Sampling and Quantization in Nonlinear Filtering and Control

Mauricio Esteban Cea Garrido Bsc. Msc.

A thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

School of Electrical Engineering and Computer Science

The University of Newcastle
Callaghan, N.S.W. 2308
Australia

October, 2012
The thesis contains no material which has been accepted for the award of any other degree or diploma in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. I give consent to this copy of my thesis, when deposited in the University Library, being made available for loan and photocopying subject to the provisions of the Copyright Act 1968.

I hereby certify that the work embodied in this thesis contains published paper/s/scholarly work of which I am a joint author. I have included as part of the thesis a written statement, endorsed by my supervisor, attesting to my contribution to the joint publication/s/scholarly work.

I hereby certify that the work embodied in this thesis contains published paper/s/scholarly work of which I am a joint author. I have included as part of the thesis a written statement, endorsed by my supervisor, attesting to my contribution to the joint publication/s/scholarly work.

________________________________________

Mauricio Esteban Cea Garrido

June, 2012
ACKNOWLEDGEMENTS

This thesis is the result of the help and support of many people in my life.

I would like to first acknowledge my wife Carmen Gloria Ormeño Vittoriano for her unconditional support along this road. You have been really determinant in who I am today and I will never forget what you did for me during this time. I hope someday to be able to repay you for all your love and help.

Thank you Graham for your wonderful supervision, I would have never made it without your help. You have been much more than a supervisor, you have been a friend and I am really grateful of getting to know you. Dianne and Jayne, before I came to Australia, Prof. Mario told me you would be like my mother here and you have been more than that. There are no words to thank you for your support, concern and the love you have showed to me in my journey. Katrina, you have been a terrific co-supervisor and friend, you have taught me more than you can imagine. Arie Feuer, David Mayne, Claus Müller, Maria, Jose de Dona, James you have all worked with me and helped me at some point. I have learned from everyone of you many lessons and experience that are greatly appreciated.

I am also extremely grateful for the opportunity to work with Ericsson AB in Sweden. In particular Linda Brus, Stephen Craig, Caroline Jacobson, Erik Geijer Lundin, Stéphane Tessier and Torbjörn Wigren provided ideas and motivation for my work.

I would also like to acknowledge CONICYT for their sponsorship under the scholarship “Becas Bicentenario de Doctorado en Australia o Nueva Zelanda convocatoria 2008” and the people involved.

Next, I would like to acknowledge my friends here in Newcastle. Thank you Alejandro, Aurelio, Damian, Beatrice, Nick, Anisa, Rocío, Ramon, Hope, Daniel for all the good memories and good
times we had together in our many social events. Special thanks to Eduardo for the best churrascos I have ever had. Thank you Diego for bringing a piece of home into my life in Newcastle. Thanks to Juan Carlos for his honest advice on this journey. Thanks Boris for all the help and uncountable funny stories you shared with us. Thanks Ricardo and Katherine for your friendship and advice.

To God and my family in Hunter Bible Church you have been essential in my life and you cannot imagine what you have done for me during this time in Australia. Josh and Giuliana, Adrian and Laetitia, Sam and Prue, Pete and Juanita, Geoff and Dawn and everyone in my growth groups, and soccer teams. You have all being incredible and I will never forget you.

To my Mom Vicky and my Dad Mauricio, you made me who I am today, you gave me the tools and love I needed to make it this far in life and I will always be grateful for that. Even though you are on the other side of the world, you have always been with me in my heart and not a day has gone by without thinking of you. To my little sister Tiky and brother Ivan I always remember you with the greatest love and kindness. Being away from you made me think of all the things we should have done all these years and I wish someday to do with you. Also, I would like to acknowledge my in-laws, Sarita and Carlos, thank you very much for all the kindness you have shown me and for allowing me to be with your wonderful daughter.

Finally to my friends in Chile, you have been supporting me more than you realize all this time. Even with simple things you tried to make me feel closer to home and I will never forget that. Special thanks to Stjpe, Bart, Nelson, Cucho, Seba, Valeria, Duran, Jose, and so many more.

I wish I have not forgotten anyone and I am deeply sorry if I have. Just remember you will always be in my heart.
To my wife

Yoyi,
CONTENTS

Abstract 1

1 Introduction 3

1.1 Overview of the Thesis Content and Contributions 3

1.2 Associated Publications 7

1 Nonlinear Filtering 9

2 Overview of Nonlinear Filtering 11

2.1 Continuous Time Nonlinear Model 14

2.1.1 Linear Systems 14

2.1.2 The Nonlinear Case 16

2.2 Design of Sampling System 16

2.2.1 Direct Sampling of the Output 16

2.2.2 Sampling after passing through an Averaging Anti-aliasing Filter 16

2.2.3 Sampling the Integral of the Output 17

2.3 Sampled Data Models for Continuous Systems 17

2.3.1 The Conventional Linear Sampled Data Model 18
2.3.2 Incremental Form of the Linear Sampled Data Model . . . . . . . . . . . . 19
2.3.3 Approximate Sampled Data Model for Linear Case . . . . . . . . . . . . 21
2.3.4 Exact Nonlinear Sampled Data Models . . . . . . . . . . . . . . . . . . . 22
2.3.5 Approximate Sampled Data Nonlinear Model . . . . . . . . . . . . . . . 22
2.4 Review of Discrete Linear Filtering . . . . . . . . . . . . . . . . . . . . . . . . 23
2.4.1 The Exact Linear Case . . . . . . . . . . . . . . . . . . . . . . . . . . . 23
2.4.2 Incremental Form of the Exact Sampled Data Linear Filter . . . . . . . . 24
2.4.3 Approximate Sampled Data Linear Filter . . . . . . . . . . . . . . . . . . 25
2.5 Sampled Data Nonlinear Filters . . . . . . . . . . . . . . . . . . . . . . . . . . 25
2.5.1 Case 1: Exact Filter; Exact Model . . . . . . . . . . . . . . . . . . . . . 26
2.5.2 Case 2: Exact Filter; Approximate Model . . . . . . . . . . . . . . . . . . 27
2.5.3 Case 3: Approximate Filter, Exact Model . . . . . . . . . . . . . . . . . . 27
2.5.4 Case 4: Approximate Model; Approximate Filter . . . . . . . . . . . . . 28
2.6 Conclusions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28

3 Temporal Sampling and Modelling Issues . . . . . . . . . . . . . . . . . . . . . . 31
3.1 Down-sampling . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 32
3.1.1 Example . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 33
3.2 Up-sampling . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35
3.2.1 The nonlinear case . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35
3.2.2 The Linear Case . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 38
4.5.2 Simulation Example ........................................ 79
4.6 MDF applied to System Identification in Chemical Processes .............. 85
4.7 Conclusion .................................................. 91

II Networked Control ........................................ 93

5 Overview of Networked Control .................................. 95
5.1 Control with Communication Constraints ................................ 95
5.2 WCDMA Broadband Communications ................................. 96
5.3 Simulator .................................................... 98
  5.3.1 Notation ................................................. 100
  5.3.2 Inner loop ............................................... 100
  5.3.3 About the UE ........................................... 102
  5.3.4 Interference ............................................. 102
  5.3.5 Absolute, relative and serving grants ............................ 104
  5.3.6 UE TFC selection ....................................... 106
  5.3.7 Traffic model ........................................... 109
  5.3.8 Scheduler ............................................... 111
  5.3.9 Other Relationships Used in the Simulator ....................... 112
5.4 Conclusions ................................................. 113

6 Adaptive Quantizer ........................................ 115
6.1 Inner Loop Power Control ............................................. 115
6.2 An MPC Based Technique for Channels with Limited Data Rate ......... 118
   6.2.1 Model Predictive Control ....................................... 118
   6.2.2 A New Paradigm for Inner Loop Power Control .................. 119
   6.2.3 Properties of the Control Law .................................. 121
6.3 Performance Comparison ............................................. 123
6.4 Robustness .............................................................. 128
   6.4.1 Experimental Setup ............................................. 128
   6.4.2 Erroneous bits, no information in the Base Station ............... 129
   6.4.3 Upper bound for power increments ................................ 132
   6.4.4 Performance with bit errors .................................... 134
   6.4.5 Overall results .................................................. 138
   6.4.6 Erroneous bits with information at the Base Station .............. 142
   6.4.7 Summary ....................................................... 143
6.5 Conclusions ............................................................ 144

7 Adaptive Quantizer - Unstable Systems .................................. 145
   7.1 A Class of Open Loop Unstable Systems ............................ 147
   7.2 Nonlinear Decoder Model Predictive Controller (NDMPC) ............. 148
   7.3 Motivational Example .............................................. 151
   7.4 Analysis .............................................................. 152
7.5 A Non-uniform Horizon Version of the NDMPC Algorithm .......................... 157

7.5.1 Examples .................................................. 157

7.6 Conclusions .................................................. 159

8 Buffer Estimation ........................................... 161

8.1 Overview of the Happy Bit ................................ 161

8.2 Using the Happy Bit for Buffer Size Estimation ................. 162

8.3 Simulation Studies .......................................... 164

8.3.1 Short number of TTI’s ................................. 164

8.3.2 Long number of TTI .................................. 165

8.3.3 Recommendation .................................... 166

8.4 Adaptive nTTI in Happy Bit ............................... 167

8.4.1 Motivation ............................................. 167

8.4.2 A Possible Algorithm ................................ 167

8.4.3 Simulation Results .................................... 168

8.5 Conclusion .................................................. 172

9 Conclusions ...................................................... 173

9.1 Summary of Contributions .................................. 173

9.1.1 Chapter 2 ............................................... 174

9.1.2 Chapter 3 ............................................... 174
Contents

9.1.3 Chapter 4 ......................................................... 174
9.1.4 Chapter 5 ......................................................... 174
9.1.5 Chapter 6 ......................................................... 175
9.1.6 Chapter 7 ......................................................... 175
9.1.7 Chapter 8 ......................................................... 175
9.2 Future Work ......................................................... 175

Bibliography ......................................................... 177
The thesis describes contributions made in two areas, namely, aspects of nonlinear filtering and control over communication channels. Specifically, the thesis is concerned with addressing the impact of temporal and spatial quantization on the performance of estimation and control algorithms.

Regarding nonlinear filtering, emphasis is placed on technical and theoretical issues regarding the impact of sampling and spatial quantization. New techniques of upsampling and downsampling are developed to address specific computational issues. Also, extensions to Lebesgue sampling are explored. In addition, a new class of nonlinear filtering algorithms, called Minimum Distortion Filters, is developed based on the use of Vector Quantization. The effectiveness of the new filtering algorithms are illustrated, by using it on a wide variety of problems including system identification, state estimation for chemical processes and radar tracking.

Regarding control over communication channels, the focus is primarily on the practical problem of inner loop power control in broadband WCDMA. This control loop is characterized by the use of a one-bit-per-sample control law. This restriction severely limits the achievable performance. To address the problem, a novel adaptive quantizer is developed in the thesis. The scheme is shown to lead to significant performance improvements. The scheme has also been extended and applied to the problem of stabilizing an open loop unstable system over a bit rate constrained channel. Other control problems in WCDMA are also explored including the use of the “Happy Bit” to signal buffer sizes for the purposes of scheduling. This control loop is also subject to a one-bit-per-sample constraint. Typically, the one bit is used for a different purpose in current implementations. Here we develop an adaptive scheme based on the Happy Bit for communicating the buffer size.
Introduction

This chapter provides an overview of the topics to be discussed in the thesis. No significant literature review is presented here. Instead, the relevant literature will be discussed at the beginning of each part of this two part thesis.

1.1 Overview of the Thesis Content and Contributions

Computational resources have increased dramatically in the last two decades. Every day more powerful computational systems are deployed in industry allowing for more complex problems to be solved. Hence, engineers have been moving from the traditional linear paradigm to dealing with more accurate nonlinear formulations of problems. Also, the growing sophistication of technology has given birth to new problems. It is becoming more common to find control systems that are distributed on a very large space. Plants and controllers are now not located next to each other and even sensors can be distributed within meters, kilometres or even across continents. Hence, the new element of a communication channel has become important in the networked control community.

This thesis is divided, for pedagogical reasons, in two parts. The first part deals with nonlinear filtering. The second part deals with aspects of networked control. Both parts are linked by the common theme of addressing the impact of temporal and spatial quantization on the performance of estimation and control algorithms.

Nonlinear filtering has been traditionally studied within the scope of mathematics. This is due to
its complex and interesting nature. However, it has not played an important role in the engineering field until recently, when computational capability has become accessible to more researchers and industries. As a consequence, nonlinear filtering has developed to the point where it is out of reach of practicing engineers. This is specially true of the continuous time case which relies upon deep mathematical concepts. Moreover, there is a clear disconnect between the understanding of continuous time and discrete time\textsuperscript{1} filters, despite the fact that they both aim to address the same, or at least similar, questions.

This thesis is aimed at closing this gap and allowing engineers to embrace this problem in an informed and clear fashion. To be able to understand the topic of nonlinear filtering one has to be capable of identifying the nature of the source of errors and difficulties arising in this problem. In this context, models are the first source of difficulties.

Continuous time models are, in some sense, idealized models. The assumptions used in developing the models are often inconsistent with real world applications. It is impossible to interact in a continuous way with real systems because of the nature of sensors, actuators and controllers. All of these require that one samples the continuous time system output, therefore leading to discrete time representations. As consequence, depending on the particular choice, one will end up with very different models. For instance, most continuous time systems cannot be discretized in an exact way, save for the special case of linear systems and some very unique nonlinear models. Furthermore, even the discretization of the same system depends on how the system is sampled. Also, there exist “traditional” discrete-time system models and “incremental” discrete time system models. This distinction adds further confusion to the understanding of the nonlinear case. In Chapter 2, these questions are addressed, leading to a structured framework in which the appropriate connections are established.

Nonlinear filtering has two major problems. Firstly the continuous and discrete time case seem to be disconnected from each other. Secondly it is impossible to find closed form solutions due to the high dimensional, possibly infinite, integrals involved. The first problem is addressed in Chapter 3. The discussion of this issue is non-trivial and requires an understanding of models,

\textsuperscript{1}In the sequel we will abbreviate these to continuous and discrete respectively, provided that time is clear from the context.
1.1 Overview of the Thesis Content and Contributions

numerical errors and filtering. The results presented here combine all of these concepts. The results are focused around the issue of sampling and especially the sampling period used in the discretization step. Connections between the continuous and discrete time cases are addressed. Also connections between discrete filters at different sampling rates are developed. Thus, Chapter 3 deals with temporal sampling and modelling issues in nonlinear filtering.

Chapter 4 addresses the second problem mentioned above, namely numerical issues. The high dimensionality of the integrals involved in nonlinear filters has been a question of interest for, at least, 4 decades. The most successful approaches have been the use of Monte Carlo methods. The resultant algorithms are also known as particle filters. Even though, these methods are widely used there is a gap in how to address this problem in a deterministic rather than stochastic fashion. In general, there is a good understanding of how to perform temporal sampling. For example, it is doubtful that anyone would think of randomly sampling the system in time. However spatial sampling remains somewhat of a mystery. With this as background, Chapter 4 is concerned with spatial sampling aspects of the nonlinear filtering problem. In particular, a novel deterministic spatial sampling technique is developed and its connection with temporal sampling is explored.

