic cost functions are more commonly used. An example of a cost function that penalises the control move increments as well as deviation from set-point is:

\[
\sum_{i=1}^{N_p} (r_i - y_i) Q (r_i - y_i) + \sum_{i=0}^{N_c} \Delta u_i^T R \Delta u_i
\]

(3.83)

where \( Q > 0 \) and \( R \geq 0 \) are the weighting matrices and \( N_c \) is the control horizon. Using the predictions of (3.82), the cost function can be written as:

\[
J = ||S - Y||_Q^2 + ||\Delta U||_R^2
\]

(3.84)

where \( ||x||_p^2 = x^T P x \), \( \bar{Q} = \text{diag}(Q) \), \( \bar{R} = \text{diag}(R) \) and \( S = \begin{bmatrix} r_1^T & r_2^T & r_3^T & \ldots & r_{N_p}^T \end{bmatrix}^T \) is the vector of future set-point trajectory. Here it is assumed that beyond the prediction horizon, \( N_p \), there is no change in the manipulated variable i.e. \( \Delta u = 0 \). Several cost functions are possible depending on the requirements for control. Other examples of cost functions include:

\[
J_2 = \sum_{i=1}^{N_p} x_i^T Q x_i + \sum_{i=1}^{N_c} \Delta u_i^T R \Delta u_i
\]

(3.85)

\[
J_3 = \sum_{i=1}^{N_p} (x_{k+i} - x_{ss})^T Q (x_{k+i} - x_{ss}) + \sum_{i=0}^{N_c-1} (u_{k+i} - u_{ss})^T R (u_{k+i} - u_{ss})
\]

\[
+ \sum_{i=1}^{N_c-1} \Delta u_{k+i}^T R \Delta u_{k+i}
\]

(3.86)

The cost function written in the form of (3.84) can be used to solve either a constrained or an unconstrained MPC problem depending on the requirements of the control problem.
### 3.2.4 Unconstrained MPC

In MPC, the aim is to optimize the cost function. For a quadratic cost, the necessary condition for a minimum is when the first derivative $\frac{\partial J}{\partial \Delta U} = 0$:

$$J = (S - Fx_0)^T \bar{Q} (S - Fx_0) - 2\Delta U^T \Phi^T \bar{Q} \Gamma + \Delta U^T (\Phi^T \bar{Q} \Phi + \bar{R}) \Delta U$$  \hspace{1cm} (3.87)

Then the optimal unconstrained control trajectory is obtained by differentiating the cost function and equating to zero:

$$\Delta U = (\Phi^T \bar{Q} \Phi + R)^{-1} \Phi^T \bar{Q} (S - Fx_0)$$  \hspace{1cm} (3.88)

$$\Delta U = -K_d x_0 + K_s S$$  \hspace{1cm} (3.89)

where $K_d = (\Phi^T \bar{Q} \Phi + R)^{-1} \Phi^T \bar{Q} F$ and $K_s = (\Phi^T \bar{Q} \Phi + R)^{-1} \Phi^T \bar{Q}$. Equation (3.89) is a linear state feedback control. Only the initial state at step $k$, $x_0$ and sometimes the set-point trajectory, $S$ change at every time step. Therefore, all the other prediction matrices can be computed off-line. The incremental control and absolute control signal at time $k$ are given in (3.90) and (3.92) respectively.

$$\Delta u(k) = \begin{bmatrix} I_m & 0 & \ldots & 0 \end{bmatrix} \Delta U$$  \hspace{1cm} (3.90)

$$= -k_d x_0 + r_s$$  \hspace{1cm} (3.91)

$$u(k + 1) = u(k) + \Delta u(k)$$  \hspace{1cm} (3.92)

where $k_d$ and $r_s$ are the first $m$ rows of $K_d$ and $K_s S$ respectively, $I_m$ is an identity matrix of order $m$.

### 3.2.5 Constrained MPC

Unlike the unconstrained case, the constrained control problem does not have a solution in closed form. It can be written in the form of a standard quadratic programming problem which can be solved using any of the available methods for solving constrained quadratic programming problems. Constraints typically exist on the manipulated variable, its rates or the output variable as presented in (3.93), (3.94) and (3.95) respectively:
CHAPTER 3. BACKGROUND CONCEPTS

\[ u_l \leq u_k \leq u_h \]  
\[ \Delta u_l \leq \Delta u_k \leq \Delta u_h \]  
\[ y_l \leq y_k \leq y_h \]  

The manipulated variable sequence over the control horizon can be expressed as:

\[
\begin{bmatrix}
    u(k+1) \\
    u(k+2) \\
    \vdots \\
    u(k+N_c)
\end{bmatrix}
= 
\begin{bmatrix}
    I \\
    I \\
    \vdots \\
    I
\end{bmatrix}
\begin{bmatrix}
    u(1) \\
    u(2) \\
    \vdots \\
    u(k-1)
\end{bmatrix}
+ 
\begin{bmatrix}
    I & 0 & 0 & \ldots & 0 \\
    I & I & 0 & \ldots & 0 \\
    \vdots & \vdots & \ddots & \ddots & \vdots \\
    I & I & \ldots & I & I
\end{bmatrix}
\begin{bmatrix}
    \Delta u(k) \\
    \Delta u(k+1) \\
    \vdots \\
    \Delta u(k+N_c-1)
\end{bmatrix}
\]  

(3.96)

The constraints of (3.93) can therefore be written as:

\[-(C_1 u(k-1) + C_2 \Delta U) \leq -U_l\]
\[(C_1 u(k-1) + C_2 \Delta U) \leq -U_h\]  

(3.97)

where \( U_l \) and \( U_h \) are column vectors of \( u_l \) and \( u_h \), respectively. The constraints on increment of manipulated variable can also be written as:

\[-\Delta U \leq -\Delta U_l\]
\[\Delta U \leq -\Delta U_h\]  

(3.98)

where \( \Delta U_l \) and \( \Delta U_h \) are column vectors of \( \Delta u_l \) and \( \Delta u_h \) respectively. And finally, the output constraints can be expressed as:

\[-(\Gamma x_0 + \Phi \Delta U) \leq -Y_l\]
\[(\Gamma x_0 + \Phi \Delta U) \leq Y_h\]  

(3.99)

where \( Y_l \) and \( Y_h \) are column vectors with elements \( y_l \) and \( y_h \) respectively. The constraints in (3.97), (3.98) and (3.99) can be written in the form:

\[M \Delta U \leq \gamma\]  

(3.100)
CHAPTER 3. BACKGROUND CONCEPTS

The constrained MPC problem then becomes:
Minimise
\[ J = \frac{1}{2} \Delta U^T H \Delta U + \Delta U^T F \] (3.101)
subject to
\[ M \Delta U \leq \gamma \] (3.102)
where \( H \) and \( F \) are obtained directly from (3.87), \( M \) and \( \gamma \) are the constraint matrices.

The constrained MPC problem can be solved using a number of available methods for solving quadratic programming problems. Some of such methods include [12, 82]; the dual methods, the primal method (active set), penalty and barrier methods, and Lagrange methods. Before the designed MPC controller is implemented on-line, it is imperative to ensure that the application will not lead to instability. Hence, the issue of closed-loop stability is discussed in the next section.

3.2.6 Stability

Traditionally, practical closed-loop stability in MPC is achieved using the tuning weights and sufficiently long horizons [21]. A common approach of ensuring theoretical closed-loop is to select the tuning weights such that the eigenvalues of the closed loop un-
constrained system \((A - BK_d)\) are located inside a unit disc. However, for the constrained case, since the control law is non-linear, different conditions for stability are required. In this case, the concept of Lyapunov function is used to ensure closed-loop stability [21].

General approach

For the model predictive control problem defined in (3.101). Assume the following conditions are satisfied:

1. Let the terminal state due to the control sequence \( \Delta u(k), \Delta u(k + 1), \ldots \Delta u(k + N_c - 1) \) be \( x(k + N_p) \).
2. At each sampling instance \( k \), there exists a solution such that the cost function (3.101) is minimised subject to constraints (3.102) and terminal constraint \( x(k + N_p) = 0 \).

Then the closed-loop model predictive control system is asymptotically stable. The proof of this is available in MPC literature [82].
CHAPTER 3. BACKGROUND CONCEPTS

Velocity model formats

In most formulations of MPC, if the plant model is stable, then the optimal control sequence will guarantee closed-loop stability if an infinite prediction horizon is used [64]. For an infinite horizon problem, the use of terminal state weight ensures closed-loop stability. However, this is not applicable to the velocity format of MPC formulation. This is because $A^{N_p} \to \infty$ as $N_p \to \infty$. Hence, a modified terminal constraint is used. In the new formulation, the terminal constraint is only on the unstable modes of the composite model [82]. The closed-loop stability conditions in [82] are presented here.

Let $A_s = W_s \Lambda_s V_s$ contain the stable part of the composite system $A$ and $A_u = W_u \Lambda_u V_u$ the unstable part. Where $W_s$ is the right eigenvector, $V_s$ is the left eigenvector and $\Lambda_s$ the eigenvalues for the respective parts of the system. Then for the unstable modes, the terminal constraint is:

$$V_u x(k + N_p) = 0. \quad (3.103)$$

And for the stable part, the predicted state from horizon $N_p$ to an arbitrary future $m$ is:

$$x(k + N_p + m) = A_s^m x(k + N_p) \quad (3.104)$$

Then a terminal state weight can be found from the Lyapunov function $P = A_s^T P A_s + C^T C$ and the output $y(k) = C_s(k)$:

$$\sum_{i=N_p}^{\infty} y(k + i)^T y(K + i) = x(K + N_p)^T P x(K + N_p) \quad (3.105)$$

The modified cost function with the terminal cost then becomes:

$$J = x(k + N_p)^T P x(k + N_p) + Y^T \bar{Q} Y + \Delta U^T \bar{R} \Delta U \quad (3.106)$$

subject to terminal constraint (3.103). This becomes that of finding a feasible solution of the constrained problem. A conflict between the terminal constraint and other inequality constraints may therefore lead to infeasibility.
3.3 Conclusions

In this chapter, some background theory was discussed. First some theory of system identification was presented. Theory on identification from noise only excited were presented. The theory of MPC was then discussed. Issues such as prediction and stability for the linear velocity form of MPC used in this thesis were presented. Some of the necessary ingredients for work done in subsequent chapters of this thesis have been discussed.
Chapter 4

Control of SISO Benchmark Processes

4.1 Introduction

The simplicity and effectiveness of PI/PID control has made its use widespread in industry. Over the past thirty years or so, PI/PID control has continued to gain popularity; it is typically considered to be the first choice of controller for most applications. Over 90% of all industrial control loops are of the PI/PID type [2]. Because of its popularity, ease of implementation, and availability in off-the-shelf hardware, practitioners are more comfortable with this control strategy. Initially neglected by the research community, PID controllers have received renewed attention during the last two to three decades [2]. This interest in PID has seen the emergence of many new tuning methods [2,9,28,31]. Despite the vast literature on PID tuning, a significant percentage of controllers in automatic mode are poorly tuned [3]. Hence, optimal performance is not always attained. The need for high quality products, reduced energy consumption (fuel and electricity), increasing market competition, lower cost and legislation to cut down emissions, make the need for improved process control performance imperative.

A question often neglected is whether such industrial processes are actually suitable for control by PI/PID or they require a more complex/advanced controller. In other words, when is it more advantageous to use a more complex controller than PID? For processes not best suited for PID control, can the use of an alternative controller improve the loop performance? This chapter seeks to address these questions by comparing the performance of MPC and PID using numerical simulations. MPC will be used as an alternative controller to PI/PID. Before proceeding any further, it is important to clarify that, the intention is not to suggest that PID should be replaced
totally by another controller. **PID** has been very successful and will continue to play a key role in process manufacturing and other industries. It is also clear that for multivariable constrained systems, **MPC** offers many benefits. The focus of this study was to determine if there were benefits in replacing PID controllers applied to SISO systems with MPC and what characteristics of the loop i.e. dynamics dictate the benefits of MPC.

In this Chapter a comparison of PID and MPC is presented. Suitable SISO benchmark models were identified and these control strategies were applied to them. The performance of these controllers were then compared. In the next section, the formulation of control problem is presented. The process models used for the study are briefly introduced in Section 4.3. The introduced process models are then discussed in more details and control techniques applied to them in Sections 4.4 – 4.6. Finally, the chapter is concluded in Section 4.7.

### 4.2 Problem formulation

To characterise the control problem, consider Figure 4.1; the process is considered to be operating at steady state. The system can be represented by a linear transfer function $G(s)$ and disturbance model $H(s)$. The aim is to consider a regulator $G_c(s)$ that will reject disturbance $d(t)$ while maintaining output $y(t)$ at set-point $y_{sp}(t)$. The input and output of $G_c(s)$ are $e(t)$ and $u(t)$ respectively. The sensor transfer function $S(s)$ is assumed to be 1. For input load disturbance, $G(s) = H(s)$.

![Figure 4.1: Feedback representation of control problem](image-url)
Efficient rejection of load disturbance is of primary importance in most process control applications. However, traditional controller tuning methods often focus on set-point tracking. Moreover, there are applications such as motion control in which set-point is of primary importance. Hence in this work both set-point and load disturbance rejection were considered. The IAE presented in (4.1) was used to measure output performance. While for input performance, the total variation (TV) of the input presented in (4.2) was used. The IAE was computed bearing in mind that the system cannot respond to $u(t < \theta)$. TV gives a very good indication of controller activity. Both parameters give quantitative assessment of controller performance. However, observation of the system response is important before conclusions can be reached.

$$IAE = \int_\theta^{T_{ss}} |e(t)| \, dt$$
$$= \sum_{i=0}^{T_{ss}/T_s} e(i) T_s$$
$$= \frac{T_{ss}}{T_s} \sum_{i=0}^\infty e(i)$$

(4.1)

$$TV = \sum_{i=1}^{T_{ss}} |u_i - u_{i-1}|$$
$$= \sum_{i=1}^{T_{ss}} |\Delta u(i)|$$

(4.2)

4.3 Complex processes

The performance achievable by most control systems is limited by a number of factors; process dynamics, disturbances, uncertainties, constraints on MV and non-linearities. In this study, while some of these factors were considered, the main focus was on process dynamics. Some benchmark process models for studying PID performance were identified in [6]; a number of these will be considered here. These processes generally have some challenging dynamics for control and are therefore referred to in this work as complex processes. Th studied processes include:

**Dead-time dominant Processes:** These are systems considered to have significant time delay between the inputs and outputs. Because of the prevalence of dead-time in the process industries, both FOPDT and SOPDT model representations are very common in process control literature [31]. These models are presented
in (4.3) and (4.4) respectively:

\[
G_\theta(s) = \frac{K}{1 + s\tau} e^{-\theta s} \quad (4.3)
\]

\[
G_\theta(s) = \frac{K(\tau_3 s + 1) e^{-\theta s}}{(\tau_1 s + 1)(\tau_2 s + 1)} \quad (4.4)
\]

Detailed discussion of these process models is given in Section 4.4

**Higher order systems:** Another class of systems often used to test controller performance is higher order systems. Some commonly used models for this class of systems are given in (4.5) and (4.6) [6]:

\[
G_n(s) = \frac{1}{(s+1)^n}; \quad n = 1, 2, 3, 4, 8 \quad (4.5)
\]

\[
G_\alpha(s) = \frac{1}{(s+1)(1+\alpha s)(1+\alpha^2 s)(1+\alpha^3 s)}; \quad \alpha = 0.1, 0.2, 0.5, 1.0 \quad (4.6)
\]

Detailed discussion of these models and difficulty associated with their control is presented in Section 4.5

**Systems with both slow and fast dynamics:** This is another class of systems that are also difficult to control due to the presence of both slow and fast dynamics. A commonly used process model in this class is given in (4.7) [6]:

\[
G_{FS} = \frac{100}{s + 10^2} \left( \frac{1}{s + 1} + \frac{1}{s + 0.05} \right) \quad (4.7)
\]

Detailed discussion on the dynamics and control difficulty of these processes is presented in Section 4.6

Other processes not discussed here include under-damped systems, systems with inverse response, disturbance and any combination of the aforementioned dynamics.