Part 2 of the thesis considers aspects of the control of broadband telecommunication systems.

Telecommunication systems are a good example of a system where plant and controller are far from each other. In this architecture, the plant, the User Equipment (UE), is far from the controller at the Base Station. The distance can range from a few meters to up to a few kilometres in rural areas. Indeed, there is a remote location in Australia where the distance is several hundred kilometres. Also, due to technological constraints the signals cannot be sent over the channel as an analogue signal, and therefore, the signals need to be quantized from a particular codebook. These problems motivate Chapter 5, where background on networked control and WCDMA communication systems is provided.

Networked control has been a topic of strong interest in the control community during the past two decades; much of the effort has focused on finding stability conditions for general architectures. In this context, many classes of encoders/decoders and quantizers have been defined. Memoryless and infinite memory quantizers, linear and nonlinear quantizers and white noise models for
quantizers are only a few example of what has been developed. Most of the existing results focus on the channel capacity, expressed as bits per sample, that can be sent. The channel capacity is considered to be an average constraint. Motivated by WCDMA systems, as presented in Chapter 5, the interest in Chapter 6 is on marginally stable systems having a very limited bit rate of 1 bit per sample. In addition, not only stability is the centre of attention but also performance and the use of nonlinear quantizers in conjunction with model predictive control theory. Chapter 7 of the thesis is aimed at extending the tools developed in Chapter 6 to unstable systems. Stability and performance are addressed. Chapter 8 addresses a related problem in broadband telecommunications namely that of transmitting information about buffer sizes from the User Equipment to the Base Station. Again a one bit per sample constraint arises.

Finally, Chapter 9 summarizes the key findings of the thesis and presents suggestions for future research.
1.2 Associated Publications

This thesis is supported by a series of papers in which the candidate is a joint author. In all cases the contributions of each author are indicated.


[P3] G.C. Goodwin (50%), J.I. Yuz (20%), J.C. Agüero (20%) and M.G. Cea (10%) “Sampling and sampled-data models”, American Control Conference, Baltimore 2010, U.S.A.


[P5] G.C. Goodwin (60%) and M.G. Cea (40%) “Temporal and Spatial Quantization in Nonlinear Filtering”, The International Symposium on Advanced Control of Industrial Processes, Hangzhou 2011, P.R. China.

[P6] M.G. Cea (50%), G.C. Goodwin (30%) and Claus Müller (20%) “A Novel Technique based on up-sampling for addressing Modeling Issues in Sampled Data Nonlinear Filtering”, 18th IFAC World Congress, Milan 2011, Italy.


Laureate Professor Graham C. Goodwin
June, 2012
Part I

Nonlinear Filtering
This chapter presents background relevant to the first part of the thesis. There exist a vast literature on this topic. Hence, the description here will be focused on particular issues concerning descriptions of models and the definitions of continuous-time and discrete-time models. In addition, connections to the filtering problem are made explicit. This helps to establish the ground work for the analysis presented in subsequent chapters.

For the linear gaussian case, it is well known [6, 7, 62], that the Kalman Filter [79] provides an elegant and simple solution to the filtering problem. However, many problems met in practical applications are neither linear nor gaussian. For example, in Chemical Engineering, almost all systems are described by nonlinear physical models. Thus, there has been long term interest, spanning the past 5 decades, in extending Kalman Filter theory to nonlinear, and or, nongaussian problems.

A natural idea to extend the Kalman filter to nonlinear systems is to linearize the system about the current state estimate. This leads to the idea of the, so called, Extended Kalman Filter (EKF) [6, 75]. This class of algorithm is widely used in practice but is largely ad-hoc. There are a number of extensions of the idea. For example, the Unscented Kalman Filer (UKF) is aimed at improving performance by passing a finite set of, so called, sigma points, through the nonlinearities and averaging on the other side [76, 77]. It is also possible to use optimization to find the maximum likelihood state estimate for nonlinear problems [60]. These methods are aimed at estimating only the first (and perhaps second) moment of the posteriori state distribution. In the first part of the thesis the focus is on methods that allow one to characterize the complete posterior distribution.
The subject of nonlinear filtering is a mainstream topic in mathematical control theory. Development of a rigorous theory for continuous time nonlinear filtering can be traced back to the work of Stratonovich \cite{127,128} in Russia in the late 1950’s. Related ideas were developed in parallel work in the United States by Kushner \cite{83,84,85}. Shiryaev developed similar ideas using Itô Calculus in the late 1960’s \cite{124}. Other early contributions included those of Bucy \cite{23}, Won-han \cite{138}, Kailath \cite{78}, Fujusaki, Kallianpur, Kunita \cite{48}, Duncan \cite{40,41}, Mortensen \cite{105} and Zakai \cite{142}. Other important contributions include the concept of robust filtering introduced by Clark in the 1970’s. In the early 1980’s Benes studied classes of nonlinear filtering problems which had finite dimensional representations. Chaleyat-Maurel, Michel, Hazlasewenkel, Markus and Sussman have proven that generally nonlinear filters are infinite dimensional. For further background, the reader is referred to \cite{93,92} and the recent book of Bain and Crisan \cite{19} which provides an introduction to mathematical theory of Nonlinear Filtering and gives an overview of relevant contributions.

Much of the technical difficulty (and mystique) of nonlinear filtering is associated with the continuous time case. On the other hand, the discrete version of the problem is widely used in practice. The structure of the continuous and discrete nonlinear filter equations are very different. For example, it is traditional to express the discrete nonlinear filtering equations as an absolute update via the Chapman-Kolmogorov equation and Bayes’ rule \cite{19}. This contrasts with the continuous nonlinear filter which is expressed in incremental form via the Fokker-Planck equation and the Kushner-Stratonovich equation \cite{19}.

The above, very brief, historical review raises many questions which will be of concern to those who are interested in using Nonlinear Filtering in practice. Some of these question are:

1. Since the continuous time case is mathematically rigorous, why bother with the discrete time case?

2. Since the discrete time case is easy to understand, why bother with the continuous time case?

3. Since most physical models are inherently continuous time in nature, how do we obtain appropriate discrete models if we want to utilize a discrete filter?
4. Do the discrete filtering equations converge, in some meaningful sense, to the continuous filtering equations as the sampling rate increases?

5. What issues underly the appropriate choice of sampling interval in discrete nonlinear filtering?

This part of the thesis is aimed at providing answers to these questions. To guide the reader on what is to follow, preliminary answers to the above questions are:

1. In practice all real world filters use sampled data, thus one needs to understand the discrete-time case.

2. Standard forms of the discrete algorithms lead to paradoxes and inconsistencies if one tries to use them with fast sampling rates. Hence, some knowledge of the continuous time case is necessary to appreciate what happens when fast sampling is deployed.

3. There exists an easy way to progress from a continuous time model to a discrete time model of arbitrary desired accuracy. This is important, especially when sampling intervals are long or data arrives at non-uniform time spacing.

4. The discrete filtering equations converge to the continuous time filtering equations. To appreciate the nature of this convergence requires an understanding of the role of the anti-aliasing filter in the discrete case.

This part of the thesis is the culmination of ideas presented in a sequence of papers co-authored by the candidate—see [26 58 28 63 57 27].

The remainder of this chapter is organized as follows. Section 2.1 introduces the continuous time nonlinear filter. Next, Section 2.2 discusses different sampling techniques of continuous time systems. Section 2.3 describes sampled data models obtained by applying the sampling techniques of Section 2.2. Section 2.4 reviews the discrete time linear filter. Finally, the sampled data nonlinear filter is discussed in Section 2.5.
2. Overview of Nonlinear Filtering

2.1 Continuous Time Nonlinear Model

Since the prime interest in the current chapter is the impact of temporal sampling on nonlinear filtering, it is appropriate to begin with a continuous time model. The linear case is first reviewed.

2.1.1 Linear Systems

The standard linear continuous time stochastic model takes the following form:

\[
\frac{dx}{dt} = A_c x + \frac{d\omega}{dt} \tag{2.1}
\]

\[
y = \frac{dz}{dt} = C_c x + \frac{d\nu}{dt} \tag{2.2}
\]

where \( x \in \mathbb{R}^n \) is the state vector and \( z \in \mathbb{R}^m \) is the measured output vector. In (2.1), (2.2), \( d\omega/dt, d\nu/dt \) represent independent continuous time "white noise" processes of intensity \( Q_c \) and \( R_c \) respectively. (We will explain the term “intensity” below.) For the moment, we can think of the autocovariance of \( d\omega/dt \) as being \( Q_c \delta(t) \) where \( \delta(t) \) is the dirac delta function. However, an important technical observation is that continuous time white noise does not exist in any meaningful sense. (Indeed, if the auto-covariance of \( d\omega/dt \) is \( Q_c \delta(t) \) then the variance is infinite). Actually, a more insightful description is to use spectral density. Spectral density is the Fourier transform of the autocorrelation i.e. we formally have:

\[
\Phi \left( \frac{d\omega}{dt} \right) = \int_{-\infty}^{\infty} Q_c \delta(t) e^{-j\omega t} dt = Q_c \tag{2.3}
\]

Thus, we see that \( Q_c \) describes the spectral density of \( d\omega/dt \). The concept of Spectral Density allows one to intuitively replace the notion of “white noise” by the notion of "broad band" noise which has constant spectrum over a wide (but not infinite) bandwidth. Indeed, it turns out that “whiteness” of the process and measurement noise is largely irrelevant to the operation of a filter in practice. What is actually important is that the spectrum be substantially constant in key regions. This issue is discussed in detail in [56] where the term “noise intensity” is associated with spectral density in key spectral regions. Indeed, this exposes a major difficulty with the common
practice of using variances in the discrete time case. For example, say one has broadband (but non-white) noise of spectral density $Q$ covering a bandwidth of $W$, where $W < 1/\Delta$, then the associated variance, $V$, is equal to the area under the spectrum, i.e. $V = WQ$. If one uses spectral density to describe the noise, then no difficulties will be encountered since the noise intensity has been correctly described. However, if one uses variance to describe the associated filter, then the variance must be artificially scaled to $\tilde{V} = V/W\Delta$ in the various filter calculations, otherwise, the associated filter will perform badly due to underestimation of the noise intensity.

A related problem is that variance does not indicate the difficulty of an estimation problem. For example, say one is sampling very fast, then $1/\Delta$ will be large. In this case, a small noise spectral density could be associated with a very large noise variance. Yet, most of this noise power will lie at frequencies above the bandwidth of the system. Intuitively this part of the noise will not affect the filter performance. Again, it is only the spectral density in relevant parts of the spectrum that affects filter performance.

Thus, it is usually best to interpret noise intensity via spectral density rather than autocovariances in both continuous and discrete cases. This, aligns the discrete and continuous cases, since spectral density (i.e. incremental variance) is exclusively used in the continuous case. We will also see later that spectral properties are approximately invariant with respect to sampling, whereas variances need to be scaled.

To deal with some of technical issues mentioned above, equations (2.1) (2.2) are better expressed in incremental form as:

$$\begin{align*}
\frac{dx}{dt} &= A_c x + d\omega \\
\frac{dz}{dt} &= C_c x + d\nu
\end{align*}$$

$x \in \mathbb{R}^n$, $z \in \mathbb{R}^m$, $A_c \in \mathbb{R}^{n \times n}$, $C_c \in \mathbb{R}^{m \times n}$, $d\omega \in \mathbb{R}^n$ and $d\nu \in \mathbb{R}^m$ are the state, measured output, system matrices, process noise and measurement noise respectively. The initial state is assumed gaussian with $E\{\bar{x}_0\} = \hat{x}_0$ and $E\{(\bar{x}_0 - \hat{x}_0)^T (\bar{x}_0 - \hat{x}_0)\} = \hat{P}_0$. We assume that $\omega$ and $\nu$ are stationary vector Wiener processes with incremental covariance $Q_c dt$ and $R_c dt$ respectively. The matrices $Q_c$ and $P_0$ are assumed to be symmetric and positive semidefinite, and $R_c$ is assumed
to be symmetric and positive definite.

2.1.2 The Nonlinear Case

An appropriate nonlinear version of (2.4), (2.5) is

\[
\begin{align*}
\frac{dx}{dt} &= f_c(x)dt + g_c(x)d\omega \\
\frac{dz}{dt} &= h_c(x)dt + d\nu
\end{align*}
\]  

(2.6) (2.7)

where the processes \( \omega \) and \( \nu \) correspond to Brownian motion process having incremental covariance \( Q_c dt \) and \( R_c dt \) respectively. Recall that the same quantities \( Q_c \) and \( R_c \) can also be interpreted as spectral densities for \( d\omega/dt \) and \( d\nu/dt \) respectively.

2.2 Design of Sampling System

In section 2.1.1 the continuous time output was defined as \( dz/dt \). Here, the interest is in the form of the various equations when we take samples. However, one need to answer the question, “Samples of what?”. There are 3 possible options for the sampled output described below.

2.2.1 Direct Sampling of the Output

Naively one may believe that one could directly sample the continuous process \( dz/dt \). However, this choice is actually an infeasible option since the samples of the associated noise, \( d\nu/dt \), would have infinite variance! Thus, samples of \( dz/dt \) do not exist in any reasonable sense.

2.2.2 Sampling after passing through an Averaging Anti-aliasing Filter

An appropriate remedy to the difficulties described in section 2.2.1 is to pass \( dz/dt \) through an anti-aliasing filter prior to sampling. A common choice for such a filter is to simply average \( dz/dt \) over the sample period. Indeed, some form of averaging is inherent, in all low pass filters, typically
used as anti-aliasing filters. In the case of averaging, the sampled output satisfies:

\[
\bar{y}_k = \frac{1}{\Delta} \int_{k\Delta}^{(k+1)\Delta} \frac{dz}{dt} \quad (2.8)
\]

\[
= \frac{1}{\Delta} \{ z((k + 1)\Delta) - z(k\Delta) \} \quad (2.9)
\]

To obtain a notation for the sampled data case which resembles the continuous case, we define the (discrete) increment in \( z \) as

\[
dz^+ = z((k + 1)\Delta) - z(k\Delta) \quad (2.10)
\]

In this case, equation (2.9) can be rewritten as

\[
\bar{y}_k = \frac{1}{\Delta} \dz^+ \quad (2.11)
\]

### 2.2.3 Sampling the Integral of the Output

This option is closely related to the choice made in section 2.2.2. In the case of integration, one simply has

\[
y_k = \int_{k\Delta}^{(k+1)\Delta} \frac{dz}{dt} \quad (2.12)
\]

\[
= dz^+ = \bar{y}_k \Delta \quad (2.13)
\]

Note that (2.13) is simply a scaled version of (2.11). In the next two sections, the impact of sampling on the associated discrete time model is explored.

### 2.3 Sampled Data Models for Continuous Systems

Before considering the nonlinear case, it is helpful to review the special case of linear systems.
2.3.1 The Conventional Linear Sampled Data Model

If one begins with the linear continuous model \(2.4, \ 2.5\), then it is straightforward to obtain an “exact” discrete time model which accurately describes the properties at the sample instants. The averaging anti-aliasing filter \(2.9\) is used.