### 4.4 Dead-time Dominant Processes

Dead-time dominant systems are very common in process control applications [138]. Dead-time is generally caused by the time required to transfer energy, material or
information in a process. Another common cause of dead-time is a cascade connection of processes in series. Dead-time leads to deterioration of process performance and also reduces the robustness margin of systems. The difficulty associated with control of dead-time dominant systems and their presence in the process industry has generated a lot of interest and research over the years [139, 140]. Most model-based control tuning methods often use either the FOPDT or SOPDT model representations which have dead-time embedded in them. The emphasis here was on the effects of dead-time on PID and MPC.

### 4.4.1 The effect on control performance

Consider the system described by (4.3). The controllability ratio \( \kappa \) (also known as normalised dead-time) is a measure of how difficult the control problem is. It gives a normalised factor for determining when a process is considered to be dead-time dominant. A process is considered to be dead-time dominant if \( \kappa \) is close to 1 but lag-dominant when \( \kappa \) is closer to 0, and a commonly used rule of thumb is to consider a system as dead-time dominant when \( \kappa > \frac{2}{3} \) i.e \( \theta > 2\tau \) [139].

\[
\kappa = \frac{\theta}{\theta + \tau} \quad (4.8)
\]

The pure dead-time component of the system model can be represented by its magnitude and phase in the frequency domain as follows:

\[
|e^{-\theta j\omega}| = 1 \quad (4.9)
\]
\[
\angle e^{-\theta j\omega} = -\theta \omega \quad (4.10)
\]

Dead-time therefore adds a phase of \(-\theta \omega \text{ rad}^{-1}\). This results in a rapid decrease in phase with increasing frequency. This in turn decreases the stability margin as the gain margin becomes smaller. As an illustration, consider the bode plot for a FOPDT system with \( K = 1, \tau = 1 \) and varying controllability ratio shown in Figure 4.2. For this example, the FOPDT system in (4.3) with \( \theta \) varied according to (4.11) was used.

\[
\theta : \kappa = 0, 0.1, 0.3, 0.5, 0.7, 0.9 \quad (4.11)
\]

The system without dead-time (i.e. \( \kappa = 0 \)) has an infinite gain margin. The gain margin
however decreases rapidly with increasing $\kappa$ as shown in Table 4.1 and illustrated graphically in Figure 4.2. This effectively decreases the attainable speed of the closed loop response.

For $\kappa > \frac{2}{3}$, PI control can be used to obtain offset free control. But to avoid oscillatory response and instability, low proportional gain and large integral time are required. This will result in a system response with closed-loop settling time that is larger than the open-loop value. This problem can be reduced using derivative action. The derivative action can be tuned to improve system phase margin, thereby alleviating (or at least reducing) the problem [139]. However, this may only be sufficient for low values of $\kappa$ or slower closed loop responses. Additional increase in controller
Table 4.1: Gain margin and phase cross-over frequency variation with κ

<table>
<thead>
<tr>
<th>κ</th>
<th>GM (dB)</th>
<th>ω_c (rads⁻¹)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>∞</td>
<td>–</td>
</tr>
<tr>
<td>0.10</td>
<td>23.39</td>
<td>14.75</td>
</tr>
<tr>
<td>0.20</td>
<td>16.82</td>
<td>6.86</td>
</tr>
<tr>
<td>0.30</td>
<td>12.72</td>
<td>4.21</td>
</tr>
<tr>
<td>0.40</td>
<td>9.63</td>
<td>2.86</td>
</tr>
<tr>
<td>0.50</td>
<td>7.09</td>
<td>2.03</td>
</tr>
<tr>
<td>0.60</td>
<td>4.92</td>
<td>1.45</td>
</tr>
<tr>
<td>0.70</td>
<td>3.05</td>
<td>1.01</td>
</tr>
<tr>
<td>0.80</td>
<td>1.50</td>
<td>0.64</td>
</tr>
<tr>
<td>0.90</td>
<td>0.41</td>
<td>0.32</td>
</tr>
</tbody>
</table>

order can improve the problem. But, this is not a practical solution and the choice of controller order becomes an open problem. Moreover, tuning will be an issue as order of controller increases. Even PID (with only 3 parameters) has a limited presence in industry due to difficulty associated with tuning the derivative part [2]. Therefore, a more reasonable solution is to use some form of dead-time compensation or prediction. Although, a lot of research has been done on dead-time compensation, it has not found much application in the industry [19]. The predictive capability inherent to MPC enables it to cater for process dead-time systematically.

### 4.4.2 PID for time delayed processes

Numerous techniques for tuning PID Controllers exist. Here we consider two such techniques to tune the controller for comparison with the MPC. Whenever a reasonable model of a process is available, model based control approaches may provide improved control performance. IMC [141], for example, is known for its ability to handle un-modelled dynamics and process uncertainties [138, 142]. Although IMC is not routinely applied to regulate process systems, it is now used extensively as a tuning tool for PID controllers [143]. The advantage it has over more traditional tuning methods, such as Ziegler-Nichols, is that the PID parameters are specified to produce desired closed-loop dynamics. However, for processes with significant time delays, the performance of PID regulators, tuned using the IMC method decreases because of modelling errors introduced by approximations made to the dead-time. A summary of
IMC controller tuning is presented in [144].

In this work, the PID controllers were tuned using the IMC method, as this is consistent with what is now routinely implemented in industry, and also using an iterative technique that identified the optimal PID parameters which gave the minimum mean square error (MSE) for a set-point change. The latter approach is typically not suitable for real applications as it tends to produce controllers with aggressive behaviour. However, it was used in this study as it provides an upper measure of the performance achievable with PID.

The iterative tuning approach used a GA to identify the optimal PID parameters. A GA is a search algorithm inspired by the theory of evolution [145], which as a result of its parallel search approach, has good speed of convergence. The performance of a GA depends on the values of various parameters. The values used for these parameters in this work are shown in Table 4.2. This is to limit the search space and allow for adequate variation in offspring population [145]. For the initial conditions, the IMC obtained PID parameters were used.

<table>
<thead>
<tr>
<th>GA parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population Size</td>
<td>20</td>
</tr>
<tr>
<td>Crossover fraction</td>
<td>0.8</td>
</tr>
<tr>
<td>Generations</td>
<td>100</td>
</tr>
<tr>
<td>Fitness function</td>
<td>Mean squared error (MSE)</td>
</tr>
</tbody>
</table>

The desired closed-loop process time constant, $\tau_c$, is key to IMC design, a number of authors suggested different values according to [5, 31, 32, 142]:

$$\frac{\tau_c}{\theta} > 0.8 \quad \text{and} \quad \tau_c > 0.1\tau$$  \hspace{1cm} (4.12)

$$\tau_c = \theta$$  \hspace{1cm} (4.13)

$$\tau > \tau_c > \theta$$  \hspace{1cm} (4.14)

where $\tau$ and $\theta$ are the process time constant and dead-time respectively.

For dead-time dominant processes, the expression in (4.14) is not applicable for values
of $\frac{\theta}{\tau} > 1$, as it violates the relationship when $\theta > \tau$. Seborg [31] suggested a value of closed loop time constant, $\tau_c = \frac{\tau}{3}$, but for dead-time dominant processes this will lead to aggressive control action. Hence, for this work the following guidelines, based on (4.12) and (4.13) were used. If the lower and upper constraints on $\tau_c$ are defined as $\tau_{c_{\text{min}}}$ and $\tau_{c_{\text{max}}}$ respectively. Then,

$$\tau_c = \max\left(\frac{\tau}{3}, \theta\right)$$  \hspace{1cm} (4.15)

$$\tau_{c_{\text{min}}} = \max(0.1\tau, 0.5\theta)$$  \hspace{1cm} (4.16)

$$\tau_{c_{\text{max}}} = 2\tau_c\hspace{1cm}$$  \hspace{1cm} (4.17)

Given the process defined in (4.3), using the expressions for IMC-based PID obtained from [31], the following IMC-based PI controller parameters were obtained:

$$K_p = \frac{\tau}{K(\tau_c + \theta)}$$  \hspace{1cm} (4.18)

$$K_I = \frac{K_p}{\tau}$$  \hspace{1cm} (4.19)

The limits for the parameters based on (4.16) and (4.17) will then be given as:

$$K_{p_{\text{min}}} = \frac{\tau}{K(\tau_{c_{\text{max}}} + \theta)}$$  \hspace{1cm} $K_{p_{\text{max}}} = \frac{\tau}{K(\tau_{c_{\text{min}}} + \theta)}$  \hspace{1cm} (4.20)

$$K_{I_{\text{min}}} = \frac{K_I}{3}$$  \hspace{1cm} $K_{I_{\text{max}}} = 3K_I$  \hspace{1cm} (4.21)

$$K_{D_{\text{min}}} = \frac{K_D}{\tau}$$  \hspace{1cm} $K_{D_{\text{max}}} = 3K_D$  \hspace{1cm} (4.22)

For the PI controller, the constraints in (4.20) and (4.21) define the search space for the GA algorithm. For the second order plant defined in (4.4) the IMC-tuning parameters were selected using the expressions in (4.23) – (4.25) [31]. The limits for the GA parameters can be computed using (4.20) – (4.22) and substituting $\tau^2 = \tau_1\tau_2$.

$$K_p = K\frac{\tau_1 + \tau_2 - \tau_3}{\tau_c + \theta}$$  \hspace{1cm} (4.23)

$$K_I = \frac{K_p}{\tau_1 + \tau_2 - \tau_3}$$  \hspace{1cm} (4.24)

$$K_D = K_p\frac{\tau_1\tau_2 - (\tau_1 + \tau_2 - \tau_3)\tau_3}{\tau_1 + \tau_2 - \tau_3}$$  \hspace{1cm} (4.25)
4.4.3 MPC for time delay process

The prediction horizon, \( N_p \), used in the MPC cost function was selected to be approximately equal to the settling time of the process as shown in (4.26). A control horizon, \( N_c \) of 3 was used in all simulations. An output weighting of 1 was used. Whilst it was possible to improve the performance of the MPC by adjusting these parameters, these values were selected to give an indication of the performance that was achievable using MPC. Moreover, the aim is to use a controller with fewer tuning parameters. For many processes, no significant improvement is obtained beyond \( N_c = 3 \) [68].

\[
N_p = \left( \frac{\theta + 5\tau}{T_s} \right)
\]  

(4.26)

Where \( T_s \) is the sampling period. With these choices of \( N_p \) and \( N_c \), MPC tuning can be achieved using the weighting, \( r_w \).

In this work the performance of the control systems was quantified using the IAE, defined by (4.1), where \( e \) is the difference between the set-point and output between the time delay, \( \theta \), and settling time, \( t_s \), of the process, following a unit step change in set-point.

4.4.4 Numerical simulations

To compare the performance of the PID and MPC controllers, the models defined in (4.3) and (4.4) were used with three different sets of model parameters. To begin, a FOPDT system with \( K = 2 \) and \( \tau = 1 \) was used. The ratio of the process time delay to time constant, \( \frac{\theta}{\tau} \) was varied over a range of values by varying \( \theta \) over the range shown in (4.27):

\[
\frac{\theta}{\tau} = [0 : 10]
\]

(4.27)

The corresponding models were then used to tune PID and MPC controllers using the methods described in sections 4.4.2 and 4.4.3 respectively. Following the design of the controllers, unit set-point step changes were made and the performance of the system evaluated using IAE as the performance measure. Two different MPC controllers were tuned. An aggressive MPC controller and a more conservative controller labelled as \( MPC_1 \) (with \( r_w = 0.1 \)) and \( MPC_2 \) (with \( r_w = 100 \)) respectively.

The controllers designed for the base case dynamics were also implemented when the plant had a 5 %, 10 % and 20 % mismatch in process model gain, \( K \), and \( \frac{\theta}{\tau} \).
A white noise with a signal-to-noise ratio of 20 was added to the output measurements and a sampling time of $T_s = 0.2s$ was used. The results of these are shown in Figures 4.3 – 4.4.

Figure 4.3: Plots of IAE against $\frac{\theta}{\tau}$ for $G(s) = \frac{2e^{-\theta s}}{s+1}$.
These figures show that in the case where there is no plant-model mismatch, the performance of PID and MPC for relatively small time delays is comparable. However, as the time delay increases, the performance of PID degrades sharply. In all cases, the performance of MPC and PID was only affected slightly by the increase in plant-model mismatch. Furthermore, the increase in the dead-time to time constant ratio was seen to degrade the performance of the PID controllers significantly, whereas for MPC the effect, was as expected, minimal. Sample responses are shown in Figures 4.5 – 4.6.
Figure 4.5: Sample plots of manipulated and control variables for $G(s) = \frac{2e^{-\theta s}}{s+1}$
Figure 4.6: Sample plots of manipulated and control variables for $G(s) = \frac{2e^{-\theta s}}{s+1}$

(a) 20% gain mismatch, $\frac{\theta}{T} = 2$

(b) 20% gain mismatch, $\frac{\theta}{T} = 8$
In a second study, PID and MPC controllers were applied to another first order system with $K = 1$ and $\tau = 7$. The performance of the controllers were analysed using both time constant and gain mismatch as in the first study; white noise with signal-to-noise ratio of 20 was applied. A sampling time of $T_s = 1$ s was used. The plots of the IAE are shown in Figures 4.7 – 4.8. In this study, the performance trend is consistent to that of the previous study, with significant improvements in control observed with MPC when the ratio of time delay to time constant exceeds a value of approximately 2. This is consistent with dead-time compensation results, which suggests an improvement in performance with dead-time compensation when $\frac{\theta}{\tau} > 1$, [146].

Figure 4.7: Plots of IAE against $\frac{\theta}{\tau}$ for $G(s) = \frac{e^{-\Theta s}}{Ts + 1}$
In the final study, the controllers were tuned for a second order system with parameters: $K = 2$, $\tau_1 = 3$, $\tau_2 = 10$ and $\tau_3 = 0$. PID and MPC controllers were applied, as before, with 5%, 10% and 20% mismatch in process gain and $\theta / \tau$; and measurement noise also white with signal-to-noise ratio of 20. A sampling time of $T_s = 2 \text{s}$ was used. The plots of the IAE against $\theta / \tau$ are shown in Figures 4.9 – 4.10. As with the first two cases, the performance of MPC was maintained as $\theta / \tau$ increases. Furthermore, the performance of the IMC tuned PID controller degrades very quickly even for very small time delays. This is an important result as it suggests that for industrial processes, which will almost certainly be of high order, even for very small delays, there may be significant benefit in using MPC to regulate SISO systems.
CHAPTER 4. CONTROL OF SISO BENCHMARK PROCESSES

4.4.5 Summary

In this section, a study into the effect that process time delay has on the performance of PI/PID and MPC controllers was conducted. The study has shown that for the two first order systems investigated, the performance of the PI controller tuned using IMC degraded almost linearly with the time delay and when the delay exceeded approximately twice the time constant, MPC was found to provide much improved performance. However, for the second order system, the IMC tuned PID controller was found to be much more sensitive to the time delay and with the time delay exceeding approximately 10% of the time constant, the performance of MPC was found to be significantly better than PID. The optimally tuned PID controller produced slightly improved results compared with the IMC tuned PID controller. However, it should
be noted that the optimally tuned PID controller is unlikely to be acceptable in an industrial application as it is too aggressive. Additional studies were carried out and comparison was done with an ideal smith predictor. However, since the aim here is for comparison with standard PID, the results for this can be found in Appendix A.1.

4.5 Higher Order Systems

Higher order systems are also commonly found in the process industry. Typical system models that have been studied are given in (4.5) and (4.6). The model in (4.5) is commonly used by controller manufacturers to test controller performance [5]. For illustration purposes, the bode plots for $G_n$ and $G_\alpha$ with varying $n$ and $\alpha$ are shown.
in Figures 4.11 and 4.12 respectively. The resulting gain margins and phase crossover frequencies are shown in Table 4.3. $G_n$ has an infinite gain margin for \( n = 1, 2 \). Therefore, for \( n = 1, 2 \), good performance can be achieved by PI and PID controllers respectively. The gain margin decreases with increasing \( n \) reaching a value of 12.08 dB with crossover frequency of 1 $\text{rads}^{-1}$ when \( n = 4 \). This leads to decrease in system stability margin and at a much lower cross over frequency. More so, the phase crossover frequency also becomes lower. Control difficulty therefore increases with increasing \( n \) and achievable performance with PI/PID is limited when \( n > 3 \). For larger values of \( n \) the behaviour of the system is similar to that of a system with dead-time. The process can be considered as a cascade of \( n \) first order systems. Hence, the attainable closed loop speed with simple controllers is limited.

![Bode plot of $G_n(s) = \frac{1}{(s+1)^n}$ for different values of $n$](image)

Figure 4.11: Bode plot of $G_n(s) = \frac{1}{(s+1)^n}$ for different values of $n$
Figure 4.12: Bode plot of $G_\alpha = \frac{1}{(s+1)(1+\alpha s)(1+\alpha^2 s)(1+\alpha^3 s)}$ for different values of $\alpha$

Table 4.3: Gain margin and phase cross-over frequency variation with $n$ and $\alpha$

<table>
<thead>
<tr>
<th>$n$</th>
<th>$\alpha$</th>
<th>$GM$ (dB)</th>
<th>$\omega_p$ (rads$^{-1}$)</th>
<th>$\alpha$</th>
<th>$GM$ (dB)</th>
<th>$\omega_p$ (rads$^{-1}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\infty$</td>
<td>-</td>
<td>0.10</td>
<td>40.84</td>
<td>31.62</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>$\infty$</td>
<td>-</td>
<td>0.20</td>
<td>29.61</td>
<td>11.18</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>18.06</td>
<td>1.73</td>
<td>0.50</td>
<td>16.59</td>
<td>2.82</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>12.04</td>
<td>1.00</td>
<td>1.00</td>
<td>12.04</td>
<td>1.00</td>
<td></td>
</tr>
</tbody>
</table>

The system of (4.6), $G_\alpha$ has four poles whose relative position is determined by $\alpha$. The system has a gain margin and phase cross-over frequency of 40.84 dB and 31.62 rads$^{-1}$ respectively when $\alpha = 0.1$. These values decrease to 12.04 dB and 1 rads$^{-1}$ at $\alpha = 1$. 
This decrease in stability margin and phase cross over frequency with increasing $\alpha$ limits the attainable closed loop speed with a simple controller like the PID. Hence, as $\alpha$ increases from 0.1 to 1, control difficulty also increases.

For PI/PID, an assumption was made that the process models were available. And the half rule model approximation methods proposed in [5] were used to obtain the equivalent FOPDT or SOPDT models as required. The resulting models were then used to design PID/PI controllers using IMC tuning. The guidelines provided in (4.12) – (4.14) were used to determine the desired closed loop time constant, $\tau_c$. For MPC, system identification steps were carried out to determine suitable models for control. The MPC was tuned using a prediction horizon that is equal to the process settling time and a control horizon of $N_c = 3$. The weighting $r_w$ was used to tune the controller. Normalised values of IAE and TV as defined in (4.28) were used as performance measures.

$$x_{\text{normalised}} = \frac{x}{\text{min}(x)} \quad (4.28)$$

where $x$ is a performance measure vector whose elements are the performance measures for the different controllers, $\text{min}(x)$ is the minimum of the performance measure vector. To verify the effects of constraints on performance, constraints on $u$ and $\Delta u$ were also considered. In this study, the optimally-tuned controller was not tuned as it is not realistic to use in industrial applications.

### 4.5.1 Numerical Simulations

For the process $G_n$, simulations were carried out for $n = 1, 2, 3, 4, 8$. First and second order approximations of the process models were used to design PI and PID controllers respectively. To allow for easy incorporation of constraints, the velocity form of PID with anti-windup was used. Set-point weighting was also used to improve controller performance. MPCs were also tuned for the respective processes. To test controller performance, unit step set-point and load disturbances were applied at $t = 1$ s and $t = t_2$ s (as indicated in the plots) respectively. Both constrained and unconstrained controllers were considered. The constraints used for the manipulated variable and its rate were $-3 \leq u \leq 3$ and $-1 \leq \Delta u \leq 1$ respectively. The normalised performance measures for the unconstrained and constrained cases are presented in Tables 4.4 and 4.5 respectively. The resulting plots of the controlled and manipulated variables are shown in Figures A.2 – A.6 in Appendix A.
Table 4.4: Controller Performance for $G_n$ (unconstrained)

<table>
<thead>
<tr>
<th>$n$</th>
<th>Normalised IAE</th>
<th>Normalised TV</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PI</td>
<td>PID</td>
</tr>
<tr>
<td>1</td>
<td>1.00</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>1.90</td>
<td>1.00</td>
</tr>
<tr>
<td>3</td>
<td>2.01</td>
<td>1.00</td>
</tr>
<tr>
<td>4</td>
<td>2.12</td>
<td>1.36</td>
</tr>
<tr>
<td>8</td>
<td>2.47</td>
<td>2.12</td>
</tr>
</tbody>
</table>

Table 4.5: Controller Performance for $G_n$ (constrained)

<table>
<thead>
<tr>
<th>$n$</th>
<th>Normalised IAE</th>
<th>Normalised TV</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PI</td>
<td>PID</td>
</tr>
<tr>
<td>1</td>
<td>1.00</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>1.06</td>
<td>1.00</td>
</tr>
<tr>
<td>3</td>
<td>1.95</td>
<td>1.00</td>
</tr>
<tr>
<td>4</td>
<td>2.05</td>
<td>1.31</td>
</tr>
<tr>
<td>8</td>
<td>2.40</td>
<td>2.06</td>
</tr>
</tbody>
</table>

In the unconstrained case, for $n = 1$, the process has first order dynamics with dead-time, $\theta = 0$. Hence, a PID controller was not applied. PI controller gave lower values of both IAE and TV. For $n = 2$, PID had the lowest IAE value. And as $n$ increased, the performance of PID decreased while MPC maintained its performance due to its predictive capabilities. When $n > 3$, MPC outperformed PID. Observing plots of system response reveal that; for lower values of $n$, the controllers had similar set-point response performance. But as $n$ became larger, the set-point performance of PI/PID decreased while the performance of MPC was maintained. On the other hand, PI/PID controllers had better disturbance rejection performance for lower values of $n$. This performance also decreased with increasing values of $n$. The performance trend is expected because of increase in control difficulty with increasing $n$. In all cases, MPC had larger values of TV. This is because MPC can take large but calculated moves thereby ensuring smooth control despite the relatively larger values of TV.

In the constrained case, the effects of constraints were more apparent at lower values of
For values of $n \leq 2$, there were improvements in MPC performance over the unconstrained case. For $n \geq 3$, constraints had very little effect on IAE. With the exception of $n = 1$, TVs for MPC for the constrained case were lower than the corresponding unconstrained values. This was as a result of relatively limited controller activity due to constraints.

Simulations were also carried out for $G_\alpha : \alpha = 0.1, 0.2, 0.5, 1^1$. First and second order approximations of the process were used to design PI and PID controllers respectively. Unit step set-point and load disturbances were also applied. Constrained and unconstrained MPCs were also implemented. The constraints of $-2 \leq u \leq 2$ on MV and $-1 \leq \Delta u \leq 1$ on its rate were used. Normalised performance measures for the constrained and unconstrained cases are presented in Tables 4.6 and 4.7 respectively. The resulting plots are shown in Figures A.8 – A.10 in Appendix A.

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>Normalised IAE</th>
<th>Normalised TV</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PI</td>
<td>PID</td>
</tr>
<tr>
<td>0.1</td>
<td>1.00</td>
<td>1.02</td>
</tr>
<tr>
<td>0.2</td>
<td>1.22</td>
<td>1.00</td>
</tr>
<tr>
<td>0.5</td>
<td>1.73</td>
<td>1.07</td>
</tr>
<tr>
<td>1.0</td>
<td>2.54</td>
<td>1.63</td>
</tr>
</tbody>
</table>

Table 4.7: Controller Performance for $G_\alpha$ (constrained)

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>Normalised IAE</th>
<th>Normalised TV</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PI</td>
<td>PID</td>
</tr>
<tr>
<td>0.1</td>
<td>1.61</td>
<td>1.65</td>
</tr>
<tr>
<td>0.2</td>
<td>1.52</td>
<td>1.49</td>
</tr>
<tr>
<td>0.5</td>
<td>1.40</td>
<td>1.12</td>
</tr>
<tr>
<td>1.0</td>
<td>2.30</td>
<td>1.51</td>
</tr>
</tbody>
</table>

In the unconstrained case, for $\alpha = 0.1$, the controllers had similar IAE values. When $\alpha = 0.2$, PID had the smallest IAE while PI had the largest. However, values of IAE for

$^1G_n = G_\alpha$ for $\alpha = 1$ and $n = 4$
both PI and PID increased with increasing $\alpha$. When $\alpha \geq 0.5$, MPC had the lowest IAE. For all values of $\alpha$, MPC had better set-point response performance. The disturbance rejection performance for low values of $\alpha$ was similar across controllers. But when $\alpha \geq 0.5$, MPC showed improvements in disturbance rejection. The TV for MPC was larger for all controllers. This is due to the ability of MPC to make large but cautious moves.

For the constrained case, MPC had a better performance for all values of $\alpha$. However, the performance for $\alpha \geq 0.5$ was lower for the unconstrained MPC. As expected again, the TV for MPC was relatively lower in the constrained case. However, because the effect on controller activity was larger in PI controller, the normalised TV was larger for MPC for the values of $\alpha = 1, 2$. Trying to obtain faster response with the PID resulted in oscillatory response especially in the unconstrained case. This is because of the relatively lower stability margins at higher values of $\alpha$.

### 4.5.2 Summary

In this section, PI/PID controllers were tuned for $G_n$ and $G_\alpha$ using IMC tuning. Similarly, MPC was also implemented and the results compared. Generally, for the system $G_n$, similar set-point performance was achieved by the controllers when $n \leq 3$, but as $n$ increased MPC achieved better set-point performance. PID had better disturbance rejection capabilities at lower values of $n$. But, when $n \geq 4$, MPC achieved better disturbance rejection. MPC performance improved with the introduction of constraints. The effects of these constraints were more pronounced at lower values of $n$. In the case of $G_\alpha$, MPC had better set-point response performance for all values of $\alpha$. However, for load disturbance rejection, PID showed better performance with values of $\alpha < 0.5$.

As the system dynamics become more complex (i.e. when $\alpha$ become larger), the disturbance rejection for MPC improved relative to PID. The performance for MPC also generally increased with constraints. In all cases, for system $G_\alpha$, PID and MPC had similar performance for low values of $\alpha$. However, performance attainable by PID decreased as $\alpha$ increased. Hence, for the IMC-tuning considered, the performance attainable by PI/PID becomes limited when $\alpha \geq 0.5$ for $G_\alpha$ and when $n \geq 3$ for $G_n$. Hence, the use of MPC is recommended for $G_n : n \geq 4$ and $G_\alpha : \alpha \geq 0.5$. 
4.6 System with fast and slow dynamics

This presents another system that has been used to evaluate the performance of PID controllers [6]. A typical system with such characteristics is:

\[ G_{FS} = \frac{100}{(s+1)^2} \left( \frac{1}{s+1} + \frac{0.5}{s+0.05} \right) \]  \hspace{1cm} (4.29)

This system has both a slow pole and a fast one with time constants 20 and 1 respectively. The slow mode has a large gain of 10 while the fast mode has a gain of 1. To obtain a suitable PI/PID controller using IMC tuning, a low order approximation of the process is required. Therefore, a first order approximation presented in (4.30) was obtained:

\[ G_{fs} = \frac{20.9}{19.3s+1} \]  \hspace{1cm} (4.30)

Unit step responses of \( G_{FS} \) and its first order approximation \( G_{fs} \) are shown in Figure 4.13. These step responses are very similar. Hence the approximation \( G_{fs} \) was used to design a PI controller using IMC-tuning. Controllers with desired closed-loop time constant, \( \tau_c = 0.5, 1 \) and 6.5 were tuned. Attempts to tune a controller with good performance using IMC tuning did not yield good results. The response for \( \tau_c = 1 \) s is shown in Figure 4.14. This may be because the IMC tuning method only used the information captured in the model approximation, \( G_{fs} \), and this model only captures the slow dynamics of the process. But it is the faster time constant that is important for closed loop response [147]. This is illustrated in the bode diagram of \( G_{FS} \) and \( G_{fs} \) which is shown in Figure 4.12. From the bode plot, the first order model was able to capture the dynamics of the system at lower frequencies of less than 1 \( rads^{-1} \) but was unable to do so at higher frequencies. The true system also has a phase cross over frequency of around 10 \( rads^{-1} \) while for the approximation it is around 100 \( rads^{-1} \). Therefore, it may not be possible to design a good PI controller using IMC tuning. Because some important properties of the system are not captured by the step response, tuning methods based on step response are not likely to yield good performance.
CHAPTER 4. CONTROL OF SISO BENCHMARK PROCESSES

Figure 4.13: Open loop step response of $G_{FS}$ and $G_{fs}$

Figure 4.14: Step and load disturbance response with IMC-tuned PI control ($\tau_c = 1\,s$)
To allow for comparison with MPC, the following parameters tuned using non-convex optimisation were obtained from [147]; $K_p = 1$, $K_i = 1.65$. An MPC controller was also designed. In this case a sampling time $T_s = 0.05\, s$, a prediction horizon $N_p = 240$, and control horizon $N_c = 10$ were used. The same sampling time was used for the PI Controller. Plots of unit step set-point and disturbance responses of the process with the PI and MPC are shown in Figure 4.16. The performance of both controllers were quite similar. However, the system is sensitive to controller parameter variation. A slight change in either PI or MPC parameters gave an entirely different performance.