By integrating \(2.1, \ 2.2\), the exact discrete model takes the following form:

\[
\bar{x}_{k+1} = A_d \bar{x}_k + \bar{\omega}_k \\
\bar{y}_k = C_d \bar{x}_k + \bar{\nu}_k
\] (2.14)

\[
E\left\{\begin{bmatrix} \bar{\omega}_k \\ \bar{\nu}_k \end{bmatrix}^T \begin{bmatrix} \bar{\omega}_k \\ \bar{\nu}_k \end{bmatrix}\right\} = \begin{bmatrix} Q_d & S_d \\ S_d^T & R_d \end{bmatrix} = \Sigma_d
\] (2.16)

The noise sequences are white, zero mean, gaussian with joint covariance:

Note that \(S_d\) is not in general zero contrary to what is often assumed in the literature on the discrete filtering problem.

\[
A_d = e^{A_c \Delta} = I + A_c \Delta + (A_c \Delta)^2/2 + \ldots
\] (2.17)

\[
C_d = \frac{1}{\Delta} C_c A_c^{-1} (e^{A_c \Delta} - I) = C_c (I + \frac{1}{2!} A_c \Delta + \frac{1}{3!} A_c^2 \Delta^2 + \ldots)
\] (2.18)

and

\[
\Sigma_d = \bar{D} \left( \int_0^\Delta e^{\bar{A} \tau} \begin{bmatrix} Q_c & 0 \\ 0 & R_c \end{bmatrix} e^{\bar{A}^T \tau} d\tau \right) \bar{D}
\] (2.19)

where,

\[
\bar{A} = \begin{bmatrix} A_c & 0 \\ C_c & 0 \end{bmatrix} \Rightarrow e^{\bar{A} \tau} = \begin{bmatrix} e^{A_c \tau} & 0 \\ C_c \int_0^\tau e^{A_c \sigma} d\sigma & I \end{bmatrix}
\] (2.20)
2.3 Sampled Data Models for Continuous Systems

\[
D = \begin{bmatrix} I & 0 \\ 0 & \frac{1}{\Delta} \end{bmatrix}
\]  
(2.21)

Even though the above sampled system is an exact description for every finite \( \Delta \), the model is a source of conceptual and numerical problems when the sampling period decreases to zero. For example, it is readily seen, that as \( \Delta \to 0 \),

\[
A_d \to I \\
\sum_d \to \begin{bmatrix} 0 & 0 \\ 0 & \infty \end{bmatrix}
\]

(2.22) \hspace{1cm} (2.23)

These results show that the discrete-time model (2.14), (2.15) will be the source of difficulties as the sampling interval goes to zero. The \( A_d \) matrix becomes the identity matrix, and the noise covariance matrix \( \Sigma_d \) tends to the troublesome values given in (2.23). In the next subsection, it is shown how these difficulties can be readily resolved by appropriate scaling of the model equations.

2.3.2 Incremental Form of the Linear Sampled Data Model

In this section, an alternative formulation of the discrete-time model, which has the same structure as the continuous-time model, but where appropriate scaling is introduced, so that the limit \( \Delta \to 0 \) is meaningful. The alternative model provides conceptual advantages and superior numerical behaviour at fast sampling rates. The material presented in [55, 44, 101] is reviewed below. Then, the extension of these ideas to the nonlinear case is presented.

The problems illustrated in (2.23), (2.22) suggest that the traditional approach to describing discrete-time models is not appropriate when fast sampling rates are employed. In particular, this precludes connecting continuous and discrete results. The remedy is to simply scale the equations and make the sample period explicit, to produce an equivalent incremental model\(^1\) expressed as follows:

\[
dx^+ = x_{k+1} - x_k = A_i x_k \Delta + \omega_k
\]

(2.24)

\(^1\)Sometimes called a delta operator model [101]
\[ dz^+ = z_{k+1} - z_k = \Delta \bar{y}_k = C_i x_k \Delta + \nu_k = y_k \quad (2.25) \]

where it is readily seen using (2.17), (2.18) that
\[
A_i = \frac{A_d - I}{\Delta} = A_c + \frac{1}{2} A_c^2 \Delta + \ldots \quad (2.26)
\]
\[
C_i = C_d = C_c + \ldots \quad (2.27)
\]

The initial state satisfies \( E\{x_0\} = \hat{x}_0 \) and \( E\{(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T\} = \hat{P}_0 \). The new process noise sequence is \( \omega_k = \bar{\omega}_k \), having covariance \( E\{\omega_k \omega_k^T\} = Q_d \). For consistency, with the continuous case, one expresses the covariance in incremental form by explicitly extracting the sample period:

One thus writes
\[
Q_d = Q_i \Delta = \left[ Q_c + \frac{\Delta}{2} (A_c Q_c + Q_c A_c^T) + \ldots \right] \Delta \quad (2.28)
\]

where \( Q_i \) is the incremental covariance.

For the system output equation, one observes that, when one uses the integrating anti-aliasing filter, the expression obtained for the output corresponds to increments of the variable \( z \), i.e.
\[
y_k = \Delta \bar{y}_k = \int_{k\Delta}^{(k+1)\Delta} dz = (z_{k+1} - z_k) \quad (2.29)
\]

The measurement noise sequence is \( \nu_k = \Delta \bar{\nu}_k \). Again, one uses incremental covariance, \( E\{\nu_k \nu_k^T\} = R_i \Delta \), where
\[
R_i \Delta = \Delta^2 R_d = \left[ R_c + \frac{\Delta^2}{3} C_c Q_c C_c^T + \ldots \right] \Delta \quad (2.30)
\]

Finally, the cross-variance \( E\{\omega_k \nu_k^T\} = S_i \Delta = S_d \Delta \).

**Remark 2.3.1** If one defines the discrete Fourier Transform as in [34] via
\[
F(\omega) = \Delta \sum_{k=\infty}^{\infty} f_k e^{-j\omega k \Delta}, \quad (2.31)
\]

Then, the discrete spectral density (i.e. the Fourier Transform of the autocorrelation) of the processes \( \{\omega_k/\Delta\} \) and \( \{\nu_k/\Delta\} \) are \( Q_i \) and \( R_i \) respectively. Again, there is a close connection to the
Remark 2.3.2 If one insists on using the model format given in (2.14), (2.15) then it is seen from (2.19) that one needs to ensure that, the variance of the process noise is scaled, so as to be proportional to $\Delta$. Whereas, the variance of the observation noise should be scaled to be inversely proportional to $\Delta$. This is well known, for further discussion see [19]. On the other hand, the incremental form (2.24) to (2.30) of the discrete-time model recovers the spirit and structure of the continuous time equations, where $Q_i \simeq Q_c$ and $R_i \simeq R_c$. A key point in this context is that, the incremental form is well defined as the sampling period goes to zero, recovering the continuous time equations. In particular, for $\Delta \to 0$ one has from (2.26) to (2.30):

\[
A_i \to A_c \quad (2.32)
\]

\[
C_i \to C_c \quad (2.33)
\]

\[
\begin{bmatrix}
Q_i & S_i \\
S_i^T & R_i
\end{bmatrix} \to \begin{bmatrix}
Q_c & 0 \\
0 & R_c
\end{bmatrix} \quad (2.34)
\]

Of course, the incremental form of the discrete-time model has been simply obtained by appropriate scaling by $\Delta$ and hence conveys the same information as does the traditional discrete-time model. However, the incremental form reinforces the connections between discrete and continuous time [44, 55, 101] and, importantly, allows us to formally take the limit $\Delta \to 0$.

2.3.3 Approximate Sampled Data Model for Linear Case

The incremental model (2.24), (2.25), (2.26), (2.27), (2.28), (2.30) is exact. However, in view of the convergence results noted in (2.32) to (2.34), it is also natural to introduce the following approximate discrete model for the linear case:

\[
d\bar{x}^+ = A_c\bar{x}\Delta + \bar{\omega}_k \quad (2.35)
\]

\[
d\bar{z}^+ = C_c\bar{x}\Delta + \bar{\nu}_k \quad (2.36)
\]
where $\bar{\omega}_k, \bar{\nu}_k$ have covariance $Q_c\Delta$ and $R_c\Delta$ respectively.

It is easily seen from (2.26), (2.27), (2.28), (2.30) that the above model truncates the matrices retaining only the terms which are of order $\Delta$. Thus, the model errors are of order $\Delta^2$.

2.3.4 Exact Nonlinear Sampled Data Models

The nonlinear analogue of the exact linear sampled data models would take the form:

$$dx^+ \triangleq x_{k+1} - x_k = f_i(x_k, \omega_k)\Delta$$

$$dz^+ \triangleq z_{k+1} - z_k = y_k = h_i(x_k, \nu_k)\Delta$$

(2.37) \hspace{1cm} (2.38)

where

$$E\{\omega_k\omega_k^T\} = Q_i\Delta$$

$$E\{\nu_k\nu_k^T\} = R_i\Delta$$

(2.39) \hspace{1cm} (2.40)

A key point is that, unlike the linear case, there is in general no closed form for the discrete nonlinear functions $f_i(\cdot), h_i(\cdot), Q_i, R_i$ as a function of the underlying continuous time system and the sample period $\Delta$. Hence, it is usually necessary, to rely upon approximate sampled data models in the nonlinear case. The simplest such model is described in the next section.

2.3.5 Approximate Sampled Data Nonlinear Model

Paralleling the ideas in Section 2.3.3, the simplest approximate nonlinear sampled data model is obtained by

$$dx^+ \triangleq x_{k+1} - x_k = f_c(x_k)\Delta + \omega_k$$

$$dz^+ \triangleq z_{k+1} - z_k = y_k = h_c(x_k)\Delta + \nu_k$$

(2.41) \hspace{1cm} (2.42)
2.4 Review of Discrete Linear Filtering

where

\[
E\{\omega_k\omega_k^T\} = Q_c \Delta \\
E\{\nu_k\nu_k^T\} = R_c \Delta
\]

(2.43) (2.44)

For further discussion of this approximation see [82].

2.4 Review of Discrete Linear Filtering

The problem of interest is to compute the conditional distribution of the state at time \(k\) given observations of \(\bar{y}_k\) up to, and including, time \(k\) i.e. \(\bar{Y}_k = \{\bar{y}_0, \ldots, \bar{y}_k\}\). The notation \(p_{x_k}(\bar{x}_k|\bar{Y}_k)\) is used to denote this conditional distribution.

2.4.1 The Exact Linear Case

To begin consider the conventional linear discrete model (2.14), (2.15), (2.16), (2.17)-(2.21). In this case, it is well known [6, 7] that the mean \(\hat{x}_{k+1}\) and covariance \(\hat{P}_{k+1}\) of \(p_{x_{k+1}}(\bar{x}_{k+1}|\bar{Y}_k)\) satisfy the traditional Kalman Filter equations:

\[
\hat{x}_{k+1} = A_d \hat{x}_k + K_{d,k}(\bar{y}_k - C_d \hat{x}_k); \quad \hat{x}_0 = \hat{x}_0
\]

(2.45)

where the gain matrix \(K_{d,k}\) is given by,

\[
K_{d,k} = (A_d \hat{P}_k C_d^T + S_d)(R_d + C_d \hat{P}_k C_d^T)^{-1}
\]

(2.46)

and

\[
\hat{P}_{k+1} = A_d \hat{P}_k A_d^T + Q_d - K_{d,k}(C_d \hat{P}_k A_d^T + S_d^T)
\]

(2.47)

where \(\hat{P}_0 = \hat{P}_0\).
2.4.2 Incremental Form of the Exact Sampled Data Linear Filter

Not surprisingly in view of (2.22), (2.23) one cannot take the limit $\Delta \to 0$ in the usual discrete Kalman Filter equation (2.45), (2.46). Specifically, these equations will become ill-conditioned for small $\Delta$. Thus, it is desirable to use a form of the filter analogous to (2.4), (2.5). Of course, for finite $\Delta$ the equations are simply obtained by scaling. However, the incremental form allows one to take the limit $\Delta \to 0$ whereas one cannot do this in the usual form. This suggests one should, rewrite the discrete Kalman filter equations in incremental form as:

$$
\begin{align*}
\dot{\hat{x}}^+ &= A_i \hat{x}_k \Delta + K_k (dz^+ - C_i \hat{x}_k \Delta) \\
\end{align*}
$$

(2.48)

where the matrix gain satisfies,

$$
\begin{align*}
K_k &= \frac{K_{d,k}}{\Delta} \\
&= \frac{[(I + A_i \Delta)P_k C_i^T + S_i][\Delta C_i P_k C_i^T + R_i]}{\Delta}^{-1}
\end{align*}
$$

(2.49)

The conditional state covariance $P_k$ satisfies the incremental form of the Riccati equation namely

$$
\begin{align*}
dP^+ &= P_{k+1} - P_k \\
&= \Delta [PA_i^T + A_i P_k - P_k C_i^T (\Delta C_i P_k C_i^T + R_i)]^{-1} C_i P_k + Q_i] \\
&+ \gamma(\Delta)
\end{align*}
$$

(2.50)

where $\gamma(\Delta)$ is a correction term of order $\Delta^2$, see [101, 55, 119] for details.

**Remark 2.4.1** The filter (2.48) to (2.50) is exact as is (2.45) to (2.47). However, it turns out that, for $\Delta$ small, the form of the discrete filter given in (2.48) to (2.50) has superior numerical properties compared with (2.45) to (2.47), see [129, 101]. This is not surprising in view of the fact
that (2.48) to (2.50) have been scaled so that there exists a well defined limit as $\Delta \to 0$ whereas this is not true for (2.45) to (2.47). Indeed, equations (2.45), (2.47) contain terms which either tend to zero or infinity as $\Delta \to 0$. This observation underlies the source of the numerical issues associated with the traditional discrete filter at fast sample rates.

### 2.4.3 Approximate Sampled Data Linear Filter

In view of the convergence results in (2.32) to (2.34), it is easy to conceive an approximate sampled data linear filter by utilizing the model of (2.35), (2.36) and retaining terms of order $\Delta$ in (2.48) to (2.50). This leads to the following approximate discrete filter:

$$d\hat{x}^+ = A_c\hat{x}\Delta + K_c(dz^+ - C_c\hat{x}\Delta)$$

(2.51)

where

$$K_c = PC_c^T R_c^{-1}$$

(2.52)

and where

$$dP^+ = (PA_c^T + A_c P - PC_c^T R_c^{-1} C_c P + Q_c)\Delta$$

(2.53)

Here (2.32) to (2.34) have been used. Since, in deriving (2.51) to (2.53) one has retained terms of order $\Delta$, then the errors are of order $\Delta^2$.

**Remark 2.4.2** It is not strictly relevant to the current work, but one can formally\(^2\) derive the continuous linear filter corresponding to (2.4), (2.5) by simply taking the limit $\Delta \to 0$ in (2.51) to (2.53). Indeed, in the continuous equations, one simply replaces $\Delta$ by the infinitesimal time increment, $dt$.

### 2.5 Sampled Data Nonlinear Filters

Recall that the problem of interest is to compute $p_{\tilde{x}_k}(\tilde{x}_k|\tilde{Y}_k)$ (The conditional distribution of the state at time $k$ given observations of $\tilde{y}_k$ up to and including time $k$ i.e. $\tilde{Y}_k = \{\tilde{y}_0, \ldots, \tilde{y}_k\}$).\(^3\)

\(^2\)For a rigorous development see, for example, [19].
Here, four cases are considered:

(i) The exact filter when the model is exact.

(ii) An approximate filter when the exact filtering equations are used but the model is approximate.

(iii) An approximate filter when the model is exact but approximate filtering equations are used.

(iv) An approximate filter when both the model and the filtering equations are approximate.

### 2.5.1 Case 1: Exact Filter; Exact Model

Here, it is assumed that the exact model described in Section 2.3.4 is known and is used.

In this case, a recursive set of equations that yield the solution to the filtering problem is described below.

One begins by assuming that \( p_{\bar{x}_0}(\bar{x}_0|\bar{Y}_{-1}) \) is known. For example, one could assume that this distribution is gaussian with mean \( \bar{x}_0 \) and covariance \( \bar{P}_0 \).

Next, it is argued by induction that one has already calculated \( p_{\bar{x}_k}(\bar{x}_k|\bar{Y}_k) \). Then, using the Markovian property (2.37) and marginalizing, one obtains the following form of the state update law:

\[
p_{\bar{x}_{k+1}}(\bar{x}_{k+1}|\bar{Y}_k) = \int p_{\bar{x}_k}(\bar{x}_k|\bar{Y}_k)p_{\bar{x}_{k+1}}(\bar{x}_{k+1}|\bar{x}_k)d\bar{x}_k \quad (2.54)
\]

Then, proceed to consider the impact of adding an observation, i.e. \( \bar{y}_{k+1} \). One has

\[
p_{\bar{x}_{k+1}}(\bar{x}_{k+1}|\bar{Y}_{k+1}) = p_{\bar{x}_{k+1}}(\bar{x}_{k+1}|\bar{Y}_k, \bar{y}_{k+1}) = \frac{p_{\bar{x}_{k+1}}(\bar{x}_{k+1}|\bar{Y}_{k})p_{\bar{y}_{k+1}}(\bar{y}_{k+1}|\bar{x}_{k+1})}{\int p_{\bar{x}_{k+1}}(\bar{x}_{k+1}|\bar{Y}_k)p_{\bar{y}_{k+1}}(\bar{y}_{k+1}|\bar{x}_{k+1})d\bar{x}_{k+1}} \quad (2.55)
\]

Equations (2.54), (2.55) are often commonly referred to as the Chapman Kolmogorov equation and Bayes’ rule respectively. Note that (2.55) is equivalent to scaling by a likelihood function followed by normalization.
Note that both equations depend upon integration of functions in high dimensional spaces. Thus, it is rarely possible in practice to implement (2.54), (2.55) unless some form of quantization is used. Methods used to solve this issue include the, so-called, Monte-Carlo based methods, also known as Particle Filters, see [39, 81, 34]. Other methods based on vector quantization are described latter in this thesis.

2.5.2 Case 2: Exact Filter; Approximate Model

The approximate filter in this case is obtained by utilizing (2.54), (2.55) but replacing the exact model by the approximations given in (2.42) to (2.44).

Remark 2.5.1 In Chapter 3 upsampling and downsampling will be considered. There, the approximate filter considered in case 2 is used. In particular, the fact that the change in conditional probability is of order \( \Delta \) and the fact that the error, due to the model approximation, is order \( \Delta^2 \) are used.

2.5.3 Case 3: Approximate Filter, Exact Model

Here, one returns to the exact model of section 2.3.4 but approximates the filter equations by ignoring terms of order \( \Delta^2 \) and higher. The formal derivation of the approximate filter follows standard arguments, using Taylor’s series expansions, see for example [85].

A key observation (that it is exploited in the sequel) is that one needs to evaluate the increment in the conditional probability. This is analogous to the idea of incremental models in Section 2.1.1 and 2.1.2.

Retaining terms of order \( \Delta \), equation (2.54) can be approximated as follows:

\[
\begin{align*}
p_{\bar{x}_{k+1}}(\alpha|\bar{Y}_k) - p_{\bar{x}_k}(\alpha|\bar{Y}_k) &= -\sum_{j=1}^{n} \frac{\partial}{\partial j} [f_i(\alpha)p_{\bar{x}_k}(\alpha|\bar{Y}_k)] \Delta \\
&+ 1/2 \sum_{i,j=1}^{n} Q_{i,j} \frac{\partial^2 p_{\bar{x}_k}(\alpha|\bar{Y}_k)}{\partial \alpha_i \partial \alpha_j} \Delta + O(\Delta^2)
\end{align*}
\]  

(2.56)
The above equation is a sampled data version of the Kolmogorov forward diffusion equation.

Similarly equation (2.55) can be approximated as follows:

\[
p_{\bar{x}_{k+1}}(\alpha | \bar{Y}_{k+1}) = p_{\bar{x}_{k+1}}(\alpha | \bar{Y}_{k}) \\
= p_{\bar{x}_{k+1}}(\alpha | \bar{Y}_{k}) \left\{(dz^+ - E\{h_i\}\Delta)^T \right. \\
\left. \cdot R^{-1}(h_i - E\{h_i\})\bigg\} + \mathcal{O}(\Delta^2) \right. \tag{2.57}
\]

where

\[
E\{h\} \triangleq \int h(\alpha)p_{\bar{x}_{k+1}}(\alpha | \bar{Y}_{k})d\alpha \tag{2.58}
\]

The above equation is a discrete form of the Kushner Stratonovich equation [84].

2.5.4 Case 4: Approximate Model; Approximate Filter

Here, the “true model” of section 2.3.4 is replaced using the approximations given in (2.41) to (2.44). Thus, one replaces \(f_i, h_i, Q_i, R_i\) in the approximate filter of equations (2.56) to (2.58) by \(f_c, h_c, Q_c, R_c\) respectively. Note that, in this case, both the model and the filter are approximations with error \(\mathcal{O}(\Delta^2)\).

**Remark 2.5.2** If one specializes to the linear gaussian case, then one obtains the approximations given earlier in (2.51) to (2.53) for the mean and covariance of the posterior distribution.

**Remark 2.5.3** As in section 2.4.3 for the linear case, one can also formally derive the continuous nonlinear filter by letting \(\Delta \to 0\) in the above equations (see [85]).

2.6 Conclusions

This chapter has outlined the structure necessary to discuss the nonlinear filtering problem. The linear case has been reviewed starting from continuous time systems and introducing sampling techniques and sampled data models. The connections between continuous and discrete time models and filters have been established in the linear case. The linear results have been mirrored in the nonlinear case as best as possible and associated difficulties have been made explicit.
In the next chapter, the issue of temporal sampling in nonlinear filtering is discussed.
2. OVERVIEW OF NONLINEAR FILTERING
TEMPORAL SAMPLING AND MODELLING ISSUES

In practice, all filters and control systems are implemented using some form of digital tool. For instance, computers, microprocessors and communication channels are commonly found in every implementation. They all use some form of sampling strategy to facilitate interacting with the underlying analogue signals found in all systems.

The sampling components impose technological constraints in every system. For example, the sampling period that can be achieved is typically bounded from below and therefore, the sampled data model has a minimum sampling period. In the nonlinear case, this introduces uncertainty in the models. Consequently, this will have an impact on the solution of the filtering problem.

Sometimes, achievable sampling rates can be either too fast or not fast enough for a given problem. In this situation, it is not clear how best to proceed. This motivates the temporal sampling issues studied in this chapter.

The remainder of the chapter is organized as follows. Section 3.1 addresses the case when the available sampling rate is unnecessarily fast. Then, Section 3.2 studies the opposite case when the sampling rate is excessively slow and the resultant errors in the nonlinear model are problematic. Finally, Section 3.3 analyses the case when the sampling is not uniform, but is triggered by events.
3.1 Down-sampling

Say that one is satisfied that a sampling period $\Delta_{\text{min}}$ would give sufficiently small errors when used in the first form of the approximate nonlinear filter as in Section 2.5.2. Also, say that the technological constraints allow a sampling period $\Delta \ll \Delta_{\text{min}}$. Of course, one could run the filter at period $\Delta$ and achieve accurate results. However, one may suspect that this would be wasteful of computational effort since one has already decided that $\Delta_{\text{min}}$ is adequate. Indeed, it is intuitively clear that not much of interest will happen on a sample by sample basis when $\Delta$ is very small (i.e. much smaller than $\Delta_{\text{min}}$). This suggests that some form of down-sampling may be helpful. However, this must be done carefully. For example, the correct strategy is definitely not to simply take every $m^{th}$ sample and throw the rest away! Clearly, this would lead to a highly suboptimal filter since most of the data will have been discarded. On the contrary, if one decides to increase the sampling period $\Delta$ from $\Delta$ to $m\Delta$, then a new anti-aliasing filter relevant to the new sample period $m\Delta$ is required. For example, say that one uses the usual integrating filter, then a new observation sequence $^2 d z'_l$

$$d z'_l = \sum_{k=1}^{m} d z_{m(l-1)+k}$$

(3.1)

is required. One can then run the filter of Section 2.5.2 at period $\Delta' = m\Delta$ with observations $d z'$. If implemented properly, the step of down-sampling can lead to major computational improvements without significant degradation in performance. Indeed, the example below, shows that, in this illustrative case, one can down-sample by several orders of magnitude with a corresponding reduction of several orders of magnitude in the computational effort with insignificant change in the computed conditional probability.

---

1We assume that $m\Delta \leq \Delta_{\text{min}}$ as defined in Assumption 1.

2Note that this step of using a sum (or equivalently average) of the fast samples is crucial.
3.1 Down-sampling

3.1.1 Example

Consider the following simple discrete nonlinear system:

\[ x_{k+1} = ax_k + \omega'_k \]  \hspace{1cm} (3.2)
\[ y_k = x_k^2 + \nu'_k \]  \hspace{1cm} (3.3)

where \( a = 0.999 \); \( E(\omega'_k^2) = 10^{-2} \); \( E(\nu'_k^2) = 10^4 \), \( \Delta = 10^{-3} \). The magnitude of \( E((\omega'_k)^2) \) and \( E((\nu'_k)^2) \) may seem counterintuitive but these scalings are well motivated as described in subsection 2.3.1. The system (3.2), (3.3) is more intuitive when expressed in the equivalent incremental form:

\[ dx^+ = f_i(x_k)\Delta + \omega_k \]  \hspace{1cm} (3.4)
\[ dz = h_i(x_k)\Delta + \nu_k \]  \hspace{1cm} (3.5)

where \( f_i(x_k) = -x_k \); \( h_i(x_k) = x_k^2 \) and \( \omega_k, \nu_k \) both have incremental covariance of \( 10\Delta \).

Note that the observations process \( \{y_k\} \) (see (3.3)) does not tell us anything about the sign of \( x \) and hence the posterior distribution will be bimodal (See Figure 3.1 presented later).

It seems intuitively clear that the sample period of \( 10^{-3} \) will lead to wasted computational effort. This suggests that one could down-sample the system by using the strategy explained in (3.1). Figures 3.1, 3.2, 3.3 show the evolution of \( p_{x_k}(x_k|Z_k) \) (evaluated via (2.54), (2.55)) for \( \Delta = 10^{-3} \) and the down-sampled versions for \( m\Delta = 10^{-2} \) and \( m\Delta = 10^{-1} \) respectively. Inspection of the plots indicates that, there is no noticeable deterioration in the computed posterior probability.

However, an important point is that at \( \Delta = 10^{-1} \), the total computational load associated with the nonlinear filter has been reduced by two orders of magnitude relative to the use of \( \Delta = 10^{-3} \)!
3. Temporal Sampling and Modelling Issues

Figure 3.1: Time evolution of the probability density function at fast sampling, $\Delta = 0.001$

Figure 3.2: Time evolution of the probability density function at fast sampling, $\Delta = 0.01$
3.2 Up-sampling

3.2.1 The nonlinear case

Next, consider the opposite situation to that discussed in section 3.1 when slow sampling, i.e. when $\Delta > \Delta_{\text{min}}$, is used. The core problem in this case is that, the approximate model resulting from using the correspondences given in (2.41) to (2.44), may be too inaccurate to give meaningful results. The fault lies with the use of the continuous functions $f_c, h_c, Q_c, R_c$, in the sampled data model at period $\Delta$.

The key tool to obtain a more accurate model is to use **up-sampling**, i.e. to run the state update part of the Nonlinear Filter at a submultiple of the given sample period. That is, choose $\Delta' = \Delta/N$ (where $\Delta/N \leq \Delta_{\text{min}}$) and run the state updates at period $\Delta'$. One can always choose $N$ sufficiently large so that $\Delta'$ is sufficiently small.

In more detail, one first takes samples $y_k = z((k + 1)\Delta) - z(k\Delta)$ and define the information random vector $\mathbf{Y}_k^\Delta := (y_1, \ldots, y_k)$, and $\mathbf{x}_k^\Delta := x(k\Delta)$. The goal is to find the conditional

\[ \mathbb{P}(\mathbf{Y}_k^\Delta | \mathbf{X}_{k-1}^\Delta) \]

\[ \mathbb{P}(\mathbf{x}_k^\Delta | \mathbf{Y}_{k-1}^\Delta) \]

\[ \mathbb{P}(\mathbf{x}_k^\Delta | \mathbf{Y}_k^\Delta) \]
density \( x \mapsto p_{k+1}^{\Delta}(x, \hat{Y}_k^{\Delta}) \) for some realization \( \hat{Y}_k^{\Delta} = (\hat{y}_1, \ldots, \hat{y}_k) \in (\mathbb{R}^p)^k \).

As discussed above, the system is up-sampled by introducing \( \Delta' = \Delta/N \). Then, one uses the discrete state space model with period \( \Delta' \). This means that one needs to interlace \( N \) versions of the state update between every observation update.

However, there is an important issue with respect to the observation update. Recall that the continuous signals must be passed through an anti-aliasing filter. Hence, extra state variables must be introduced in the model to describe the effect of the anti-aliasing filter. (In the simplest case, this takes the form of an averaging filter or integrator). Thus, it is necessary to introduce further state variables to describe the low pass filtering action inherent in the calculation of the average \( y_k = \int_{k\Delta}^{(k+1)\Delta} dz \).

The up-sampled model becomes

\[
\begin{align*}
x_{k+1+LN}^u &= x_{k+LN}^u + \Delta'f(x_{k+LN}^u) + w_{k+LN}^u \\
h_{k+1+LN}^u &= h_{k+LN}^u + \Delta'h(x_{k+LN}^u) + v_{k+LN}^u \\
y_{L+1}^{(N)} &= h_{(L+1)N}^u - h_{LN}^u
\end{align*}
\]

with \( 0 \leq k < N \), \( L \in \mathbb{N}_0 \), initial state \( (x_0^u, h_0^u) = (x_0, 0) \) and Gaussian white noise \( \begin{pmatrix} w_k^u \\ v_k^u \end{pmatrix} \) with covariance \( \begin{pmatrix} \Delta'Q & 0 \\ 0 & \Delta'R \end{pmatrix} \). Furthermore, \( h \) is an extra state to track the effect of the anti-aliasing filter.