### 4.6.1 Summary

For the system with both fast and slow dynamics $G_{FS}$, the design of both PI controller and MPC with good response was not straight forward. However, if specifically developed methods are used, a good PI controller can be tuned. For the MPC, same
controller design method as that used for dead-time dominant and higher order systems was used but with larger horizons. In general, extra measures were required to design controllers with good performance for this class of systems. If first order approximation of the model are used for controller tuning, then controller performance may be limited. For MPC, the major challenge is the computational requirements as small sampling times are needed to capture the fast dynamics while a large prediction horizon is needed to enable prediction to steady state. For this system, prior knowledge of process dynamics and control engineers experience will likely play an important role in deciding an appropriate control scheme for the process. An engineer having vast experience with MPC may prefer to use MPC while an engineer that is not very comfortable with MPC is likely to use PI.

### 4.7 Conclusions

In this chapter, numerical simulations were used to compare the performance of PI/PID controller (tuned using IMC-tuning) and MPC (tuned using a single tuning parameter).
To ensure simplicity of design and more transparent tuning of MPC, additional features such as set-point trajectory were not used in MPC design. For PID, other features that enhanced the performance of PID were considered in its implementation. The velocity form with anti-windup and set-point weighting was used. In line with industrial practice, the derivative action was applied on the output rather than the error. However, there was an exception in the system with both slow and fast dynamics. For this system a different tuning method was used for PI controller, and larger horizons were used for MPC as the default methods used for other systems did not yield satisfactory results.

With these considerations and tuning procedures described in this chapter, the following observations were made. For dead-time dominant systems, MPC gives a better performance when the ratio $\theta / \tau$ approximately equals 2 or above (or equivalently when the normalised dead-time $\kappa \geq \frac{2}{3}$). For higher order systems, it was observed that for $n = 1$ and $\alpha = 0.1$, PI controller gave good performance and there was no need to consider a more advanced controller. For intermediate values of $n$ and $\alpha$, PID controller gave a better or equivalent performance with MPC. For higher values i.e. $n \geq 4$ and $\alpha \geq 0.5$, MPC gives a relatively better performance. Therefore, with the right infrastructure in place, there are incentives for using MPC in SISO loops that are dead-time dominant and in higher order systems with order of at-least 4. It is also worth noting that tuning the derivative term is not always trivial and therefore PID controllers have a relatively limited presence in the industry. Moreover, the MPC tuning method used in this work is straightforward with only one varying tuning parameter, thereby limiting the performance achievable by MPC.

In conclusion, significant improvements can be obtained in industrial applications by replacing existing PI controllers with PID in suitable processes. However, the challenge associated with tuning the derivative part can still limit the application of PID control. For more complex processes, MPC can be considered to improve loops controlled by PI/PID controllers. The benefits are more in processes with constraints. This is achievable with less tuning efforts when formulations such as the one used in this work are exploited. Hence, MPC is straightforward to apply than PID and gives performance that is at least as good. The limitation is that it still requires dedicated software, but with recent advances, the technology could be integrated with relative ease on to PLCs and other low level control systems. Another major challenge is the effort associated with plant test and model development. However, this is not peculiar to MPC as most controller tuning methods now use some form of model for example IMC tuning of PID. Some of these factors will be considered further in subsequent
To verify and appreciate the simulation results obtained in this chapter, an industrial case study was conducted. The details of this are presented in the next chapter.
Chapter 5

Industrial Case-Study: Brine Dechlorination

5.1 Introduction

This Chapter describes a case-study application at the INEOS ChlorVinyls site in Runcorn. The work was as a result of discussions with Engineers at the Runcorn site. They were considering ways of improving the performance of certain loops in the plant. Initial studies of one of these loops revealed that the associated process had significant dead-time and measured disturbance that is characterised by complex dynamics. Previous studies have shown that benefits can be achieved by replacing PID with MPC in dead-time dominant processes [23]. The results in Chapter 4 further showed that benefits are also achievable in processes with certain characteristics. The case study considered also falls into the category of plants with dynamics that can benefit from small scale MPC because of the significant dead-time and associated complex dynamics. Details are presented in the relevant sections. The study is therefore within the scope of this thesis and will allow for verification of results from Chapter 4 on an industrial process. The plant has very experienced personnel but with little or no experience with MPC. Hence, they were not too comfortable with MPC being applied on the process. And an earlier attempt to implement MPC on the process was not successful. To minimise plant disruptions associated with controller trials, a mechanistic model of the process was used for initial controller tuning before final implementation on the process.

The chapter is structured as follows. In section 5.2, a background on process (industrial
electrolysis of brine) is presented and the role of pH regulation in dechlorination highlighted. Control objectives are defined in Section 5.4 and the MPC formulations used are briefly discussed in Section 3.2. Section 5.5 discusses the use of mechanistic model to quantify benefits achievable by MPC. Section 5.6 focuses on the control of actual plant. Here, controller design, implementation and tuning are presented and benefits obtained analysed. The chapter is finally concluded in section 5.7.

5.2 Problem Background

The main products from a chloro-alkali process are Chlorine Gas (Cl\textsubscript{2}) and caustic soda (NAOH). The production of these products is achieved by the electrolysis of brine using either mercury, membrane or diaphragm cell technologies \[148\]. The process described here uses the membrane technology. A block diagram depicting the processes involved is presented in Figure 5.1.

Salt solution (brine) obtained either from sea water or mining is saturated and purified before it is sent to the electrolysis cells. Cl\textsubscript{2} and hydrogen gas (H\textsubscript{2}) are produced by the electrolysis. Saturated brine at a pH of around 2 flows into the anode section of the electrolytic cell, and water flows into the cathode section. A cation exchange membrane separates the cathode from the anode chamber. A picture of the membrane cell is shown in Figure 5.2.
When electric current is passed through the electrodes. At the anode, chloride ions (Cl\(^-\)) are oxidised to form Cl\(_2\):

\[
2\text{Cl}^{(aq)} \rightarrow \text{Cl}_2(g) + 2e^- 
\]

At the cathode, hydrogen ions (H\(^+\)) are reduced to produce H\(_2\) leaving behind hydroxyl ions (OH\(^-\)):

\[
\text{H}_2\text{O}(l) \rightleftharpoons \text{H}^{+(aq)} + \text{OH}^{-(aq)} \\
2\text{H}^{+(aq)} + 2e^- \rightarrow \text{H}_2(g) 
\]

The cation exchange membrane does not allow the movement of water, gas or anions across it. Hence, it only allows sodium ions (Na\(^+\)) which cross to form NAOH in the cathode (or caustic) compartment [148]. The anolyte (depleted brine) at a higher pH flows out of the anode chamber while NAOH flows out of the cathode chamber. The overall chemical equation describing this process is:

\[
2\text{NaCl}^{(aq)} + 2\text{H}_2\text{O}(l) \rightarrow \text{Cl}_2(g) + 2\text{NaOH}^{(aq)} + \text{H}_2(g) \tag{5.1} 
\]
Increasing the magnitude of electric current results in faster reaction thereby leading to more consumption of reactants and vice versa. Therefore, the magnitude of electric current is correlated with brine flow rate. A change in electric current (and brine flow) affects the pH of depleted brine flowing out of the cell. Cl₂ bubbles also occupy part of the area at the anode chamber, making the volume expand with increase in electric current and vice versa. This produces non-linear dynamics in the pH response. The anolyte exiting the anode is saturated with dissolved Cl₂. Hence, it is sent for dechlorination before being recycled or purged out as waste water.

Dechlorination process is in stages; first physical dechlorination to extract chlorine for its economic value. And subsequently, chemical dechlorination to remove all traces of chlorine from the anolyte. Before proceeding with physical dechlorination, pH of depleted brine is lowered to around 2 by the addition of HCl. This decreases the solubility of Cl₂ in depleted brine [148]. It is important to regulate the pH of anolyte at this point as it affects the final pH of the purge; any fluctuation in pH is further amplified down stream. The goal is for the purge to be of neutral pH i.e. pH of 7. Whenever pH is off specification, the waste brine needs to be treated before it is purged out.

Physical dechlorination entails spraying the acidified brine into a vacuum of over 50 kPa [148]. The extracted chlorine at this point is fed to the chlorine stream. The evaporated water from the dechlorinated brine is condensed and the condensate now at a much higher pH passes through two stages of chemical dechlorination before disposal or further treatment if pH is off specification.

5.3 Control Objectives

The section of interest in the process is presented in the process flow diagram, Figure 5.3. The solubility of Cl₂ in anolyte depends on its pH. Good pH regulation is therefore essential for physical extraction of chlorine. A low pH of around 2.0 reduces the formation of hypochlorate. The pH of electrolyte exiting the membrane cell is affected by change in magnitude of electric current passed through the cell. Brine enters the cell at a pH of about 2.0 but leaves at a higher pH due to the presence of hydroxyl ions (OH⁻). HCl is therefore used to lower the pH. Chlorine bubbles formed in the anode chamber of the cell make the system dynamics directionally dependent. Further more, any disturbance to the pH of the anolyte is further amplified downstream,
thereby affecting the pH of waste brine/water. Because of the risk of electrolytic cells running dry during load current ramps up, the rate of change of brine flow is much higher during ramp up than it is during load ramp down. This in addition to the bubbles described earlier contribute to the directionally dependent dynamics of the process.

5.3.1 Control structure

Because of correlation between change in electric current and change in brine flow, only change in electric current was used as a measured disturbance in modelling and controller development. A model with SISO structure and a measured disturbance was used. This structure was selected to capture the dynamics between HCl valve position (MV) and pH of anolyte (CV). The magnitude of electric current was used as the measured disturbance (MD). This set up is shown in Figure 5.4.

A PID with feed-forward lead-lag compensator was used to regulate the process. This PID controller was considered to be of very good performance, significant effort by an experienced control engineer was used to achieve its design. This was achieved using an improved mechanistic model of the process developed in VisSim\(^1\). The mechanistic model was improved by fitting its parameters to plant data. The use of this model in a general purpose software (VisSim) allowed for easy implementation of control

\(^1\)A visual simulation software
algorithms that are available within the DCS. Before defining the control problem, a study was carried out on the variability of pH during normal plant operations i.e. when there were no significant load current changes.

### 5.3.2 Variability Assessment

Variability assessment is often used to quantify control systems improvements during steady state operations [150]. A commonly used variability metric is the standard deviation. For a measured data sequence \( y(i), i = 1, 2, \ldots, n \), assuming the measurement noise standard deviation \( S_{\text{meas}} \) is less than 10 %, then the noise standard deviation can be neglected and the product standard deviation \( S_{\text{prod}} \) is equal to the total standard deviation, \( S_{\text{tot}} \):

\[
S_{\text{prod}} = S_{\text{tot}} - S_{\text{meas}} (\approx 0) \quad (5.2)
\]

\[
S_{\text{tot}} \approx S_{\text{prod}} = \sqrt{\frac{\sum_{i=1}^{n} (y(i) - \bar{y})^2}{n - 1}} \quad (5.3)
\]

where \( \bar{y} \) is the mean of the data sequence. During normal plant operations, when control is poor, plants are characterised by long-term drifts, cycles from special causes and inherent random variabilities [150]. Therefore, improving control can lead to reduction in variability. An ideal condition is when all variances are eliminated except for the natural variability of the process. This is known as the process capability and it is the least achievable variability with a given process. The capability standard deviation \( S_{\text{cap}} \) can be estimated using the mean square successive difference (MSSD) [150]:

\[
S_{\text{cap}} = \sqrt{\frac{\sum_{i=2}^{n} (y(i) - y(i-1))^2}{2(n-1)}} \quad (5.4)
\]
\[ S_{\text{cap}} = \sqrt{\frac{\sum_{i=2+k}^{n} (y(i) - y(i - 1 - k))^2}{2(n - 1 - k)}} \]  

(5.5)

where \( k \) is the discrete dead-time in samples. Equation (5.5) is a modification of (5.4) to account for the effect of dead-time. Minimum variability is achievable with ideal feedback such as minimum variance control. However, due to its aggressiveness, minimum variance control can result in undesirable process upsets. Hence, an estimate standard deviation for minimum variance control \( S_{\text{apc}} \) was computed using the Fellner’s formula [151]:

\[ S_{\text{apc}} = S_{\text{cap}} \sqrt{2 - \left[ \frac{S_{\text{cap}}}{S_{\text{tot}}} \right]^2} \]  

(5.6)

And the resulting percentage reduction in standard deviation is given as:

\[ S_{\text{red}}(\%) = \left( 1 - \frac{S_{\text{apc}}}{S_{\text{tot}}} \right) \times 100 \]  

(5.7)

To define the control objectives, the variability metrics introduced were used to estimate the capability of the process. To achieve this, historical plant data was collected and analysed. It was observed that different set-points for pH ranging between 1.7 and 2.1 were used in the plant. As such data for the different set-points was collected. The data was preprocessed to discard regions with significant electric current disturbance. The variability metrics computed for the different set-points are shown in Table 5.1. The values show that total variability standard deviation \( S_{\text{tot}} \) varied between 0.01 and 0.02, while the estimated plant capability \( S_{\text{cap}} \) varied between 0 and 0.01. The estimated achievable improved variability \( S_{\text{apc}} \) was also in the range of 0.01. Bearing in mind that measurement noise was neglected in obtaining these results, not much improvement is expected in terms of variability during steady state plant operations i.e. when there is no change in electric current.