Based on the above up-sampled model, one can easily derive a discrete filter for the conditional density \( \pi \mapsto p_{k+1}^{(N)}(\pi, \hat{Y}_k^{(N)}) \) of \( \pi = \begin{pmatrix} x_{(k+1)N}^u \\ h_{(k+1)N}^u \end{pmatrix} \) given the realization \( \hat{Y}_k^{(N)} = (\hat{y}_1^{(N)}, \ldots, \hat{y}_{k+1}^{(N)}) \) of \( Y_k^{(N)} = (y_1^{(N)}, \ldots, y_{k+1}^{(N)}) \). We also can marginalize this density to obtain the conditional density \( x \mapsto p_{k}^{(N)}(x, \hat{Y}_k^{(N)}) \) of \( x_{(k+1)N}^u \) given \( Y_k^{(N)} = \hat{Y}_k^{(N)} \).

This idea is illustrated below using the same example as earlier.

\(^4\)Here the superscript \( u \) denotes up-sampled variables and \( (N) \) denotes the variables with sampling period \( \Delta'N \) but obtained with the model at sampling rate \( \Delta' \).
3.2 Up-sampling

Example

Consider a continuous-time system defined by (3.2), (3.3). The sample period is given as $\Delta = 1$ which is assumed greater than $\Delta_{\text{min}}$. After testing several up-sample factors, namely $N = \{2, 4, 8\}$. The corresponding posterior distributions at $t = 8$ are shown in Figure 3.4.

As it is seen from the figure, the discrete model at period $\Delta$ gives a poor result which differs from the results obtained for all up-sampled models. Also note that, as one increases the up-sample factor $N$, the posterior probabilities converge.

![Figure 3.4: Probability density functions at $t = 8$](image)
3. Temporal Sampling and Modelling Issues

3.2.2 The Linear Case

To further support the idea of upsampling, a complete analysis for the corresponding linear problem is presented below. Consider the linear stochastic model \((2.4), (2.5)\) with matrices \(A, C \in \mathbb{R}^{n \times n}\) and \(B, D \in \mathbb{R}^{p \times n}\).

As mentioned above one chooses some sample period \(\Delta > 0\) and take samples \(y_k = z((k + 1)\Delta) - z(k\Delta)\) for \(k \geq 0\). Define \(x_{k+1}^\Delta = x_{k\Delta} = x(k\Delta)\), one can obtain the exact discrete model, see Section 2.3,

\[
x_{k+1}^\Delta = A x_k^\Delta + w_k^\Delta \\
y_k = C x_k^\Delta + v_k^\Delta,
\]

where \(A^\Delta = e^{A\Delta}, C^\Delta = C \int_0^\Delta e^{\tau A} d\tau\), and \((w_k^\Delta, v_k^\Delta)\) is Gaussian white noise with covariance

\[
\begin{pmatrix}
Q^\Delta & S^\Delta \\
S^\Delta & R^\Delta
\end{pmatrix} = \int_0^\Delta e^{\tau A} \begin{pmatrix}
Q & 0 \\
0 & R
\end{pmatrix} e^{\tau A}^T d\tau,
\]

where \(A = \begin{pmatrix} A & 0 \\ C & 0 \end{pmatrix}\).

Given the information vector \(\hat{Y}_k^\Delta\) at time \((k)\Delta\) the discrete system gives the exact solution of the continuous system at times \(k\Delta\). The density \(p_k^\Delta(\cdot, \hat{Y}_k^\Delta)\) is Gaussian and is thus characterized by its mean \(\mu_k^\Delta\) and covariance \(\Sigma_{k|k}^\Delta\). These quantities satisfy the usual Kalman Filter:

\[
\mu_k^\Delta = A \mu_{k-1}^\Delta + K_k^\Delta(\Delta \hat{y}_k - C \mu_k^\Delta)
\]

where the Kalman gain matrix \(K_k^\Delta\) satisfies the equations

\[
K_k^\Delta = (A \Sigma_{k|k-1}^\Delta C^T + S^\Delta)(C \Sigma_{k|k-1}^\Delta C^T + R^\Delta)^{-1}
\]

\[
\Sigma_{k+1|k}^\Delta = A \Sigma_{k|k-1}^\Delta A^T + Q^\Delta - K_k^\Delta(C \Sigma_{k|k-1}^\Delta C^T + R^\Delta)K_k^\Delta^T.
\]

\(^5\)Input from Dr. Claus Müller is acknowledged in producing the results presented here. See list of authors publications for details of percentages of contributions.
Here $\mu_0 = \mu$ and $\Sigma_0 := \Sigma_0$. Moreover $\Sigma_{k+1|k} = \text{cov}(x_{k+1|k}^\Delta_k = y)$ is independent of $y \in (\mathbb{R}^p)^{k+1}$.

In the spirit of the non-linear case discussed in Section 3.2.1, also define the up-sampled system as follows:

\[
\begin{align*}
\tilde{x}_{k+1}^{(N)} &= A_N x_k^{(N)} + w_k^{(N)} \quad (3.11) \\
\tilde{y}_{k}^{(N)} &= H_{N,N} x_k^{(N)} + v_k^{(N)}, \quad (3.12)
\end{align*}
\]

where $A_N = I_n + \Delta'A$ and $H_{l,N} = \Delta'C \sum_{k=0}^{l-1} A_N^k$, and $I_n$ is the $n \times n$ identity matrix.

Moreover,

\[
\begin{pmatrix}
\tilde{w}_k^{(N)} \\
\tilde{v}_k^{(N)}
\end{pmatrix} = \sum_{l=1}^N \begin{pmatrix}
A_{N-1}^l & 0 \\
H_{l-1,N} & I_p
\end{pmatrix} \begin{pmatrix}
w_{(k+1)N-l}^{(N)} \\
v_{(k+1)N-l}^{(N)}
\end{pmatrix} \quad (3.13)
\]

and

\[
\begin{pmatrix}
A_N^l \\
H_{l,N} \\
I_p
\end{pmatrix} = (I_{n+p} + \Delta'A)^l,
\]

thus

\[
\text{cov} \begin{pmatrix}
\tilde{w}_k^{(N)} \\
\tilde{v}_k^{(N)}
\end{pmatrix} := \begin{pmatrix}
Q^{(N)} & S^{(N)} \\
S^{(N),T} & R^{(N)}
\end{pmatrix} = \Delta' \sum_{l=0}^{N-1} f_l(A) \begin{pmatrix}
Q & 0 \\
0 & R
\end{pmatrix} f_l(A^T), \quad (3.14)
\]

where $f_l(z) = (1 + \Delta'z)^l$.

The density $p_{k}^{(N)}(\cdot, \tilde{y}^{(N)}_k)$ is Gaussian, and its mean $\mu_k^{(N)}$ and covariance $\Sigma_{k+1|k}^{(N)}$ satisfy the up-sampled Kalman filter

\[
\mu_{k+1}^{(N)} = A_N \mu_k^{(N)} + K_{k}^{(N)} \left( \tilde{y}_{k+1}^{(N)} - H_{N,N} \mu_k^{(N)} \right), \quad (3.15)
\]

where $K_{k}^{(N)} = \Sigma_{k+1|k}^{(N)} H_{N,N}^T (H_{N,N} \Sigma_{k+1|k}^{(N)} H_{N,N}^T + R^{(N)})^{-1}$.
where the Kalman gain matrix $K_k^{(N)}$ satisfies the equations

$$
K_k^{(N)} = \left( A_N^{(N)} \Sigma_{k|k-1}^{(N)} H_{N,N}^T + S^{(N)} \right) \left( H_{N,N} \Sigma_{k|k-1}^{(N)} H_{N,N}^T + R^{(N)} \right)^{-1} 
$$

$$
\Sigma_{k+1|k}^{(N)} = A_N^{(N)} \Sigma_{k|k-1}^{(N)} A_N^{(N)^T} + Q^{(N)} \ldots
$$

$$
\ldots - K_k^{(N)} \left( H_{N,N} \Sigma_{k|k-1}^{(N)} H_{N,N}^T + R^{(N)} \right) K_k^{(N)^T}.
$$

(3.16)

Here $\mu_0^{(N)} = \mu$ and $\Sigma_{0|1}^{(N)} = \Sigma_0$. Moreover, $\Sigma_{k+1|k}^{(N)} = \text{cov}(x_{k+1}^{(N)} | y_k^{(N)} = y)$, is independent of $y \in (\mathbb{R}^p)^{k+1}$.

One can quantify the convergence of $p_k^{(N)}(\cdot, \hat{y}_k^{\Delta})$ to $p_k^{\Delta}(\cdot, \hat{y}_k^{\Delta})$ in the sense made precise in the following theorem.

**Theorem 3.2.1** For all $k \geq 0$ and all measurements $\hat{y}_k^{\Delta}$ the sequence of densities $p_k^{(N)}(\cdot, \hat{y}_k^{\Delta})$ converges uniformly and in $L_1$ to the density $p_k^{\Delta}(\cdot, \hat{y}_k^{\Delta})$, with upper error rate in both cases of at most $O(1/\sqrt{N})$.

**Proof.** We first study convergence of the various quantities appearing in the up-sampled model.

We begin by examining the convergence of \text{cov} \left( \begin{bmatrix} \overline{v}_k^{(N)} \\ \overline{w}_k^{(N)} \end{bmatrix} \right). We recall that if the function $r$ is entire and $A$ is some $n \times n$ matrix then

$$
r(A) = \frac{1}{2\pi i} \oint_K r(z)(zI_n - A)^{-1} dz,
$$

(3.17)

where $K$ is a positive orientated circle with radius greater than $\|A\|$, the matrix norm induced by the Euclidean norm. This formula allows one to deduce matrix convergence from convergence of corresponding functions. Thus one can establish:

**Lemma 3.2.1** If $z, w \in \mathbb{C}$ such that $|z| = 2|w| > 0$ and $N \geq 4\Delta|w|$ then

$$
\left| \int_0^\Delta e^{t(z+w)} \, dt - \frac{\Delta}{N} \sum_{i=0}^{N-1} f_i(z) f_i(w) \right| \leq c_1/N,
$$

(3.18)

where $c_1$ only depends on $\Delta$ and $|w|$.
3.2 Up-sampling

**Proof.** The assumptions on $z, w$ and $N$ show that $z + w \neq 0$ and $z + w + \Delta z w / N \neq 0$, thus

\[
\int_0^\Delta e^{t(z+w)} \, dt - \frac{\Delta}{N} \sum_{l=0}^{N-1} f_l(z) f_l(w) = e^{\Delta(z+w)} - 1 - \frac{(1 + \Delta z / N)^N (1 + \Delta w / N)^N - 1}{z + w + \Delta z w / N},
\]

(3.19)

\[
= e^{\Delta(z+w)} - (1 + \Delta z / N)^N (1 + \Delta w / N)^N \left( \frac{z + w}{z + w + \Delta z w / N} \right) + \frac{(1 + \Delta z / N)^N (1 + \Delta w / N)^N - 1}{z + w + \Delta z w / N}.
\]

(3.20)

Recall the mean value theorem for some analytic function $(u, v) \mapsto f(u, v)$ in two complex variables, which states that

\[
|f(u + z, v + w) - f(u, v)|^2 \leq (|z|^2 + |w|^2) \max_{t \in [0, 1]} \left| \frac{\partial f}{\partial u}(u + tz, v + tw) \right|^2 + \left| \frac{\partial f}{\partial v}(u + tz, v + tw) \right|^2,
\]

(3.21)

One then obtains

\[
|\frac{1}{1 + z/N} \left( \frac{1 + w}{1 + w/N} \right)^N e^{-z+w} - 1| \leq \frac{|z|^2 + |w|^2}{N^2} e^{2(|z| + |w| - \text{Re}(z+w))} \left( \left( \frac{|z|}{1 + |z|/N} \right)^2 + \left( \frac{|w|}{1 + |w|/N} \right)^2 \right),
\]

(3.22)

thus

\[
|\frac{1}{1 + z/N} \left( \frac{1 + w}{1 + w/N} \right)^N - e^{z+w}| \leq \frac{|z|^2 + |w|^2}{N} e^{|z| + |w|}.
\]

(3.23)

Moreover

\[
\left| \frac{1}{z} + \frac{1}{w} + \frac{\Delta}{N} \right| \geq \frac{1}{|w|} - \frac{1}{|z|} - \frac{\Delta}{N} \geq \frac{1}{2|w|} - \frac{1}{4|w|} \geq \frac{1}{4|w|}.
\]

(3.24)
and thus
\[
\frac{N}{\Delta} \left| 1 - \frac{z + w}{z + w + \Delta z w / N} \right| = \left| \frac{zw}{z + w + \Delta z w / N} \right| = \frac{1}{\frac{1}{w} \frac{1}{z} + \frac{\Delta}{N}} \leq 4|w|. \tag{3.25}
\]

This shows that
\[
\left| \frac{(1 + \Delta z / N)^N (1 + \Delta w / N)^N - 1}{z + w} \right| - \left| \frac{(1 + \Delta z / N)^N (1 + \Delta w / N)^N - 1}{z + w + \Delta z w / N} \right| = \frac{(1 + \Delta z / N)^N (1 + \Delta w / N)^N - 1}{z + w} \left| 1 - \frac{z + w}{z + w + \Delta z w / N} \right| \leq \frac{1 + e^{\Delta (|z| + |w|)}}{|z + w|} \cdot \frac{4\Delta|w|}{N} \leq 4\Delta \frac{1}{N} \left( 1 + e^{\Delta (|z| + |w|)} \right). \tag{3.26}
\]

In summary,
\[
\left| \int_0^\Delta e^{t(z+w)} \, dt - \frac{\Delta}{N} \sum_{l=0}^{N-1} f_l(z) f_l(w) \right| \leq \frac{\Delta^2}{N|z + w|} e^{\Delta (|z| + |w|)} + \frac{4\Delta}{N} \left( 1 + e^{\Delta (|z| + |w|)} \right), \tag{3.27}
\]

which yields the desired inequality after we let $|z| = 2|w|$. ■

As a consequence of (3.17) and Lemma 3.2.1 we have:

**Lemma 3.2.2** If $N \geq 4\Delta (1 + \|A\|)$, then
\[
\left\| \begin{pmatrix} Q_{\Delta} & S_{\Delta} \\ S_{\Delta}^T & R_{\Delta} \end{pmatrix} - \begin{pmatrix} Q^{(N)} & S^{(N)} \\ S^{(N),T} & R^{(N)} \end{pmatrix} \right\| = O(1/N). \tag{3.29}
\]

**Proof.** If $K_\varepsilon$ is the positively orientated circle centered at 0 with radius $\tau = \varepsilon + \|A\|$ we have,
if $\varepsilon, \varphi > 0$,

$\begin{pmatrix} Q_{\Delta} & S_{\Delta} \\ S_{\Delta}^T & R_{\Delta} \end{pmatrix} = \int_{\mathbb{R}} e^{\mathbb{A}} \begin{pmatrix} Q & 0 \\ 0 & R \end{pmatrix} e^{\mathbb{A}^T} dt$

$= \frac{1}{(2\pi)^2} \int_{\mathbb{R}_{\varepsilon}} \int_{\mathbb{R}_{\varphi}} e^{tz} (zI_{n+p} - \mathbb{A})^{-1} dz \begin{pmatrix} Q & 0 \\ 0 & R \end{pmatrix}$

$\int_{\mathbb{R}_{\varphi}} e^{tw} (wI_{n+p} - \mathbb{A}^T)^{-1} dw dt$ \hspace{1cm} (3.30)

and

$\begin{pmatrix} Q^{(N)} & S^{(N)} \\ S^{(N),T} & R^{(N)} \end{pmatrix} = \frac{\Delta}{N} \sum_{l=0}^{N-1} f_l(\mathbb{A}) \begin{pmatrix} Q & 0 \\ 0 & R \end{pmatrix} f_l(\mathbb{A}^T)$

$= \frac{\Delta}{N(2\pi)^2} \sum_{l=0}^{N-1} \int_{\mathbb{R}_{\varepsilon}} f_l(z) (zI_{n+p} - \mathbb{A})^{-1} dz \begin{pmatrix} Q & 0 \\ 0 & R \end{pmatrix}$

$\int_{\mathbb{R}_{\varphi}} f_l(w) (wI_{n+p} - \mathbb{A}^T)^{-1} dw,$ \hspace{1cm} (3.31)

thus

$\left\| \begin{pmatrix} Q_{\Delta} & S_{\Delta} \\ S_{\Delta}^T & R_{\Delta} \end{pmatrix} - \begin{pmatrix} Q^{(N)} & S^{(N)} \\ S^{(N),T} & R^{(N)} \end{pmatrix} \right\| \leq \frac{1}{4\pi^2} \left\| \int_{\mathbb{R}_{\varepsilon}} \int_{\mathbb{R}_{\varphi}} \left( \int_{0}^{\Delta} e^{t(z+w)} dt - \frac{\Delta}{N} \sum_{l=0}^{N-1} f_l(z) f_l(w) \right) (zI_{n+p} - \mathbb{A})^{-1} (Q \begin{pmatrix} 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ R \end{pmatrix} wI_{n+p} - \mathbb{A}^T)^{-1} dw dz \right\|$ \hspace{1cm} (3.32)

$\leq \frac{\tau_{\varepsilon} \tau_{\varphi}}{\varepsilon \varphi} \left\| \begin{pmatrix} Q & 0 \\ 0 & R \end{pmatrix} \right\| \max_{|z|=r_{\varepsilon}, |w|=r_{\varphi}} \left| \int_{0}^{\Delta} e^{t(z+w)} dt - \frac{\Delta}{N} \sum_{l=0}^{N-1} f_l(z) f_l(w) \right|.$ \hspace{1cm} (3.33)

The assertion follows by Lemma 3.2.7 if we let $\tau_{\varepsilon} = 2\tau_{\varphi}$ and $N \geq 4\Delta \tau_{\varphi}$. \hfill \blacksquare
The same strategy allows one to establish convergence of $A_N^N$ and $H_{N,N}$. First establish the following result regarding convergence of functions:

**Lemma 3.2.3** For all $z \in \mathbb{C}$ and all $N \in \mathbb{N}^+$ we have

$$|(1 + z/N)^N - e^z| \leq \frac{|z|^2 e^{|z|}}{N + |z|},$$  \hspace{1em} (3.34)

and

$$|(1 + z/N)^N - 1| \leq \frac{|z| e^{|z|}}{1 + |z|/N},$$  \hspace{1em} (3.35)

**Proof.** If $f$ is entire one has the known estimate

$$|f(z + w) - f(z)| \leq |w| \max_{0 \leq t \leq 1} |f'(z + tw)|$$  \hspace{1em} (3.36)

for all $z, w \in \mathbb{C}$. Letting $f(z) = (1 + z)^N e^{-Nz}$ one obtains

$$|(1 + z/N)^N e^{-z} - 1| = |f(z/N) - f(0)|$$  \hspace{1em} (3.37)

$$\leq \frac{|z|}{N} \max_{0 \leq t \leq 1} |tze^{-t(1 + (t^1)/N)}|$$

$$\leq \frac{|z|^2}{N} \max_{0 \leq t \leq 1} te^{-tRez} \frac{e^{|z|}}{1 + (t|z|)/N},$$  \hspace{1em} (3.38)

since $(1 + x/N)^N \leq e^x$ for all $x \geq 0$ and all $N \in \mathbb{N}^+$. Thus

$$|(1 + z/N)^N - e^z| \leq |z|^2 \max_{0 \leq t \leq 1} \frac{te^{(1-t)Rez} e^{|z|}}{N + t|z|}$$  \hspace{1em} (3.39)

$$= \frac{|z|^2 e^{|z|}}{N + |z|},$$  \hspace{1em} (3.40)

which shows the first assertion. If we let $f(z) = (1 + z)^N$ and estimate $|f(z/N) - f(0)|$ with the same technique one obtains the second one.

Again as a consequence of (3.17) and Lemma 3.2.3 it follows:

**Lemma 3.2.4** 1)

$$\|A_\Delta - A_N^N\| = O(1/N).$$  \hspace{1em} (3.41)
3.2 Up-sampling

2)

\[ \| C_\Delta - H_{N,N} \| = \mathcal{O}(1/N). \] (3.42)

**Proof.** 1) Define \( f(z) = e^{\Delta z} - (1 + \Delta z/N)^N \) and obtain, if \( r_\varepsilon = \varepsilon + \| A \| \) and \( K_\varepsilon \) is the circle centered at 0 with radius \( r_\varepsilon \).

\[
\| A_\Delta - A^N_N \| = \left\| \frac{1}{2\pi i} \oint_{K_\varepsilon} f(z)(zI_n - A)^{-1} dz \right\| 
\leq \frac{r_\varepsilon}{\varepsilon} \max_{|z|=r_\varepsilon} |f(z)|
\leq \frac{r_\varepsilon}{\varepsilon} \frac{\Delta^2 r_\varepsilon^2 e^{\Delta r_\varepsilon}}{N + \Delta r_\varepsilon}
\] (3.43)

by Lemma 3.2.3

2) Define

\[
f(z) = \int_{0}^{\Delta} e^{\tau z} \, d\tau - \frac{\Delta}{N} \sum_{k=0}^{N-1} (1 + \Delta z/N)^k
= \frac{e^{\Delta z} - (1 + \Delta z/N)^N}{z}
\] (3.45)

and apply the same technique as above.

Next the convergence of covariances and Kalman gains are studied:

**Lemma 3.2.5** For all \( k \in \mathbb{N}_0 \)

1)

\[
\| \Sigma_{k|k-1}^{(N)} - \Sigma_{k|k-1}^\Delta \| = \mathcal{O}(1/N). \] (3.46)

2)

\[
\| K_k^{(N)} - K_k^\Delta \| = \mathcal{O}(1/N). \] (3.47)
3. Temporal Sampling and Modelling Issues

Proof. 1) From

\[
\Sigma_{k+1|k}^{(N)} = A_N^N \Sigma_{k|k-1}^{(N)} A_N^{N,T} + Q^{(N)} - (A_N^N \Sigma_{k|k-1}^{(N)} H_{N,N}^T + S^{(N)})
\cdot (H_{N,N} \Sigma_{k|k-1}^{(N)} H_{N,N}^T + R^{(N)})^{-1} (H_{N,N} \Sigma_{k|k-1}^{(N)} A_N^{N,T} + S^{(N),T})
\]  (3.48)

it follows by induction, using Lemma 3.2.2 and Lemma 3.2.4, that there are positive definite matrices \(\Lambda_k\) such that

\[
\|\Sigma_{k|k-1}^{(N)} - \Lambda_k\| = O(1/N)\text{ for all } k,
\]

and

\[
\Lambda_{k+1} = A_\Delta \Lambda_k A_\Delta^T + Q_\Delta - (A_\Delta \Lambda_k C_\Delta^T + S_\Delta)
\cdot (C_\Delta \Lambda_k C_\Delta^T + R_\Delta)^{-1} (C_\Delta \Lambda_k A_\Delta^T + S_\Delta^T).
\]  (3.49)

This shows that \(\Lambda_k = \Sigma_{k|k-1}^{\Delta}\) for all \(k\). Now 2) follows straightforwardly.  

Now one investigates the question if, given the measurement \(\hat{Y}_k^\Delta = (\Delta \hat{z}_1, \ldots, \Delta \hat{z}_{k+1}) \in (\mathbb{R}^p)^{k+1}\) of the 'true' system, the density \(p_{k+1}^{(N)}(\cdot; \hat{Y}_k^\Delta)\) of \(x_{k+1}^{(N)}\) given \(Y_k^{(N)} = \hat{Y}_k^\Delta\) converges to the density \(p_{k+1}^{\Delta}(\cdot; \hat{Y}_k^\Delta)\)of \(x_{k+1}^{\Delta}\) given \(Y_k^{\Delta} = \hat{Y}_k^\Delta\).

Both these random vectors are Gaussian with covariance \(\Sigma_{k+1|k}^{(N)}\) and \(\Sigma_{k+1|k}^{\Delta}\), respectively. Contrary to their means \(\mu_{k+1}^{(N)} := E(x_{k+1}^{(N)}|Y_k^{(N)} = \hat{Y}_k^\Delta)\) and \(\mu_{k+1}^{\Delta} := E(x_{k+1}^{\Delta}|Y_k^{\Delta} = \hat{Y}_k^\Delta)\) their covariances do not depend on \(Y_k^{\Delta}\). The means are given by the Kalman equations

\[
\mu_{k+1}^{(N)} = A_N^N \mu_k^{(N)} + K_k^{(N)} (\Delta \hat{z}_{k+1} - H_{N,N} \mu_k^{(N)})
\]  (3.50)

and

\[
\mu_{k+1}^{\Delta} = A_\Delta \mu_k^{\Delta} + K_k^{\Delta} (\Delta \hat{z}_{k+1} - C_\Delta \mu_k^{\Delta}),
\]  (3.51)

where \(\mu_0^{(N)} = \mu_0^{\Delta} = \mu\). From Lemma 3.2.4 and Lemma 3.2.5 it follows that, using induction, \(\mu_{k+1}^{(N)} - \mu_{k+1}^{\Delta} = O(1/N)\) for all \(k \geq 0\), and given the special form of Gaussian densities. The Theorem follows.
Example

The above analysis is illustrated by a simple example. Consider the following continuous-time linear system

\[ dx(t) = -x(t)dt + d\omega(t) \]  \hspace{1cm} (3.52)
\[ dz(t) = x(t)dt + d\nu(t) \]  \hspace{1cm} (3.53)

where \( d\omega \) and \( d\nu \) have incremental covariance \( E\{d\omega d\omega^T\} = 1dt \) and \( E\{d\nu d\nu^T\} = 10dt \). The sample period \( \Delta = 1 \) and the up-sample factor \( N \).

The filtered state and its variance has been evaluated for (i) the exact discrete-time system, (ii) the approximate model at period \( \Delta \) and (iii) the up-sampled model with \( N = 2 \) and \( N = 10 \). In Figure 3.5 the posteriori probability density function at time \( t = 5 \), obtained by the different methods, is shown. (Namely, exact filter, up-sampled models and Euler approximation).

As predicted by the above analysis, the answer provided by the filter using the approximate model
at period $\Delta$ is a poor approximation to the “true” density. On the other hand, the up-sampled filter with $N = 10$ provides an answer which is almost indistinguishable from the exact result.

3.3 Lebesgue Sampling

Most current implementations of digital control and estimation use regular sampling with fixed period $\Delta$, see e.g. [101, 44, 15, 73]. However, there is often strong practical motivation to change this paradigm to one in which one only takes samples “when something interesting” happens. This changes the focus to, so called, “event based” sampling. In this section, it is considered that a measurement is sent only when the measured variable crosses a given threshold. Thus, the sampling is not regular. The latter strategy has many advantages, including conserving valuable communication resources, in the context of networked control or sensor networks.

There is a growing literature on event based sampling. An early seminal paper was that of [17] where the intuitive term of “Lebesgue sampling” is used for event based sampling. Other related publications include [14, 9, 8, 24, 110, 130, 87, 98, 114, 141]. As pointed out in [10], event based sampling and control are particularly attractive for non-linear systems since the nature of the system response can be operating point dependent and this may mean that different sampling strategies are desirable at different operating points.

The current section examines some of the issues related to event based sampling for non-linear filtering. An event based non-linear filter is developed.

One issue that needs careful consideration in the context of event based filtering is that of the anti-aliasing filter. It is argued here that an alternative viewpoint needs to be adopted for the design of this filter.
3.3 Lebesgue Sampling

3.3.1 Event Based Sampling

Let \( \{q_{ij}\} \) be a set of quantization levels for the \( j^{th} \) output. These quantization levels could, for example, be evenly spaced so that

\[
q_{i+1,j} - q_{i,j} = L_j \in \mathbb{R} \quad \text{for } j = 1, \ldots, n
\] (3.54)

In event based sampling, the measured output is sampled (and reset in the filter) only when a quantization level has been crossed. Moreover, provided no bits are lost and provided a starting signal level is known, then only 1 bit/sample needs to be sent to indicate that the signal has moved to the next interval above (+1) or the next interval below (-1). The difference between Riemann and Lebesgue sampling is illustrated in Figure 3.6.

Next consider the design of the anti-aliasing filter. Here, a little more care is needed than in the case of Riemann sampling. Specifically, it is required that “interesting events” should trigger sampling. This raises the need to trade-off noise immunity versus sensitivity to change. To illustrate, say that one uses the averaging filter given in (2.8), (2.9). Then a sudden change in output may be masked by the effect of averaging an (almost) constant signal over a long period of time. Hence it is desirable to place a lower limit on the bandwidth of the anti-aliasing filter so as to achieve a compromise between sensitivity and noise averaging. This trade-off does not arise in Riemann sampling since there is no need to detect changes. In the case of the averaging filter, the trade-off can be achieved by simply resetting the averager when the sample period goes beyond some pre-determined upper limit, say, \( \Delta_{max} \).

There also exists a close connection between, the choice of the anti-aliasing filter bandwidth and, the quantization thresholds used in the event based sampler. The reason is that, one needs to ensure that measurement noise does not cause frequent triggering of the event based sampling even if the signal component is substantially constant.

Simple design guidelines can be developed as follows:

Say that the measurement noise is broadband with spectral density \( R \) and that an anti-aliasing
filter with reset period $\Delta_{\text{max}}$ is used. Then the corresponding discrete measurement noise will have variance of approximately $R/\Delta_{\text{max}}$. Assume that the quantization level spacing is $L$ and say that spurious triggering of the event based sampler should be avoided with high probability. This can be achieved by requiring that there is only a small probability that the discrete measurement
noise has magnitude greater than \( L/2 \). To achieve this one might require, for example, that

\[
2\sigma \leq L/2
\]  
(3.55)

where \( \sigma \) is the discrete noise standard deviation, i.e. \( \sqrt{R/\Delta} \). Equation (4.25) is equivalent to

\[
L \geq 4\sqrt{R/\Delta_{\text{max}}}
\]  
(3.56)

This equation links the anti-aliasing filter bandwidth, \( 1/\Delta_{\text{max}} \), the noise spectral density \( R \) and the quantization level spacing \( L \) to achieve a low probability that the noise will be greater than \( L/2 \). In practice, it is desirable to choose \( \Delta_{\text{max}} \) as small as possible to satisfy (3.56) since large values of \( \Delta_{\text{max}} \) compromise one’s ability to detect changes in the signal component.