### 5.3.3 Statement of control problem

As a result of the variability analysis, the control problem can be stated as: to regulate the pH of brine flowing into the anolyte tank such that the variability during steady state operation is at least maintained and to improve the rejection of electric current (and brine flow) disturbance. This will in turn ensure efficient dechlorination of brine and the final pH is within allowable specification for safe disposal. Effective
Table 5.1: Variability metrics at different set-points

<table>
<thead>
<tr>
<th>Variability metric</th>
<th>S_{rot}</th>
<th>S_{cap}</th>
<th>S_{apc}</th>
<th>S_{red}(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set-point</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.70</td>
<td>0.012</td>
<td>0.003</td>
<td>0.004</td>
<td>67</td>
</tr>
<tr>
<td>1.78</td>
<td>0.021</td>
<td>0.005</td>
<td>0.007</td>
<td>67</td>
</tr>
<tr>
<td>1.90</td>
<td>0.020</td>
<td>0.005</td>
<td>0.006</td>
<td>70</td>
</tr>
<tr>
<td>2.10</td>
<td>0.009</td>
<td>0.002</td>
<td>0.002</td>
<td>78</td>
</tr>
</tbody>
</table>

disturbance rejection will reduce the frequency of diverting depleted brine to off-specification tank for treatment thereby reducing the costs associated with such task.

5.4 MPC algorithms used

Both MATLAB and PerceptiveAPC were used to achieve the results in this chapter. All MATLAB simulations were carried out using the linear velocity state space MPC formulation discussed in Section 3.2. Although MATLAB would not be used on the actual industrial process, using it on the simulation allowed for comparison of an academic form of MPC with one that is more frequently applied in industry. For the actual plant control, the MPC tools presented in [152] were used. This formulation which has found application in industrial processes [125], uses incremental ARX models [117] for prediction. The cost function, used for this MPC design is:

\[
J = \sum_{i=1}^{N_p} e_{i+1} Q e_{i+1}^T + \sum_{i=1}^{N_c} \Delta u_i R \Delta u_i^T \tag{5.8}
\]

where \( \Delta u_i \) is the incremental input vector, \( e_i (= r - y) \) is the vector of set-point tracking errors at sample instance \( i \), \( y \) is the output vector, \( r \) the set-point vector, \( N_p \) is the prediction horizon, \( N_c \) is the control horizon, \( Q \) and \( R \) are weighting matrices, and \( \Delta \) is the difference operator. The incremental ARX model used for prediction is represented as follows:

\[
A(z) \Delta y(k) = B(z) \Delta x(k) + \xi(k) \tag{5.9}
\]

where \( A(z) \) and \( B(z) \) are polynomials in \( z^{-1} \), \( x(k) \) is a vector of manipulated and disturbance variables measured at sample time \( k \), \( \xi(k) \) is a white noise sequence. This model representation offers improved flexibility and robustness to disturbances and
noise when compared with absolute model representation [125]. This formulation of MPC also ensures off-set free control because of the embedded integral action. Variable delay spreads\(^2\) [125] are used to reduce the computational cost associated with control calculation. The MPC tool used also uses some horizons to achieve move blocking [153]. These horizons are briefly introduced:

- **Constrained Horizon**: The number of future samples over which the controller calculates control moves.
- **Compressed Horizon**: The number of samples in the constrained horizon that are compressed together for computational efficiency.
- **Compression Width**: Future samples in the compressed horizon that are grouped into blocks. The compression width defines how many samples there are in a block.

These horizons and the benefits that they offer are described in Appendix B.

### 5.5 Performance assessment using mechanistic model

Plant trials associated with controller performance evaluation could lead to expensive and undesirable upsets in plant operations. The use of improved mechanistic models have been shown to reduce the cost associated with controller trials significantly [111]. Improved mechanistic models are obtained by fitting mechanistic models with process historic data (plant inputs, outputs and disturbances). Other benefits of using these improved mechanistic models include: the cost of hardware and other new equipment can easily be estimated, better understanding of plant operations and developing confidence in new controller designs [111]. The mechanistic model if available in general modelling software can also be used for operator training. In this case, the model can also be used to investigate reasons for earlier problems with MPC implementation.

The mechanistic model developed by the plant engineers was studied and control system designed for the model. The PID controller used to control the process was designed within the software. However, to develop the MPC, MATLAB was initially used before final implementation using PerceptiveAPC. The Open Connectivity (OPC) [154] standards was used to communicate between the different software i.e. VisSim to/from MATLAB and VisSim to/from PerceptiveAPC.

\(^2\)delay spread is a term used in perceptiveAPC which means total number of samples.
Assuming a discrete time linear transfer function structure, the plant model was described as:

\[ y(t) = G(q)u(t) \]

where the matrix \( G(q) = \begin{bmatrix} g & g_d \end{bmatrix} \), \( u(t) = \begin{bmatrix} u_1(t-k) \\ u_d(t-k_d) \end{bmatrix} \), \( u_1 \) is the MV with discrete dead-time \( k \) and \( u_d \) is the disturbance variable (DV) with dead-time \( k_d \). In this section, the mechanistic model serves as the plant. Hence the study that follow in this section was carried out on the mechanistic model. This model has been used by plant operators in previous studies and was considered an accurate approximation of the process. The feed forward control used on the process was tuned using this mechanistic model.

Step responses were used to obtain estimates of dead-time and process time to steady state. The estimated dead-time and time to steady state from HCl valve to pH were 18 seconds and 67 seconds respectively, with a dead-time to time constant ratio of about 1. The corresponding values for current to pH were 77 seconds and 240 seconds. Linear ARX models were developed from identification data collected by exciting the mechanistic non-linear model. Step tests were used for response \( u_1 - y \). Step tests are not allowed on the electric load current on actual process. Therefore, to obtain realistic results, series of ramps were used for the response \( u_d - y \). Typical responses of mechanistic model to step disturbances are shown in Figures 5.5 and 5.6.

\[ \text{Figure 5.5: Model step response of HCl valve to pH} \]

\[ \text{Figure 5.6: Typical responses of mechanistic model to step disturbances} \]
In MATLAB, the augmented velocity state space format was used to design an MPC controller. A sampling interval of 3 seconds was used. Penalty on control move was used as the main tuning parameter. PerceptiveAPC on the other hand uses unbiased ARX models identified using UD-RLS [122] for its prediction. Both the penalties on move and error were used as MPC tuning parameters in PerceptiveAPC.

For the mechanistic model, the dynamics of the process also depend on certain parameters. Three of these parameters labelled as $G_{V_0}$, $G_K$ and $G_\theta$ were used in the mechanistic model to capture disturbances such as fluctuations in raw material and other unmodelled plant disturbances. These parameters were used to ensure that controllers tuned were robust enough to accommodate uncertainties in process dynamics and other variations in process parameters.

The developed controllers were applied to the simulation via an OPC connection to the VisSim model. Mean absolute error (MAE) was used as a performance measure. To test controller robustness, the parameters $G_{V_0}$, $G_K$ and $G_\theta$ were varied within $\pm 20\%$ of their nominal values. Simulations were then carried out with ramp disturbances. The ramp disturbance used is shown in Figure 5.7. The improvements in percentage MAE are summarised in Table 5.2. The corresponding responses with low and high model parameters are shown in Figure 5.8. The responses for control using MATLAB are shown in Figure 5.9. For PerceptiveAPC software, there was an improvement in MAE of up to 94% while the lowest improvement was about 65%. To further test the
performance of the controller, simulations were also carried out with step disturbances. The step responses are presented in Figures B.1 – B.3 in the Appendices.

Figure 5.7: Ramp Disturbance

Table 5.2: Controller Performance

<table>
<thead>
<tr>
<th>Process Parameter</th>
<th>Percentage improvement in MAE over PID</th>
</tr>
</thead>
<tbody>
<tr>
<td>( G_k ) ( G_{V0} ) ( G_\theta )</td>
<td>PerceptiveAPC</td>
</tr>
<tr>
<td>Ramp (%)</td>
<td>Step (%)</td>
</tr>
<tr>
<td>Low value</td>
<td>16 126 48</td>
</tr>
<tr>
<td>Nominal Value</td>
<td>20 140 60</td>
</tr>
<tr>
<td>High values</td>
<td>24 168 72</td>
</tr>
</tbody>
</table>
(a) PH and MV with nominal parameters

(b) PH and MV for low parameter values

(c) PH and MV with high parameter values

Figure 5.8: Plots of PH and MV due to Ramp disturbance using PerceptiveAPC
Figure 5.9: Plots of PH and MV for Ramp disturbance using MATLAB
These results show that if a reasonable model of the plant is available for MPC design, significant improvements in load current disturbance rejection are achievable. The difference in controller performance between MATLAB and PerceptiveAPC is a subject of future research. However, it may be related to the modelling approach and MPC formulation and tuning. Based on study carried out, it is believed that earlier problems with MPC implementation could be due to a number of factors such as the modelling capabilities of the software used and also quality and quantity of test data. However, not much can be improved in terms of data quality as only ramps are possible on electric current. Another important factor may be that enough time was not spent to fully understand the process. The next section discusses the development of controller for the plant and challenges encountered in the process of developing the MPC.

5.6 Plant Control

The study of plant variability in Section 5.3 showed that little or no improvements are achievable in terms of pH variation about set-point during normal operation (when there are no significant electric current disturbances). Therefore, while efforts were made to improve this variability, emphasis was on improvement in rejection of electric current disturbance.

The DCS in use in the chlorine plant is Delta V (from Emerson Process Management). Therefore, an OPC link was set up between the DCS and PerceptiveAPC software. Once the communication link with all the safety features were in place, plant tests were carried out to obtain data for model development. Multiple steps and ramps were used to generate open loop data for HCl valve position to pH and electric current to pH respectively. PerceptiveAPC was used to identify incremental ARX models using UD-RLS. Plots of sample data collected during plant tests are shown in Figure 5.10. Plots of more data used for identification are shown in Figures B.4 and B.5 in Appendix B.
Figure 5.10: Plant test data at pH around 1.8
5.6.1 Controller tuning and testing

Based on estimates of settling time obtained in the simulation of Section 5.5, a sampling time of 12 seconds was used for the initial controller design. The step responses of incremental models developed with this sampling time are shown in Figures 5.11 and 5.12.

![Figure 5.11: 12 seconds model step response from HCl to pH](image1)

![Figure 5.12: 12 seconds model step response from Load current to pH](image2)
The estimated dead-time and process settling time for HCl valve to pH were 24 seconds and 60 seconds respectively. The corresponding values for electric current to pH were 48 seconds and 276 seconds. For these models, to capture the dynamics of the slower response a constrained horizon of 26 was used. A compressed horizon of 15 and compression width of 5 were used to reduce computational load. Set-point and move rate weightings of 1.5 and 3.1 were used respectively.

Some improvements were observed in the rejection of electric current disturbance. But, initial analysis showed a number of issues. The MPC controller was affected by occasional spikes in the electric current measurements. This measurement noise was ranging between 100 A to 300 A, with 100 A being more prevalent. Therefore, a filter was used to reduce the effect of noise on the electric current measurement. A plot showing sample load current measurement and its filtered version is given in Figure 5.13.

![Figure 5.13: Plot of measured and filtered load current](image)

Another problem observed was that of stiction in HCl valve. This was more pronounced at lower load currents (below 80 kA), when demand from plant was relatively low. The effect was more noticeable in MPC. This effect is shown on Figures 5.14 and 5.15 for PID and MPC respectively. Spikes with overshoots of around 2% were
observed in valve position readback for both PID and MPC. The more pronounced effect in MPC is expected since it was only sampling after 12 seconds while PID was sampling every second. The faster sampling for PID controller ensured that the valve was moving more frequently while taking more cautious steps. For MPC, the valve moved at intervals of 12 seconds and was therefore more prone to deterioration of performance due to stiction.

Two steps were taken to minimise the effect of stiction. First the readback of valve position was used for MPC control computations and decision. Secondly, a faster MPC was designed. More frequent sampling could reduce the effect of stiction and improve controller performance. However, this will be at additional computational cost. The effect of stiction on MPC with readback is shown in Figure 5.16

A sampling time of 2 seconds was used for the faster model. More accurate estimates of the dead-time and time to steady state were obtained. The estimates of dead-time and time to steady state for HCl valve to pH are 32 seconds and 66 seconds respectively. The corresponding values for the electric current to pH were 54 seconds and 276 seconds. The step responses for these models are shown in Figures 5.17 and 5.18.
Figure 5.15: Plot showing effect of stiction on MPC

Figure 5.16: Plot showing effect of stiction on MPC with readback
An MPC controller was tuned using set-point and move rate weights of 1.2 and 17
respectively. A constrained horizon of 120 (prediction horizon), compressed horizon of 100 and compression width of 10 were used. For these settings, the control horizon is equal to the prediction horizon. However, the use of compressed and compression horizons allowed for control moves to be kept constant over 10 sample steps after the initial 20 samples.

### 5.6.2 Performance evaluation

The controller was tested for electric current ramp disturbance. Sample responses for an initially tuned MPC controller, PID (with feed forward compensator) and finally tuned MPC are shown in Figure 5.19.

![Figure 5.19: Sample of response to load disturbance with different controllers](image)

For electric current disturbance rejection, MPC was able to maintain the pH within ±0.1 of set-point. The values of MAE for the respective controllers are given in Table 5.3. Using MAE as performance measure, an improvement of about 57% was achieved. The estimated improvements using mechanistic model ranged between 67 to 97%. Hence, the estimate using mechanistic model was reasonable given uncertainties and other unmeasured effects affecting dynamics of the actual plant. Due to plant
operations and other factors, it was not possible to test controller performance at low loads. This is because the plant cannot be arbitrarily dropped to these low loads.

Table 5.3: MAE for controllers

<table>
<thead>
<tr>
<th>Controller</th>
<th>Ramp size (kA)</th>
<th>MAE</th>
<th>% improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID</td>
<td>8.5</td>
<td>0.07</td>
<td>-</td>
</tr>
<tr>
<td>MPC (Initial tuning)</td>
<td>8.5</td>
<td>0.04</td>
<td>42.85</td>
</tr>
<tr>
<td>MPC (Final tuning)</td>
<td>12.2</td>
<td>0.03</td>
<td>57.14</td>
</tr>
</tbody>
</table>

As done for the PID, data was collected and portions of data with variability due to change in load current removed. The total variability with MPC control, $S_{mpc}$ for a set-point of 2.1 was computed as 0.01. This is in the range of values obtained for PID controller, thereby satisfying the first objective of the controller. However, observing the plots in Figure 5.19 show that for the PID controller, there was a periodic long term variability with a period of about 50 minutes. This periodic variability was eliminated by the MPC controller. However, as plant was maintained at the set-point of 2.1 during the period of the trial, it was not possible to compute variability for other set-points.