At first glance it may seem that the extension to event based sampling is immediate, i.e. all one need to do is run the state update (2.54) at period \( \Delta \) (chosen sufficiently small so that Euler Integration gives an adequate approximation) and then to use the observation update (2.55) when one decides that a “sufficiently interesting” change in the output has occurred. Certainly the observations are only needed when a threshold has been crossed. However, it is not true that there is zero relevant information between threshold crossings. On the contrary, there is a valuable piece of information, namely that the output has not crossed a threshold. Hence, estimates can continue to be updated between threshold crossings provided an appropriate change is made to the observation update formula. Specifically, consider the situation illustrated in Figure 3.7 where, at the \( k^{th} \) time instant, it is known that

\[
y_k \in [Q_a, Q_b] = \bar{Q}_k
\]  
(3.57)

The observation update (2.55) in the nonlinear filter can now be modified to the following which explicitly utilizes (3.57):

\[
p(x_{k+1}|y_k, y_{k+1} \in \bar{Q}_k) = \frac{\int_{\bar{Q}_k} p(x_{k+1}|y_k)p(y_{k+1}|x_{k+1})dy_{k+1}}{\int \int_{\bar{Q}_k} p(x_{k+1}|y_k)p(y_{k+1}|x_{k+1})dy_{k+1}dx_{k+1}}
\]  
(3.58)
Note that if one simply chooses not to update the states, then the state estimation uncertainty will grow due to the drift term inherent in the state update (2.37). Use of (3.58) avoids this problem. Actually, this is different from the common strategy used in much of the existing event based sampling literature where updates are usually restricted to cases when a threshold is crossed ([14, 9, 8, 24, 110, 130, 87, 98, 114, 141]). Some authors e.g. [125, 96] have noted that, for the case of linear filtering, it is desirable to continue to update based on the known fact that the output lies within the quantization threshold. This is the idea captured in (3.58) for the case of nonlinear filtering. Of course, in practice, the integrals in (3.58) will need to be approximated. The approximation issue is discussed later in Chapter 4.

### 3.3.2 Example

Consider the practical problem of radar tracking using range and bearing measurements. Event based sampling, is used and compared with regular sampling. Consider the following two state
model equation,

\begin{align*}
    x^{(1)}_{k+1} &= x^{(1)}_k + \Delta v^{(1)}_k + \omega^{(1)}_k \\
    x^{(2)}_{k+1} &= x^{(2)}_k + \Delta v^{(2)}_k + \omega^{(2)}_k
\end{align*}

(3.59) (3.60)

where \( \Delta = 0.1 \) is the sampling period and \( x = [x^{(1)} x^{(2)}] \in \mathbb{R}^2 \) is the state vector. The input \( v = [v^{(1)} v^{(2)}] \in \mathbb{R}^2 \) corresponds to the speed of the object in Cartesian coordinates, \( \omega = [\omega^{(1)} \omega^{(2)}] \in \mathbb{R}^2 \) is process noise (say wind gusts or unmeasured speed variations) with covariance:

\[
Q_d = \begin{bmatrix}
100 & 0 \\
0 & 100
\end{bmatrix} \Delta
\]

(3.61)

The range and bearing measurements are given by the following equations (46):

\begin{align*}
    y^{(1)}_k &= \sqrt{(x^{(1)}_k)^2 + (x^{(2)}_k)^2} + \nu^{(1)}_k \\
    y^{(2)}_k &= \arctan \left( \frac{x^{(1)}_k}{x^{(2)}_k} \right) + \nu^{(2)}_k
\end{align*}

(3.62) (3.63)

The measurement vector is thus \( y = [y^{(1)} y^{(2)}] \in \mathbb{R}^2 \), the measurement noise \( \nu = [\nu^{(1)} \nu^{(2)}] \in \mathbb{R}^2 \) is taken to have variance:

\[
R_d = \begin{bmatrix}
0.6 & 0 \\
0 & 0.06
\end{bmatrix} \frac{1}{\Delta}
\]

(3.64)

Regular (Riemann) sampling and event based (Lebesgue) sampling are compared. For the former, 8 bits were utilized to represent each sample and one sample was taken per second. For the second case, the quantization thresholds were set at 50 and 0.9 respectively for range and bearing. Figure 3.8 compares the reconstructed range and bearing for the two filters. Figure 3.9 shows the sampling times for range (upper two traces) and bearing (lower two traces).

It can be seen from Figure 3.8 that the estimates produced by Lebesgue sampling are extremely close to those produced by Riemann sampling. This occurs despite of the obvious difference in sampling rates shown in Figure 3.9. Indeed, the Riemann sampling strategy uses 8 bits/sample
and 1 sample/second, i.e. a data rate of 8 bits/second. On the other hand, the Lebesgue sampling strategy uses only 1 bit/sample (up or down) at an average of 0.2 samples/second. The latter
3.4 Conclusion

This chapter has studied the impact of sampling period on filtering problems. It has been seen that to have a fast sampling period can cause very high computational cost which does not necessarily yield improvements in a practical sense. To counter this, an effective downsampling technique has been developed. On the other hand, when a slow sampling rate is used, the errors generated by the model can have a dramatic impact on the result. In this case, it has been shown how to use an upsampled version of the nonlinear filter. This filter overcomes the errors arising in the model at a slow sampling rate. Finally, it has been shown how to extend the idea of nonlinear filtering to cases when event-based sampling is used. The appropriate scaling of variances has been derived and an appropriate version of the observation update equation developed.

This chapter has not covered numerical issues arising from the need to perform multidimensional integration to solve the problem of nonlinear filtering. Chapter 4 addresses these problems.
As is clear from the previous chapter, the discrete-time filtering equations cannot be solved explicitly in the nonlinear case. Hence some form of numerical approximation is necessary. Substantial effort has been invested over the past decades to address this problem. In the current chapter, additional ideas are proposed.

Most of the focus in approximation methods has been dedicated to the successful Monte-Carlo techniques, also known as Particle Filters. In this chapter, a novel breed of deterministic filters based on Vector Quantization is presented and applied to practical problems.

There exists a huge volume of research on approximate algorithms for discrete time nonlinear filtering. Useful reviews can be found in [19, 31, 13]. The existing algorithms can be broadly classified into 5 categories:

1. Linearization algorithms:

   - Here one linearizes about the current estimate $\hat{x}$. This leads to the Extended Kalman Filter (EKF) [7, 75]. Various embellishments are possible, e.g. re-linearizing about the updated estimates, leading to the Iterated Extended Kalman Filter (IEKF) [50, 75].

The advantage of these algorithms is that they are very simple. The disadvantage is that they will frequently fail when the nonlinearities are far from linear.
2. Mixed Algorithms:

- Here one uses a Gaussian approximation, but then chooses several representative points to pass through the nonlinearities. These are re-averaged after passing through the nonlinearity: An example of this class of algorithms is the Unscented Kalman Filter (UKF) \([76, 77]\). A more recent algorithm from the same general class is the algorithm described in \([13]\) which uses Gauss–Hermite Quadrature.

Again these algorithms are very simple. However, the disadvantage is that they only work for simple nonlinearities. Also they focus on estimating the mean of the posterior distribution. This can be acceptable in some case but is, in general, an inadequate description of the posterior distribution as is clear from the bimodal distribution shown in Figure 3.1.

3. Deterministic Griding Algorithms:

- One can obtain an approximate filtering algorithm by simply representing the distribution of the states on a finite grid. One choice would be a uniform grid. However, this is often infeasible since a very large number of grid points are typically needed. Furthermore, the number of grid points typically grows exponentially with the number of dimensions.

- Another related idea is to, a-priori, choose a grid that is more focused on the “likely” areas of the state space where the states might lie. For example \([111]\) uses vector quantization to choose a grid based on the prior distribution for the state \(\hat{x}\).

Unfortunately, these methods do not account well for disturbances or uncertainty in the state trajectories.

4. Monte Carlo/Particle Filtering:

- This technique accounts for disturbances by drawing a set of random samples from the disturbance distribution. Thus, a discrete approximation to the posterior distribution is generated which is based on a set of randomly chosen points. The approximation converges in probability with order \(1/\sqrt{N}\), where \(N\) is the number of chosen samples. The main disadvantages of this class of algorithm is that a very large number of points
may be needed and also these points need to be, in some sense, related to the distribution of interest. Also, the number of points grows exponentially with time unless some form of reduction is used. Thus, there are many ad-hoc fixes needed to get this type of algorithm to work in practice. Such fixes include the use of proposal distributions, resampling methods, etc see [31][122].

In this chapter a new algorithm based on vector quantization will be developed. Before describing the vector quantization idea, particle filtering methods will be briefly described.

The outline of the remainder of the chapter is as follows. Section 4.1 provides an introduction to particle filtering. Section 4.2 develops the new algorithm, Minimum Distortion Filtering (MDF). Section 4.3 presents several examples to illustrate the potential benefits of the MDF idea. Section 4.4 explains how the nonlinear system identification problem can be translated to one of nonlinear filtering. Section 4.5 uses the filters defined in Section 4.1 and 4.2 for an identification problem arising in systems having non-uniform sampling period. Section 4.6 shows the application of MDF techniques to a typical chemical process problem. Finally, Section 4.7 concludes the chapter.

4.1 Particle Filtering Methods

Here the common scheme for nonlinear filtering based on Particle filtering is described. Actually this is one of the most commonly used schemes in practical nonlinear filtering problems.

Particle methods deal with the problem of recursively estimating the probability density function \( p(\bar{x}_k|\mathcal{Y}_k) \) by using Monte Carlo ideas. The key idea is to represent the probability density function by a set of random samples having associated weights.

\[
p(\bar{x}_k|\mathcal{Y}_k) = \sum_{i=1}^{M} q_k^{(i)} \delta(\bar{x}_k - \bar{x}_k^{(i)}), \quad \sum_{i=1}^{M} q_k^{(i)} = 1, \quad q_k^{(i)} \geq 0 \tag{4.1}
\]

where \( \delta(\cdot) \) is the Dirac delta function and \( q_k^{(i)} \) denotes the weight associated with the particle \( \bar{x}_k^{(i)} \). The subscript \( k \) indicates the discrete-time index and the superscript \( (i) \) denotes a “particle”.

In obtaining this approximation, one has to be able to draw random samples from general dis-
tributions. The approximation (4.1) can also be obtained using stochastic integration ideas, see e.g., [52, 20] for related, but slightly different, approaches. In practice, one needs to use a relatively large number of random samples to adequately represent a given distribution. It is important to note that, the problem of generating random samples from complicated distributions has previously been assessed in a non-recursive setting using Markov Chain Monte Carlo methods (MCMC).

The generation of the random samples presents a major problem. In the literature, one can find various ideas on how to handle the fact that one cannot generate samples directly from the target density. One option is to use a marginalized particle filter. This method can be employed when there is a linear, Gaussian sub-structure available in the model equations. For further details on this topic see [122] and the references therein. Further details regarding the particular implementation used in this thesis are outlined below:

4.1.1 Random Number Generation

The problem of interest is to generate samples from some known probability density function, referred to as the target density $t(x)$. However, since one cannot generate samples from $t(x)$ directly, the idea is to employ an alternate density that is simpler to draw samples from, referred to as the sampling density $s(x)$. When a sample $\bar{x} \sim s(x)$ is drawn, the probability that it was in fact generated from the target density can be calculated. This probability can then be used to decide whether $\bar{x}$ should be considered as a sample from $t(x)$ or not. This probability is referred to as the acceptance probability, and it is typically expressed as a function of another variable $q(\bar{x})$, defined by the following relationship,

$$t(\bar{x}) \propto q(\bar{x})s(\bar{x})$$

(4.2)

Depending on the exact details of how the acceptance probability is computed different methods are obtained. Some of the best known methods are Sampling Importance Resampling, Acceptance-Rejection Sampling and Metropolis - Hastings Independence Sampling. For a more detailed explanation, see, e.g., [122, 118, 53, 132]. A comparison of different methods is provided in [94].
4.2 Minimum Distortion Filtering

4.1.2 Particle Filter

In the Bayesian framework used in the current Thesis, the observation update equation given by (2.55) which is then interpreted as in (4.2). Then, one can use this to define the target and sampling density as follows:

$$p(x_{k+1}|y_{k+1}) = \frac{p(x_{k+1}|y_k)p(y_{k+1}|x_{k+1})}{\int p(x_{k+1}|y_k)p(y_{k+1}|x_{k+1})dx_{k+1}} \propto p(x_{k+1}|y_k)p(y_{k+1}|x_{k+1})$$  \hspace{1cm} (4.3)

The typical Particle Filter is derived using the Sampling Importance Resampling technique. Many derivations can be found in the literature, see \[70, 64, 122\]. Later it was independently rediscovered by \[81, 74\]. Some early ideas relating to the particle filter are given in \[100, 69, 2, 71\].

The Particle Filter used in this Thesis is described in Table 4.1. It is based on ideas presented in \[122\]. (See algorithm 4.4 of \[122\]).

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Initialization: Quantize $p(x_0)$ to $M$ particles ${x_{0(i)}}_{i=1}^M$;</td>
</tr>
<tr>
<td>2</td>
<td>Use the State Update equation to propagate the Particles to produce $p(x_{k+1}</td>
</tr>
<tr>
<td>3</td>
<td>Measurement Update: Calculate the importance weights ${q_k^{(i)}}<em>{i=1}^M$ according to $q_k^{(i)} = \frac{p(y</em>{k+1}</td>
</tr>
<tr>
<td>4</td>
<td>Resampling: Draw $M$ particles, with replacement, according to $p(x_{k+1} = x_{k+1}^{(i)}) = q_k^{(i)}$</td>
</tr>
<tr>
<td>5</td>
<td>Iterate from step 2</td>
</tr>
</tbody>
</table>

Next a novel class of nonlinear filtering algorithms developed by the candidate (in collaboration with others)\(^1\) is described.

4.2 Minimum Distortion Filtering

The key idea underlying this class of algorithm is to utilize Vector Quantization to generate, online, a finite approximation to the a-posteriori distribution of the states.