5.7 Conclusions

In this chapter, pH control for dechlorination of depleted brine in a Chloro-Alkali plant was discussed. The industrial process considered has dynamics that can benefit from the application of small scale MPC. The industrial case study was carried out to further ascertain the simulation results (presented in Chapter 4) which show that benefits are attainable by replacing existing PID controllers with small scale MPC in suitable processes. Variability assessment was carried out to quantify achievable benefits. An improved mechanistic model was also used to estimate benefits achievable by replacing existing control with MPC.

Improved disturbance rejection was achieved using MPC. The improvements were representative of estimates obtained using the mechanistic model. However, initial benefits were limited due to practical problems such as noise, modelled dynamics and valve stiction. But further improvements were obtained by taking practical steps to address the encountered problems. The improved disturbance rejection will ensure less frequent diversion of brine to the off-specification tank, thereby saving cost associated with brine treatment.
To achieve the improved control presented in this chapter, plant tests were still used. This led to some disturbances in plant operations. Harnessing the information embedded in stored historic data could reduce the amount of time spent on plant tests and disturbance of plant schedules. This is explored in the next chapter.
Chapter 6

MPC from Routine Data

6.1 Introduction

Despite the success of Model Predictive Control (MPC) in industry, the presence of small scale MPC is still relatively limited. A number of factors contribute to this limited presence. These include the engineering efforts required, cost of implementation and the lack of availability of MPC in off-the-shelf control equipment [19, 83]. The cost associated with plant test and modelling account for over 70% of the incurred cost associated with model based control [113]. In this chapter, a method of designing MPC controllers based on the knowledge of existing Proportional Integral Derivative (PID) controller parameters, structure and routine plant data is proposed. This approach is motivated by the concept of closed-loop identification. Once developed, the procedure can be programmed into the distributed control system (DCS) and could allow PID controllers to be replaced by MPC with little effort. The method will have the advantage of not interfering with plant operations. It therefore reduces considerably both the cost associated with model development and the engineering efforts associated with MPC design. The MPC can be initially designed to have comparable performance to the existing controller. Although the performance of this MPC controller may not be optimal, this is a key step in replacing the PID controller with an appropriate MPC. And once a working controller is available on-line, further tuning can be used to achieve improved performance.

MPC is known to effectively deal with constraints during transient, but simple tools to analyse frequency domain properties such as sensitivity are lacking [86]. Hence, designing an MPC that is equivalent to the existing linear feedback control can allow the predictive controller inherit some characteristics of the linear feedback control
CHAPTER 6. MPC FROM ROUTINE DATA

122

schemes [22]. Many applications in which MPC is a practical option already have some form of feedback in place [87]. And replacing an existing controller with MPC is justifiable if it is a desirable upgrade for improved performance [22]. Hence, the minimum acceptable requirement is that the MPC be of at-least equivalent performance to the replaced linear feedback controller. A good starting point will be to start with an equivalent MPC, which by extension will have same unconstrained closed loop behaviour as the replaced feedback [88]. This initially tuned equivalent MPC can then serve as a baseline for further tuning thereby reducing the efforts associated with MPC design and tuning. Most existing works in this area are based on the assumption of the availability of plant models [22, 86–88], which is not usually the case. Moreover, most existing industrial controllers are of the Proportional Integral (PI) type. Hence, this chapter focuses on designing the replacement MPC based on knowledge of existing control and routine plant operation data; the restricting assumption of the existence of process model is removed in this formulation.

In the next section a discussion on routine plant data and the requirements for identifiability from such data is presented. Section 6.3 discusses MPC and its development from routine plant data. Simulations and results are presented in Section 6.4 before concluding the chapter in Section 6.5.

6.2 Routine plant data and identifiability

Consider a closed loop system with plant $G$, controlled by a controller $G_c$, with output disturbance, $H$ as depicted in Figure 6.1. Let the discrete time transfer functions, $G(z)$, $H$ and the control error $e$.
\( G_c(z) \) and \( H(z) \) be defined as follows:

\[
G(z) = \frac{z^{-1}B(z^{-1})}{A(z^{-1})F(z^{-1})} \tag{6.1}
\]

\[
G_c(z) = \frac{X(z^{-1})}{Y(z^{-1})} \tag{6.2}
\]

\[
H(z) = \frac{C(z^{-1})}{D(z^{-1})A(z^{-1})} \tag{6.3}
\]

With polynomials \( A, X \) and \( B \) of orders \( n_A, n_X \) and \( n_B \) respectively defined as:

\[
A(z^{-1}) = 1 + \sum_{i=1}^{n_A} a_i z^{-i},
\]

\[
X(z^{-1}) = x_0 + \sum_{i=1}^{n_X} x_i z^{-i},
\]

\[
B(z^{-1}) = \sum_{i=1}^{n_B} b_i z^{-i}.
\]

\( z^{-1} \) is the backward shift operator and the polynomials \( C, D, F \) and \( Y \) are defined similarly to \( A \). The closed-loop expressions for the output, \( y(t) \) and input, \( u(t) \) are:

\[
y(t) = \frac{G_c G}{1 + G_c G} r(t) + \frac{H}{1 + G_c G} e(t), \tag{6.4}
\]

\[
u(t) = \frac{G_c}{1 + G_c G} r(t) - \frac{G_c H}{1 + G_c H} e(t). \tag{6.5}
\]

For the purpose of this work, routine operation data is considered as discrete plant data obtained from a process in closed-loop without any excitation except for the natural plant disturbances. Assuming that there are no set-point changes in the routine data i.e. \( r(t) = 0 \), then (6.4) and (6.5) become (6.6) and (6.7) respectively:

\[
y(t) = \frac{H}{1 + G_c G} e(t), \tag{6.6}
\]

\[
u(t) = -\frac{G_c H}{1 + G_c H} e(t). \tag{6.7}
\]

For a model to be identifiable from routine plant data, certain requirements need to be met. These requirements are on a number of parameters including; model structure, plant model orders, disturbance model orders, and relative order of controller. Order
based conditions for model identifiability using prediction error framework are given in (3.67) of Theorem 3.3. These order based conditions are equivalent to testing the richness of data [136]. However, since dead-time can readily be estimated using well known methods [138], a more practical approach is to express the theorem in terms of discrete dead-time. The order based conditions were therefore resolved to obtain Theorem 6.1 which gives conditions on discrete dead-time [135].

**Theorem 6.1.** ([135]) A system can be identified using the general prediction error (PE) framework without any external excitation if the following constraint on the discrete time delay is satisfied:

\[
\begin{aligned}
  n_k &\geq n_d + n_y - n_x + 2n_F + n_A \\
  + \min(n_C, n_A, n_B - n_F + n_X - n_Y) &\geq n_y - n_X + n_A + n_F - n_B \\
  n_y - n_X &\geq n_D + 2n_B
\end{aligned}
\]

(6.8)

The implication of (6.8) is thus; if the discrete time delay is large enough, then given a fixed model structure, it is possible to identify the system. The exact requirement for how large \( n_k \) should be is given by the first row of (6.8). Other factors that determine if a model is identifiable are the relative degree of controller \((n_X - n_Y)\) and that of the process model \((n_B - n_F)\).

Equation (6.8) can be used to determine conditions on identifiability for commonly used controllers and model structures. In this work, the following controllers were considered:

1. Proportional (proportional (P)) controllers: \( n_X = 0, n_Y = 0, \)
2. Proportional and Integral (PI) controllers: \( n_X = 1, n_Y = 1, \)
3. Proportional, integral and derivative (PID) controllers: \( n_X = 2, n_Y = 1. \)

While the model structures considered were:

1. Autoregressive with exogenous input (ARX): \( n_F = 0, n_D = 0, \)
2. Output Error (OE): \( n_C = 0, n_D = 0, n_A = 0, \)
3. Box-Jenkins (BJ): \( n_A = 0, \)
4. Autoregressive Moving Average with exogenous input (ARMAX): \( n_D = 0, n_F = 0. \)
Since both P and PI have the same relative degree \((n_X - n_Y)\), the identifiability conditions for these controllers are the same. The identifiability conditions for P/PI and PID controllers are presented in Tables 6.1 and 6.2 respectively. The general constraint (GC) is obtained from the equation on the right hand of the first row of (6.8), while the second constraint (SC) is obtained from the expression on the left side of the first row. In all cases, the SC is larger than the corresponding GC. For first order models (i.e. when all the model polynomials are of order 1), with a PI controller, the GC requirement for ARX and BJ is that the discrete dead-time should at least be 1 and 3 respectively while both OE and ARMAX require the discrete dead-time to be at least 2. With a PID controller, the requirement is that the discrete dead-time should be at least 0 and 2 for ARX and BJ respectively while both OE and ARMAX require \(n_k \geq 1\).

The highest identifiability requirement for controllers discussed here is \(n_k \geq 3\). In this work, this condition is first checked and if satisfied then a model is identifiable irrespective of model structure or controller used. Whenever this requirement is not met, then requirements for the specific controller in use are checked to ensure that the discrete dead-time meets the minimum requirement. If the discrete dead-time does not meet the requirements, then a more advanced controller than the PID is required for identifiability. And this is beyond the scope of this work.

The constraints on discrete dead-time are easily extended to requirements on sampling time. The relationship between discrete time dead-time and continuous time dead-time is:

\[
n_k = \frac{\theta}{T_s}
\]  

(6.9)

where \(\theta\) is the continuous dead-time and \(T_s\) the sampling time. The implication of this is that any process with a dead-time can be identifiable if the sampling time is fast enough. Hence, in addition to the requirement for sampling at a rate that captures the dynamics of the process adequately, another requirement is that sampling should be fast enough to ensure that the minimum requirement on discrete dead-time given by Theorem 6.1 is achieved. This condition is easily achievable for dead-time dominant systems. Moreover, it has been demonstrated in Chapter 4 that dead-time dominant systems are likely to benefit from the replacement of PID loops with MPC even in the single-input single-output (SISO) case, which is also in line with the suggestions in [12].
Table 6.1: Constraints on $n_k$ for P/PI controller and different process models

<table>
<thead>
<tr>
<th></th>
<th>PE</th>
<th>ARX</th>
<th>OE</th>
<th>ARMAX</th>
<th>BJ</th>
</tr>
</thead>
<tbody>
<tr>
<td>GC</td>
<td>$n_A + n_F - n_B$</td>
<td>$n_A - n_B$</td>
<td>$n_F - n_B$</td>
<td>$n_A - n_B$</td>
<td>$n_F - n_B$</td>
</tr>
<tr>
<td>SC</td>
<td>$n_D + 2n_F + n_A$</td>
<td>$n_A$</td>
<td>$2n_F + \min(0, n_B - n_F)$</td>
<td>$n_A + \min(n_C, n_A, n_B)$</td>
<td>$n_D + 2n_F$ + $\min(0, n_B - n_F)$</td>
</tr>
</tbody>
</table>

| First order | GC   | 1   | 0 | 0 | 0 | 0 |
|             | SC   | 4   | 1 | 2 | 2 | 3 |

Table 6.2: Constraints on $n_k$ for PID controller and different process models

<table>
<thead>
<tr>
<th></th>
<th>PE</th>
<th>ARX</th>
<th>OE</th>
<th>ARMAX</th>
<th>BJ</th>
</tr>
</thead>
<tbody>
<tr>
<td>GC</td>
<td>$n_A + n_F - n_B - 1$</td>
<td>$n_A - n_B - 1$</td>
<td>$n_F - n_B - 1$</td>
<td>$n_A - n_B - 1$</td>
<td>$n_F - n_B - 1$</td>
</tr>
<tr>
<td>SC</td>
<td>$n_D + 2n_F + n_A - 1 + \min(n_C, n_A, n_B - n_F)$</td>
<td>$n_A - 1$</td>
<td>$2n_F - 1 + \min(0, n_B - n_F)$ + $\min(n_C, n_A, n_B + 1)$</td>
<td>$n_A - 1 + \min(n_C, n_A, n_B + 1)$</td>
<td>$n_D + 2n_F - 1 + \min(0, n_B - n_F)$ + $\min(n_C, n_A, n_B + 1)$</td>
</tr>
</tbody>
</table>

| First order | GC   | 0   | -1 | -1 | -1 | -1 |
|             | SC   | 4   | 0  | 1  | 1  | 2  |
In continuous time, the plant is generally modelled using first order plus dead-time (FOPDT) or second order plus dead-time (SOPDT) models represented by (6.10) and (6.11) respectively. Assuming first order discrete plant model (6.12) obtained using exact first order discretisation [155], then the FOPDT parameters of (6.10) can be recovered using (6.13) and (6.14).

\[
G(s) = \frac{Ke^{-\theta s}}{\tau s + 1} \quad (6.10)
\]

\[
G(s) = \frac{K(\tau_2 s + 1)e^{-\theta s}}{\tau^2 s^2 + 2\zeta\tau s + 1} \quad (6.11)
\]

\[
\hat{G}(z^{-1}) = \frac{b_1 z^{-n_k} - 1}{1 + a_1 z^{-1}} \quad (6.12)
\]

\[
K = \frac{b_1}{1 + a_1}, \quad (6.13)
\]

\[
\tau = \frac{-T_s}{\ln(-a_1)}. \quad (6.14)
\]

### 6.3 MPC based on routine plant data

Using the state space velocity format MPC presented in Chapter 3. The unconstrained optimal sequence of control moves and the move at time \( k \) are given as:

\[
\Delta U(k) = \left( \Phi^T \bar{Q} \Phi + R \right)^{-1} \Phi^T \bar{Q} (S - Fx_0) \quad (6.15)
\]

\[
\Delta u(k) = \begin{bmatrix} I_m & 0 & \ldots & 0 \end{bmatrix} \Delta U(k) \quad (6.16)
\]

\[
u(k) = u(k-1) + \Delta u(k) \quad (6.17)
\]

where all parameters are as defined in section 3.2.4. A suitable prediction horizon, \( N_p \), is the settling time of the process as defined in (6.18). The value of \( P \) obtained from (6.18) and \( N_c = 3 \) were used in this chapter. These values are fixed to reduce the number of MPC tuning parameters. With these choices of \( N_p \) and \( N_c \), MPC tuning is achieved using the weighting, \( r_w \).

\[
N_p = \frac{(\theta + 5\tau)}{T_s} \quad (6.18)
\]

Assuming a FOPDT model and a known dead-time, the MPC defined by (6.15) only needs the parameters of the model; \( K, \tau \) and \( \theta \), weight, \( r_w \), and measured output, \( y_p(t) \), to give the change in control input \( \Delta u(k) \) at any time instance \( k \). Therefore, defining
these parameters is equivalent to designing an MPC and the parameters, $K$, $\tau$ and $r_w$ are referred to as the \textit{MPC-parameters}, $\theta_r$. In the MPC formulation used, off-set free control can be achieved without explicit disturbance modelling. Hence, for simplicity, disturbance modelling is not included in the formulation of routine data MPC (RMPC).

Consider the block diagram of Figure 6.1 with the controller $G_c$ as a PID. Assume also that the only excitation is the noise signal $e(t)$. Let the data be sampled to meet the constraints in (6.8) and to adequately capture the dynamics of the process. The data so obtained therefore meets the requirement for identifiability from routine operation data. Assume also we have a knowledge of the structure of the PID controller, $G_c$ and its parameters, $K_p$, $\tau_I$ and $\tau_D$. The input and output of the controller $e_r (= -y)$ and $u$ respectively are obtained directly from the routine data. Now let the PID controller in Figure 6.1 be replaced with an MPC such that for any given $e_r$, the MPC gives the same $u$ as the PID controller. This implies that the PID controller is to be replaced with an equivalent MPC controller.

The problem described can be stated as, find the \textit{MPC-parameters} that will give the same performance as a PID controller of known structure and parameters given the controller input, $e_r$ and output, $u$. The practical approach here is to find the parameters that will minimise an error function between the PID controller output and the MPC output given a set of routine data sequence; the sum of squared error (SSE) was used in this case.

The aim therefore is to design an MPC controller based on the knowledge of the existing PID controller and available routine data. Consider the unconstrained MPC cost function of (6.15), using the prediction horizon defined in (6.18) and a control horizon of 3. Assuming a FOPDT model, the model parameters and tuning weight that will minimise the error between the output of the PID and the MPC algorithm of (6.15) can be obtained. The solution of the optimisation will be the MPC-parameters, $\theta_r = \begin{bmatrix} K & \tau & r_w \end{bmatrix}^T$.

Given the routine operation data sequence, $\{z(t)\}_{t=1}^{N}$:

$$z(t) = \begin{bmatrix} y_p(t) & u_p(t) \end{bmatrix}^T,$$

(6.19)

where $y_p(t)$ is the measured plant output and $u_p(t)$ is the controller output. Let $u_m(t)$ be the control move obtained at time $t$ by an MPC algorithm with MPC-parameters $K$, $\tau$ and $r_w$ obtained using (6.16). Let the error between the PID controller output and the
MPC output, $e_u$ be defined as:

$$e_u(t) = u_p(t) - u_m(t) \quad t = 1, 2, \ldots n$$  \hspace{1cm} (6.20)

where $n$ is the number of data samples collected. Since the plant is described using a FOPDT model, the solution to the optimisation problem of (6.21) gives the vector of MPC-parameters, $\theta_r$.

$$\theta_r^* = \arg\min_{\theta_r} J_r(\theta_r)$$

$$J_r(\theta_r) = \sum_{t=1}^{n} e_u^2(t)$$  \hspace{1cm} (6.21)

This optimisation is a non-linear problem that can be solved using any of the available routines for solving non-linear optimisation problems. Gradient based methods can encounter problems with local minima when solving discontinuous non-smooth cost functions [156]. But, search based methods such as GA and Simulated Annealing (SA) are suitable for solving this class of problem. SA is known to be good for solving non-linear global optimisation problems even when they are discontinuous and/or not well defined [156]. Since the problem is solved off-line, it will be possible to analyse results and carry out any necessary adjustments before proceeding with controller implementation. In this work, SA is used to solve the problem defined by (6.21).

SA is a meta-heuristic search algorithm inspired by the process of annealing in solids. It is often used to solve unconstrained and bound-constrained optimisation problems. When used to solve global optimisation problems, solutions stochastically converge to the global optimum. Annealing involves cooling of molten solids in a heated bath. When crystalline solids are heated beyond their melting points and then cooled very slowly until the minimum lattice energy is reached, and crystals are formed. The structural properties of these crystals depend on the final lattice energy and the rate of cooling, the most regular structures are achieved at the minimum energy level. Cooling too fast (quenching) results in crystals with irregular structure. Larger regular structures are therefore achieved by cooling the material gradually [157].

SA does not require the objective function to be differentiable or continuous. To enhance convergence to global optimum, SA uses hill-climbing moves to move out of local optima. This hill climbing feature is achieved by accepting solutions that raise the objective [158]. In each iteration, SA algorithm randomly generates a solution.
Solutions that lower the objective are accepted automatically. Solutions with higher objective are accepted with a certain probability for example the Boltzmann probability distribution [157]. The distance of the new solution from the previous is based on a parameter called the temperature such that at higher temperatures larger distances from the previous points are allowed but as the temperature becomes lower, the distance to a new point decreases. These characteristics of SA enable the hill-climbing feature that ensures gradual convergence to the global minimum as the temperature decreases.

A problem with SA is the chance of premature convergence to suboptimal states. There is no known method for completely preventing premature convergence [156]. But, the chances can be drastically reduced by striking the right balance between the exploration and exploitation feature of the algorithm. In SA, the temperature, $T$ determines the extent of exploration from the current solution. While both the $T$ and $\Delta f$ determine the probability of accepting a solution which does not improve the objective. Temperature decrease is achieved as the algorithm proceeds using an annealing schedule. The rate of decrease in temperature gives a compromise between convergence to a global minimum and speed of convergence. A procedure known as reannealing is used to periodically increase the temperature after the acceptance of a few solutions. This feature aids the algorithm to escape from local minima thereby also reducing the chance of premature convergence [156, 157].

A number of stopping conditions are used in SA. Some of which include: maximum number of iterations, maximum number of function evaluations, time limit, fitness limit, stall time limit and function tolerance. For more discussion on these, please refer to [156, 157]. A list of some important SA parameters and the values used in this work are presented in Table 6.3. To compute the initial conditions for the SA, it was assumed that the PI parameters were obtained using Internal Model Control (IMC)-tuning [31]. Therefore, the following expressions were used to obtain the initial values for MPC-parameters, $\theta_{r_0} = \left[K_0 \quad \tau_0 \quad r_{w_0}\right]^T$:

\[
K_0 = \frac{\tau_l}{K_p (\tau_c + \theta)}, \\
\tau_0 = \tau_l, \text{ and} \\
r_{w_0} = 1 
\]  
(6.22)

Where $\tau_c$ is the closed-loop time constant.

To obtain the upper and lower constraints for the MPC-parameters, different tuning
methods were studied to determine the relationship between the PI parameters and closed loop dynamics of the process. Although, no key relationship was found but it was noticed that these parameters were always within a certain range of the values obtained using IMC-tuning. The upper and lower bounds on the search space specified in (6.23) – (6.25) generated acceptable results\(^1\). The SA algorithm is presented in Algorithm 6.1.

\[
\begin{align*}
K_{\text{min}} &= \frac{K_0}{1.2} & K_{\text{max}} &= 2K_0 \\
\tau_{\text{min}} &= \frac{\tau_0}{1.2} & \tau_{\text{max}} &= 2\tau_0 \\
r_{w_{\text{min}}} &= 0.01 & r_{w_{\text{max}}} &= 1000
\end{align*}
\]

**RMPC algorithm**

Based on the formulation of RMPC, the step-by-step procedure for the design of RMPC is given in Algorithm 6.2. Given routine data set, \(z(t) = [u(t) \ y(t)]^T\) with no change in set-point, i.e. \(r(t) = 0\). An algorithm for identifying portions of data in which \(r(t) \neq 0\) is given in [133]. Here \(T_{\text{max}}\) is the maximum sampling time that ensures identifiability condition (6.8) is met.

In the post processing step, the parameters obtained from \(N\) Monte-Carlo runs were checked for normality. Lilliefors test was used to check the null hypothesis that the data is normally distributed [159]. If data has a normal distribution, parametric statistics can be used to characterize the distribution of the MPC-parameters.

\(^1\)Simulations without these bounds also converged to the vicinity of the true value but at a high computational cost
Algorithm 6.1 Simulated Annealing

**Input:** Current (Initial) Solution $x_0 = \theta_{r0}$, Initial Temperature $T_0$

**Output:** Optimisation parameter

Compute current Cost, $f$

while STOP criteria not met do

Compute New state, $x_{\text{new}}$

Compute New Cost, $f_{\text{new}}$

$\Delta f = f_{\text{new}} - f$

if $\Delta f \leq 0$ then

$x = x_{\text{new}}$

else

accept $x_{\text{new}}$ with probability $Pr = \frac{1}{1+e^{\frac{\Delta f}{T}}}$

end if

$f = f_{\text{new}}$

Decrease temperature, $T$

end while

Algorithm 6.2 RMPC

**Input:** Historic data $\{z(t)\}_{t=1}^n$

**Output:** MPC-parameters, $\theta_r \sim \mathcal{N}(\bar{\theta}_r, \sigma^2_{\theta_r})$

if $T_s \leq T_{\text{max}}$ or $\theta \geq \theta_{\text{min}}$ then

for $i = 1$ to $N$ do

find $\theta_r(i)$ using simulated annealing (Algorithm 6.1)

end for

Post processing

else {Model not identifiable from data}

exit

end if

6.4 Simulation and results

To test the proposed method of designing RMPC, the process described by the block diagram in Figure 6.1 was used with the plant and disturbance models given in (6.26) and (6.27) respectively. The true values of the MPC-parameters, $\tau$ and $K$ are given below while $r_w$ has no true value:

\[
G(s) = \frac{1.54e^{-20s}}{200s + 1}, \quad \text{(6.26)}
\]

\[
H(s) = \frac{1}{20s + 1}. \quad \text{(6.27)}
\]
A PI controller was used in this simulation as it is more prevalent than PID. Moreover, the requirements for identifiability are more stringent for the PI controller. The controller was implemented in discrete form with a proportional gain of 2 and an integral time of 200 seconds. A unit variance white noise was used as the output disturbance, \( e(t) \). Using a sampling time of 2 seconds, the simulation was run for 10,000 samples. Using the concept of Monte-Carlo simulations, 200 runs of the optimization algorithm were carried out and the MPC-parameters were obtained using the SA optimization described earlier. The plot of MPC-parameters obtained for the runs are presented in the Figures 6.2 – 6.4. Observing these figures, estimates were found to cluster close to the true values of the parameters. However, some estimates were relatively far away from the clusters and true values.

![Figure 6.2: Estimated, true value and mean of gain, \( K \)](image)

Both estimates of \( K \) and \( \tau \) did not pass the Lilliefors test. Hence parametric statistics that apply to the normal distribution are not applicable. This is also seen on histograms of the parameters overlayed on a histogram of normal distribution fitted to the data shown in Figures 6.5 and 6.6.
Figure 6.3: Estimated, true and mean value of time constant, $\tau$

Figure 6.4: Estimated and mean values of move weight, $r_w$
Figure 6.5: Histogram of gain estimates

Figure 6.6: Histogram of time constant estimates
Bearing in mind that the distributions were not normal, the means were close to the true parameter values (refer to Figures 6.2 and 6.3). The move weight, $r_w$, was observed to be relatively lower for runs with parameters closer to the true value. This is expected as the controller needs to be more conservative with larger errors in MPC-parameters. To further understand the estimates, values of the fitness function, $J_r$, were explored. $J_r$ captures the error in both the estimated parameters $K$ and $\tau$. Plot of $J_r$ is given in Figure 6.7. As expected, parameters that were closer to the true value were observed to have lower values of $J_r$. Bearing in mind that estimates were obtained using SA which is prone to premature convergence.

Figure 6.7: Plot of fitness function value, $J_r$

To compare control performance, the MPC controller was implemented on the process model defined and the results compared with that of the PI controller. The step responses were found to be similar as shown in Figure 6.8. Since other tuning parameters were fixed, RMPC can be tuned using the weighting on control move, $r_w$. Hence to obtain a faster response, $r_w$ was changed from 0.7 to 0.3. Plot of improved MPC response with that of PID is shown in Figure 6.9. The result shows that only some minor changes on-line are needed to improve performance. This significantly reduces the efforts associated with MPC tuning.
Figure 6.8: Plant step response with PI and RMPC

Figure 6.9: Plant step response with PI and improved RMPC
In a second study, a process with plant and disturbance models given in (6.28) and (6.29) respectively was used.

\[ G(s) = \frac{1.2e^{-5s}}{70s + 1}, \]  
\[ H(s) = \frac{1}{20s + 1}. \]

For this simulation, the controller was also implemented in discrete form with a proportional gain of 4.5 and an integral time of 70 seconds. A unit variance white noise was used as the output disturbance, \( e(t) \). Using a sampling time of 1 second, the simulation was run for 10,000 samples. 200 runs of the optimization algorithm were also carried out and the MPC-parameters were obtained using SA optimization. The plots of MPC-parameters obtained for the runs are presented in the Figures 6.10 - 6.12. Plot of values of fitness function is shown in Figure 6.13.

![Figure 6.10: Estimated, true value and mean of gain, K](image)

The mean of the estimates of MPC-parameters obtained in the second example were also close to the true values of the parameters. Similarly, these parameters did not pass the Lilliefors test for normality. An MPC was implemented on the process model...
Figure 6.11: Estimated, true value and mean of time constant, $\tau$

Figure 6.12: Estimated and mean values of move weight, $r_w$
CHAPTER 6. MPC FROM ROUTINE DATA

Figure 6.13: Plot of fitness function value, $J_r$

using the estimated parameters, and the resulting response compared with that of the PI controller. Plots of the step responses are shown in Figure 6.14. The performance of the MPC was similar to that of the PI, and MPC performance can also be easily improved using the tuning weight $r_w$.

In both examples, the estimates of tuning parameters were biased. This bias which is more pronounced in the estimates of $K$ could be due to the effect of feedback and the fact that disturbance modelling was not considered in RMPC formulation. However, the estimates obtained gave reasonable responses. In all cases, the mismatch in parameter estimates was less than 10%. And results in Chapter 4 have shown that the MPC formulation used in this work is robust to mismatch of up to 20% in parameters. The results also show that using only the tuning weight $r_w$, improved performance was achieved. It is expected that additional improvement in performance is achievable by using other tuning parameters such as set-point weight, move weight and set-point trajectory. However, this will result in additional tuning difficulty and is outside the scope of this work since other tuning parameters were fixed in RMPC formulation.
6.5 Conclusions

In this chapter, a method of designing MPC from routine plant data was developed. This was achieved using data without any form of excitation other than the natural plant disturbance, modelled using filtered white noise. This disturbance was simulated using a unit variance white noise filtered through a disturbance model. Using a PI controller of known structure, SA was used to minimise an error function to obtain the MPC-parameters, $K$, $\tau$ and $r_w$. The aim was to design an MPC with equivalent performance with the PI as suggested in previous research. Using the concept of Monte-Carlo simulations, estimates of the MPC-parameters that achieved this aim were obtained. Bearing in mind that the parameters obtained did not follow a normal distribution, their mean values were close to the true values. In general MPC-parameter estimates were biased. But the bias was not large enough to affect controller performance significantly. The observed bias may be due to the presence of feedback and unmodelled disturbances. Hence, an area of future work may be to consider the effect of disturbance modelling. The results obtained are promising and represent a key step in designing MPC from plant routine data – RMPC. The approach has the advantage of not interfering with plant operations while minimising the efforts associated with tuning. Improved performance was achieved using only a single tuning weight.
Chapter 7

Conclusions and Recommendation

Over the years, Proportional Integral Derivative (PID) control has been the dominant process control strategy. Its prevalence has been due to several factors including historical reasons, ease of implementation and availability in off-the-shelf hardware. However, tuning and the lack of an agreed standard structure have remained challenging factors. Despite these challenges, PID is considered as the default controller for regulating most processes. Hence, it is routinely used to regulate processes that are more suitable for control by more advanced controllers. Consequently, a significant number of industrial loops are poorly tuned and are therefore operating far from achievable optimal performance. The increasing competition in the industry, rising cost of energy and more strict regulations on environmental pollution have made the need for improved operation and control of loops imperative. This need for improved control is likely to rise with time.

On the other hand, Model Predictive Control (MPC) is one of the optimal and advanced control schemes that has had the most notable impact in the process industries. And it is also finding acceptability in other industries. This success can be attributed to a number of factors which include its ability to handle optimisation in the loop and other practical issues such as constraint handling, transparency of tuning and ability to effectively handle multi-variable systems. However, its presence in the industry is still not widespread considering benefits associated with its implementation. The relatively limited presence is due to a number of factors notably; the cost associated with implementation, requirement for expertise and the need for stand alone software.

In this thesis, a study was carried out to understand the characteristics of processes that will benefit from the replacement of PID controllers with MPC, to quantify some of the
benefits that could be achieved as a result of such replacements. Some methods and strategies for mitigating some the problems leading to the limited presence of MPC were explored, and a novel method of designing MPC from routine plant data without external excitation (except natural plant disturbances) was developed. A summary of the findings in this thesis and recommendations for future work are presented in Sections 7.1 and 7.2 respectively.


c5

7.1 Summary and Conclusions

In Chapter 4, a systematic study was carried out to understand the characteristics of process systems that will benefit from the replacement of PID with single-input single-output (SISO) MPC, and the benefits achieved by such replacements were studied. To achieve this, benchmark SISO systems with certain complex dynamics were identified and simulations carried out to compare the performances of PID and MPC when applied to such systems. From the simulation studies carried out it was discovered that processes characterised by certain dynamics are likely to benefit from the implementation of SISO MPC. Amongst these include processes with dead-time dominant dynamics, higher order systems and processes characterised by both slow and fast dynamics. Even though some of these problems can be solved using other extensions of the PID controller, this study concentrated on standard PID controller. This is because studies have shown that practitioners are reluctant in applying such extensions for example the smith predictor. Even the derivative term of standard PID controllers is rarely used because of the additional difficulty associated with tuning among other reasons. In the chapter, it was demonstrated that MPC is simpler to apply than PID and gives a performance that is at least as good. However, the limitation is that dedicated software is required for MPC, but with recent advances, the technology could be integrated with relative ease on to programmable logic controllers (PLCs) and other low level control hardware.

In Chapter 5, an industrial case study was carried out to verify the findings of Chapter 4 in an industrial setting. In agreement with the simulation studies of Chapter 4, significant benefits were obtained by replacing an existing PID controller with an MPC in the studied industrial process. Moreover, the replaced PID controller was not a standard implementation. It had a feed forward lead lad compensator in place and significant effort was used to tune the controller. It was therefore considered to be of very good performance. MPC design was achieved in a systematic way using a
mechanistic model to minimise disturbances due to plant test and trials. Variability analysis was also used to estimate expected performance improvements and ensure the viability of the control improvement project. The results of Chapters 4 and 5 show that, with the right infrastructure in place, there are incentives for using SISO MPC as a replacement for PID on suitable processes.

Developing design procedures that will reduce the engineering effort and costs associated with MPC will motivate more use of the technology making it a widespread rather than a niche application. In line with the objectives of this thesis, a novel approach of achieving these reductions was developed in Chapter 6. In the chapter, a method of designing MPC from routine plant data (without plant tests) was developed. This contribution is in an area of active research as there have been recent efforts to reduce both the cost and engineering efforts associated with model based control. Building on recent results in closed-loop system identification, a novel method for tuning SISO MPC without plant tests was developed. The developed algorithm was successfully applied to process models and tuned to obtain improved performance. Using the developed method, the difficulty associated with MPC was drastically reduced, and tuning to achieve improved performance became less daunting. The arduous task of system identification and disruptions due to plant tests were eliminated. The results obtained are promising and can be good starting points for achieving industrial applications. This is in line with earlier research which suggested that a good starting point for designing MPC controllers is to begin with controllers that have similar/same performance as the existing controller. The method could also open up MPC implementation to non experts.

7.2 Recommendations

From the results of research carried out in this thesis, a number of areas for further research are highlighted.

7.2.1 Performance of efficient MPC algorithms

The computational requirement of MPC has been a major source of concern especially in low level control hardware such as PLCs and other devices with limited computational capabilities such as field programmable gate arrayss (FPGAs). There have been research efforts in developing efficient MPC algorithms for implementation in
such devices, example [160]. A research direction will be to understand the effect of using such efficient algorithms on the benefits highlighted in Chapter 4. It is worth investigating to see if benefits achievable with traditional MPC formulations (such as the one used in this work) are affected by using such algorithms. And if there are any performance losses due to the use of these efficient algorithms, then it is also important to find out if the computational savings justify such losses. Some steps that could be taken to achieve this include:

1. Identify suitable efficient MPC formulations to be used for comparison and ensure they are suitable for use with benchmark systems identified for the studies. If they are not suitable, steps can be taken to extend them to accommodate such systems. Typical algorithms are given in [160, 161]

2. Decide on performance measures to use. And also develop a criteria for deciding when achieved speed and simplicity of controller outweighs loss in performance.

3. Carry out systematic studies to compare the benefits and shortcomings of the controllers and come up with conclusions based on observations.

### 7.2.2 Analysis of error and bias in routine data MPC (RMPC)

Another research direction will be to carry out analysis of the bias observed in estimates of MPC-parameters in Chapter 6. An approach will be to consider adding disturbance modelling to the algorithm to study its effect on accuracy of estimates and bias. This could also be achieved by considering the following steps:

1. Carry out studies to understand if bias in estimates of MPC-parameters is from the Simulated Annealing (SA) algorithm, due to the effect of feedback or due to non-modelling of disturbances. The findings of this will then dictate the next steps to be taken to cater for the bias.

2. If bias is due to the effect of feedback (which is most likely going to be the case). Then methods of solving the problem could be considered. An approach will be to consider the inclusion of disturbance modelling in RMPC formulation. And then to study the resulting effects on estimates of RMPC-parameters.

3. If bias is found to be due to problems from SA algorithm, then methods of improving the selection of SA parameters can be explored. Results could also
be compared with results obtained by using other meta-heuristic methods such as genetic algorithm (GA).

4. If bias is found to be due to both SA and feedback, then all the steps identified above could be pursued to solve the problem.

### 7.2.3 Extension of RMPC to multi-input multi-output (MIMO) and higher order processes

Since benefits of model predictive control are more in MIMO systems, another research direction would be to explore ways of extending the developed RMPC method to two-input two-output (TITO) systems and subsequently MIMO processes. An approach would be to consider building such controllers in a distributed/decentralised fashion. Methods of modelling interaction could be explored to cater for coupling between loops. The method developed in this work also assumed that the process was adequately modelled by a first order plus dead-time (FOPDT) model. Which is usually but not always the case for all industrial processes. Therefore, it would be beneficial to extend the method to cater for higher order models. Extensions for second order plus dead-time (SOPDT) could be initially considered. This could later be generalised to general discrete time model structures. The discrete time formulation is more likely to accommodate disturbance modelling in a seamless manner and could be used for processes with more complex dynamics. Some steps that could be followed to achieve this include:

1. Develop a system of characterising loop interactions using existing methods such as relative gain array (RGA) and singular value analysis (SVA) and/or their extensions. A criteria for deciding if a loop should be implemented in a distributed or decentralised fashion needs to be developed.

2. Develop an interaction modelling procedure that could easily be incorporated in the extended RMPC for MIMO systems (starting with TITO processes). A method could be developed based on symbolic signal flow graph [162] – this would be suitable for large systems with different degrees of coupling.

3. Extend RMPC to SOPDT and then to higher order systems. More general prediction error models could be considered.
4. Application of the developed MIMO RMPC to simulation benchmarks. Example of benchmarks that could be considered include; the binary distillation column [163], the fluid catalytic control unit (FCCU) [164] and the Tennessee Eastman process [165].

7.2.4 Application to industrial process

The RMPC results obtained through simulation are promising. While simulation is helpful, especially in checking the feasibility of developed methods. It is imperative to verify applicability of developed techniques experimentally and/or in real applications. Exploiting applicability in real time applications is a fundamental direction for future work. For example, the case study presented in Chapter 5 or a similar industrial process could be used. However, for the case study presented in Chapter 5, extensions are needed to cater for measured disturbances. The problem in Chapter 5 can also be formulated as a multi-variable control problem and used to verify extensions of RMPC which can be deployed on the process incrementally in a decentralised/distributed fashion. Application could be straight forward for SISO processes. But in the case of MIMO systems, results from Section 7.2.3 may be required. Some possible steps that could be taken to achieve this include:

1. Identify an industrial process that is likely to benefit from the application of MPC and take note of the performance of existing PID controllers. The industrial case study in Chapter 5 could be used. Other laboratory scale processes such as the quadruple tank [161] could be also be considered.

2. Apply standard MPC and quantify performance over the existing PID controller. The case study in Chapter 5 fits in directly here, since both PID and MPC have already been applied on the process. The quadruple tank process also fits in directly as it has been used to compare distributed MPC (DMPC)/decentralised MPC (dMPC) performance [107] can also be used.

3. Obtain plant data from any of the selected plants and carry-out preprocessing to remove areas of data with set point changes and other unwanted disturbances.

4. Design RMPC using preprocessed data; the algorithm developed in Chapter 6 or improved versions of the algorithm could be used. Steps to ensure the developed controller would give stable response if applied to the plant must be taken.
5. Develop procedures for smooth transition from PID control to RMPC i.e. bump-less transfer control [22]. This could involve studying existing methods for bump-less transfer control, developing decision criteria for when to switch to/back from RMPC.

6. Develop/explore methods for safety testing of RMPC before on-line implementation. Some existing methods that could be adapted include SIMPLEX [166] and ORTEGA [167]

7. Implementation of controller on process. Improve controller tuning and compare performance with that of traditional MPC and existing control scheme.
Bibliography


Appendix A

Complex Systems

A.1 Dead-time dominant systems with smith predictor

The lowest value of IAE can be achieved using an ideal Smith Predictor. However, this is only possible with an ideal model and infinite actuation. Most practitioners seldom implement the smith predictor as it drastically increases the operational complexity of controllers. Loosely tuned PID controllers are often used. For robust tuning, \( \tau_c \leq \theta \) is recommended [33]. Selecting \( \tau_c = \theta \) gives a very good compromise between IAE, TV and robustness [5]. Therefore, this value of \( \tau_c \) will be used.

Simulation was carried for \( G_\theta \) with \( \tau = 1 \) and \( \theta \) varied over the range \([0, 0.5, 1, 2, 3, 4, 5, 6]\). PID controllers, Smith Predictor PI controller and MPC controllers were applied. IAE and TV were used as performance measures for both set-point and load disturbance. Three different predictive controllers, \( MPC_1, MPC_2 \) and \( MPC_3 \) with move weights, 0.1, 1 and 10 respectively were applied. Prediction and control horizons used were \( 5\tau + \theta \) and 3 respectively. The plots of IAE and TV against \( \frac{\theta}{\tau} \) and sample response for \( \frac{\theta}{\tau} = 6 \) are shown in Figure A.1. The relatively conservative \( MPC_3 \) has a lower IAE than the PID controller when \( \frac{\theta}{\tau} > 2 \). A modest \( MPC_2 \) outperforms the PID when \( \frac{\theta}{\tau} \) is approximately 1.5 at a lower value of TV. The smith predictor has a similar performance with the aggressive \( MPC_1 (rw = 0.1) \). But the predictive controller has a lower TV.

A.1.1 Conclusion

The results show that, for more active MPC, when compared with PID you get improvements when \( \frac{\theta}{\tau} > 0.5 \). The smith predictor is commonly found in most process
Figure A.1: Plots of IAE, TV and a sample response

automation software. However, practitioners have been reluctant to implement the smith predictor as controller tuning becomes more complex. The simulation results have shown that MPC achieves similar performance as the smith predictor. However, this performance was achieved using MPC with less control activity (smaller value of TV). This can lead to longer lasting valves and actuators. Therefore significant benefits can be achieved by implementing MPC at the regulatory layer for dead-time dominant systems.
A.2 Higher order Systems

Figure A.2: Plots of manipulated and control variables for $G_n(n = 1)$
Figure A.3: Plots of manipulated and control variables for $G_n(n = 2)$
Figure A.4: Plots of manipulated and control variables for $G_n(n = 3)$.
Figure A.5: Plots of manipulated and control variables for $G_n(n = 4)$
Figure A.6: Plots of manipulated and control variables for $G_n(n = 8)$
Figure A.7: Plots of manipulated and control variables for $G_\alpha (\alpha = 0.1)$
(a) no constraints

(b) with constraints on $\Delta u$ and $u$

Figure A.8: Plots of manipulated and control variables for $G_\alpha(\alpha = 0.2)$
Figure A.9: Plots of manipulated and control variables for $G_\alpha(\alpha = 0.5)$
Figure A.10: Plots of manipulated and control variables for $G_\alpha(\alpha = 1)$
Appendix B

Industrial case study
B.1 Mechanistic model plots

(a) PH and MV with nominal parameters

(b) Step disturbance

Figure B.1: Plots of PH and MV for both MPC and PID due to Step disturbance
Figure B.2: Plots of PH and MV for both MPC and PID due to Step disturbance

(a) PH and MV for low parameter values

(b) PH and MV with high parameter values
Figure B.3: Plots of PH and MV for Step disturbance in MATLAB
B.2 Plant test data plot

Figure B.4: Plant test data at pH around 1.7
Figure B.5: Plant test data pH around 1.7
B.3 MPC Horizons

In the example shown in this figure, the number of samples over which controller calculates moves is 30. The controller can make a move at each sampling time during the first 15 samples but can only make one move within 5 sampling instances during the last 15 samples i.e 3 moves.

- Constrained Horizon: The number of future samples over which the controller calculates control moves.

- Compressed Horizon: The number of samples in the constrained horizon that are compressed together for computational efficiency

- Compression Width: Future samples in the compressed horizon that are grouped into blocks. The compression width defines how many samples there are in a block.