Say that one begins with a discrete approximation to the distribution of $x_0$ on $N_x$ grid points. Also

\(^1\)“see list of authors” publications for details of percentages of contributions.
assume that one has a finite approximation to the distribution of the process noise on \( N_w \) grid points. Then, utilizing the discretized version of equation (2.54), one obtains a finite approximation to \( p(x_1) \) on \( N_x \times N_w \) grid points. Then, one uses the discrete equivalent of (2.55) to obtain a finite approximation to \( p(x_1|y_1) \) on \( N_x \times N_w \) points. Finally, one uses vector quantization ideas to re-approximate \( p(x_1|y_1) \) back to \( N_x \) points. (The next section describes, in detail, how this crucial step is performed). Then, one returns to the beginning to obtain a discrete approximation to \( p(x_2|y_1) \) on \( N_x \times N_w \) points and so on. The algorithm is summarized in Table 4.2.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Initialization: Quantize ( p(x_0</td>
</tr>
<tr>
<td>1</td>
<td>Approximate ( p(x_{k+1}</td>
</tr>
<tr>
<td>2</td>
<td>Update the new weights evaluating ( p(x_{k+1}</td>
</tr>
<tr>
<td>3</td>
<td>Quantize back to ( N_x ) points. Go to 2.</td>
</tr>
</tbody>
</table>

The key step in the MDF algorithm is the vector quantization step (step 5 in Table 4.2). Details of this step are given below.

Assume one has a vector discrete distribution for some distribution \( p(x) \), where \( x \in \mathbb{R}^n \), is quantized to a very large (but finite) set of points. The goal is to quantize \( p(x) \) to a smaller finite set of points \( x_i, p_i \), \( i = 1, \ldots, N \). The first step in Vector Quantization is to define a measure to quantify the “Distortion” of a given discrete representation. This measure, is then optimized to find the optimal representation which minimizes the cost. In summary, one looks for a finite set \( \mathcal{W}_x = \{x_1, \ldots, x_N\} \) and an associated collection of sets \( \mathcal{S} = \{S_1, \ldots, S_N\} \) such that \( \bigcup_i S_i = \mathbb{R}^n \) and \( S_i \cap S_j = 0; i \neq j \). \( \mathcal{W}_x, \mathcal{S}_x \) are chosen by minimizing a cost function of the form:

\[
\mathcal{J}(\mathcal{W}_x, \mathcal{S}_x) = \sum_{i=1}^{N} E \{ (x - x_i)^T W (x - x_i) | x \in S_i \} P[S_i]
\]

where \( W = \text{diag}(W_1, \ldots, W_n) \). Other choices of the distance measure, can also be used; e.g. Manhattan, \( L_1 \), Jaccard, etc..., see [131].

If one fixes \( x_1, \ldots, x_N \) (the set of grid points), then the optimal choice of the sets \( S_i \) is the, so
4.2 Minimum Distortion Filtering

called, Voronoi cells \[65, 51\]

\[S_i = \{ x | (x - x_i)^T W (x - x_i) \leq (x - x_j)^T W (x - x_j) ; \forall j \neq i \} \] (4.5)

Similarly, if one fixes the sets \( S_1, \ldots, S_N \), then the optimal choice for \( x_i \) is the centroid of the sets \( S_i \), i.e.

\[ x_i = E(x | x \in S_i) \] (4.6)

Many algorithms exist for minimizing functions of the form (4.4) to produce a discrete approximation. One class of algorithm (known as Lloyd’s algorithm \[65, 51, 95\]) iterates between the two conditions (4.5) and (4.6).

Thus Lloyd’s algorithm begins with an initial set of grid points \( W_x = \{ x_i ; i = 1, \ldots, N_x \} \). Then one calculates the Voronoi cells \( S_x \) of \( W_x \) using (4.5). Next, one computes the centroids of the Voronoi cells \( S_x \) via (4.6). We then return to the calculation of the associated Voronoi cells and so on. Lloyd’s algorithm iterates these steps until the distortion measure (4.4) reaches a local minimum, or until the change in the distortion measure falls below a given threshold, i.e.

\[ \frac{J(W^k_x, S^k_x) - J(W^{k+1}_x, S^{k+1}_x)}{J(W^k_x, S^k_x)^k} \leq \varepsilon \] (4.7)

where \( W^k_x \) and \( S^k_x \), is the codebook and Voronoi cells at iteration \( k \) respectively.

4.2.1 Algorithm Details

In order to obtain satisfactory results using the MDF algorithm various embellishments are needed in the basic algorithm. We outline these embellishments below.

Local Minimum

The distortion measure (4.4) is a non-convex function of the unknowns \( W_x \) and \( S_x \). Hence, all algorithms for Vector Quantization including Lloyd’s algorithm, converge, at best, to a local mini-
mum of the cost function. Thus when initializing the algorithm, e.g. by selecting discrete approximations to $p(x_0)$ and $p(\omega)$, it is necessary to begin with a range of initial guesses for the grid. Of course, this step is carried out off-line and thus having many starting points does not affect the on-line computational load.

**Fast Sampling**

Further to the issue raised above, the Vector Quantization algorithm will also face local minimum issues at every time step. In this regard, the use of fast sampling is extremely beneficial. The reason is that, with fast sampling, the posteriori distribution for $x$ i.e. $p(x_k|\mathcal{Y}_k)$ evolves slowly from sample to sample (see detailed discussion in Chapter 3). Hence, if one has a good discrete grid at time $k$, then this will be an excellent initial condition for the grid at time $k+1$. This mitigates the local minimum problem. Indeed, in the simulation studies presented later it has been found that one iteration of Lloyd’s algorithm often suffices at each time step. This leads to a substantial simplification in the computational load.

**Scaling**

Numerically, the choice of $W$ in (4.5) is very important. The reason is that physical variables often have very different scales; e.g. a temperature might be 300 °C whereas a concentration might be 0.8.

To deal with this issue, it is desirable to scale the variables. One idea that has been useful is, at each time step, to scale via the current marginal variance, i.e. to choose

$$W_i = \frac{1}{N_x N_\omega \sum_j (x_{i,j} - \bar{x}_i)^2 p_j}$$

where the subscript $i$ identifies each state, the subscript $j$ identifies the quantization points and
where

\[ \bar{x}_i = \sum_j x_{i,j}p_j \]  \hspace{1cm} (4.8)

### 4.2.2 Clustering

As remarked earlier, fast sampling ensures that the posteriori distribution evolves slowly. Nonetheless, with a small number \((N_x)\) of grid points, it is still possible that grid points from the previous time step become inappropriate for the next time step. This is especially true for grid points at the extremities. It is crucial to address this problem for the satisfactory operation of the algorithm. Thus, a mechanism for detecting poor grid points is necessary.

First note that, inherent in the algorithm is a calculation of the individual distortion associated with each cell. Thus one can compute

\[ J_i = \sum_x \{(x - x_i)^T W_i(x - x_i)\mid x \in S_i}\]  \hspace{1cm} (4.9)

Then one compares these with a given threshold \(\zeta\). Let \(S_{\zeta} = \{i \mid J_i \leq \zeta\}\). Next, choose \(j \in S_{\zeta}\) such that \(J_j \leq J_i\) for \(i \in S_{\zeta}\). One then moves \(x_j\) into the cell \(S_k\) where \(k = \arg \max_i J_i\) to a point different from \(x_k\). One then form \(S_{\zeta}' = \{i \mid J_i \leq \zeta; i \neq j\}\). Next choose the index in \(S_{\zeta}'\) with smallest distortion and move the corresponding \(x_i\) to the set corresponding to the second largest distortion. One continues until \(S_{\zeta}'\) is emptied. Then, the new Voronoi cells are evaluated. This is straightforward as one only needs to calculate distances for the grid points that have been moved. Then, again check if any cell has distortion less than \(\zeta\). One has to continue until no cell has distortion less than \(\zeta\). Only then, does one proceed to the next sample time.

A key point is that this internal loop, which eliminates “poor” grid points, is computationally straightforward because one only needs to compute distances for the points that are relocated.

### 4.3 Simulation Examples

Four examples are used to illustrate the power of the method. Also a comparison with several other algorithms available in the literature is provided.
4.3.1 Example 1

Consider again the Markovian system from example 3.1.1:

\[
\begin{align*}
\frac{dx^+}{dt} &= x_{k+1} - x_k = -1x_k \Delta + \omega_k \\
\frac{dz^+}{dt} &= (x_k)^2 \Delta + \nu_k
\end{align*}
\]  

(4.10) (4.11)

where \( E\{\omega_k \omega_k^T\} = 10 \Delta \), \( E\{\nu_k \nu_k^T\} = 10 \Delta \) and \( \Delta \) is the sampling period.

Assume a fast sampling rate\(^2\) with \( \Delta = 0.1 \).

In general, it is impossible to compute \( p_{x_k}(x_k|Z_k) \) exactly. However, because this is a simple example, it is possible to obtain a very good approximation by using a fine grid. This is used to have a “benchmark” against which to compare the performance of the MDF algorithm. In particular, 1001 points to grid \( x \) between \( -10 \) and \( +10 \) are used.

![Quantized Vector evolution](image)

**Figure 4.1:** Grid points corresponding to algorithm (i)

The performance of three algorithms is compared.

(i) The “true” evolution of \( p_{x_k}(x_k|Z_k) \). Then, so as to form a comparison with the other algorithms, a quantized form of this “true” distribution is produced at each sample by running many

---

\(^2\)Note that \( \Delta = 0.1 \) could be considered a moderate sampling rate. However, it is \( 1/10^{th} \) the response time of the system and this is sufficiently “fast” in the context of this example.
iterations of Lloyd’s algorithm. (This result is used as a benchmark against which to compare the other algorithms.)

(ii) The MDF algorithm is used. However, only one iteration of the Lloyd algorithm, in the quantization step, per sample is performed.

(iii) The standard MDF algorithm is used with many iterations of Lloyd’s algorithm.

To provide a common basis for comparing the three algorithms, the resulting grid points at each sample time are plotted. The results are shown in Figures 4.1, 4.2, 4.3 respectively for algorithms
Comparison of Figures 4.1 and 4.3 shows that the multi-step Lloyd algorithm gives excellent results for this example. Comparison of Figures 4.2 and 4.3 shows some deterioration when only one iteration of the Lloyd algorithm is used per sample. However, key features of the posteriori distribution are captured. This is further illustrated in Figure 4.4 which shows the cumulative probability distribution \( \int_{-\infty}^{x_k} p_{x_k}(x_k|Z_k)dx_k \) for the three algorithms at sample 10 together with the cumulative distribution corresponding to the “true” posteriori density (calculated with a fine grid). It is clear that the MDF algorithm gives a very good approximation to the probability distribution for this example.

4.3.2 Example 2

This problem was suggested in [3]. The system has several inherit difficulties. For example, it is argued in [3] that any algorithm of the Extended Kalman Filter type for estimating the posterior mean will be incapable of crossing “various” boundaries in the state space, i.e. it will “lock up”.

The system in continuous time is:

\[
\begin{align*}
    dx &= [x(t) - x(t)^3]dt + d\omega \\
    dz &= [x(t)^2 - 0.5x(t)]dt + d\nu
\end{align*}
\]
Figure 4.5: Time evolution of the probability density function at fast sampling, \( \Delta = 0.001 \)

Figure 4.6: Cumulative distribution function at \( k=1 \)

where \( d\omega \) and \( d\nu \) have incremental covariance \( Q dt = R dt = 0.1 dt \).

Consider an approximate fast sampled version of this system, i.e.:

\[
\begin{align*}
\dot{x}^+ &= [x_k - x_k^3] \Delta + \omega_k \quad (4.14) \\
\dot{z}^+ &= [x_k^2 - 0.5x_k] \Delta + \nu_k \quad (4.15)
\end{align*}
\]

where \( \omega \) and \( \nu \) have incremental covariance \( Q_i \Delta = R_i \Delta = 0.1 \Delta \).
Choose $\Delta = 10^{-3}$ to allow the posterior density to evolve smoothly, from sample to sample. Also, to capture the “lock up” phenomena reported in [3], the initial state distribution is chosen to have a gaussian distribution with mean $-1$ and covariance 1. Also the true (but unknown to the estimator) initial state is 0.5, the true state evolves and finally settles near $+1$. A feature of this problem is that the transients in the state are very rapid (due, in part, to the presence of the $x^3$ term in the state evolution). Figure 4.5 shows the evolution of the “true” posterior probability density calculated on a fine grid. Note that the “true” distribution evolves so as to be concentrated near $+1$.

Algorithms (i) and (ii) are compared, as discussed in relation to example 1. Figures 4.6, 4.7, 4.8.
4.3 Simulation Examples

4.3.3 Example 3

This example is a simplified radar tracking problem. Assume measurement of range and bearing \( y = [r; \theta] \) in polar coordinates, of an object \( A \). Also, assume that at a given time instant the distribution for \( r \) and \( \theta \) is randomly Gaussian distributed with \( N \sim (\bar{y}, \Sigma) \), where \( \bar{y} = [80; 0.61] \) and \( \Sigma = \text{diag}(60, 0.6) \). The goal is to obtain an estimate of the position of the object in Cartesian Coordinates. The mapping is described by

\[
x = \begin{bmatrix} r \cos \theta \\ r \sin \theta \end{bmatrix}
\]  

(4.16)

The MDF algorithm (5 points per dimension) is applied to this mapping to obtain an estimate of the mapping of \( A \) to Cartesian Coordinates. This result is then compared with the “true” answer obtained by using a deterministic finite grid with a large number (\( L=3000 \)) of points.

Figure 4.10 shows the answer provided by the 5 point, per dimension, MDF algorithm (red circles).
As expected, the algorithm allows one to estimate the shape of the cloud of points (true answer). Moreover, the MDF algorithm provides a satisfactory answer to moment estimation for the first and second order moments and above. This is illustrated in Table 4.3 which shows average results based on 100 trials.

Table 4.3: Average Result: Numerical Radar Tracking

<table>
<thead>
<tr>
<th>Moment</th>
<th>Value x &amp; y</th>
<th>Error %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>49.45 &amp; 34.56</td>
<td>1.80 &amp; 1.82</td>
</tr>
<tr>
<td>2</td>
<td>1120 &amp; 1695</td>
<td>7.02 &amp; 2.76</td>
</tr>
<tr>
<td>3</td>
<td>-46116 &amp; -56619</td>
<td>12.55 &amp; 1.85</td>
</tr>
</tbody>
</table>

Another example of this situation is described in [135], where format conversion for cellular positioning technology is described.
Finally, the radar tracking problem using range and bearing measurements as discussed in Section 3.3.2 is revisited. Recall the two state model used in (3.59), (3.60). Consider the following two state model,

\[
x_{k+1}^{(1)} = x_k^{(1)} + \Delta v_k^{(1)} + \omega_k^{(1)} \\
x_{k+1}^{(2)} = x_k^{(2)} + \Delta v_k^{(2)} + \omega_k^{(2)}
\]

where \(\Delta = 0.1\) is the sampling period and \(x = [x^{(1)} \ x^{(2)}] \in \mathbb{R}^2\) is the state vector. The input \(v = [v^{(1)} \ v^{(2)}] \in \mathbb{R}^2\) corresponds to the speed of the object in cartesian coordinates, \(\omega = [\omega^{(1)} \ \omega^{(2)}] \in \mathbb{R}^2\) is process noise (say wind gusts or unmeasured speed variations) with covariance:

\[
Q_d = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix} \Delta
\]

The range and bearing measurements are given by the following equations (4.6):

\[
y_k^{(1)} = \sqrt{(x_k^{(1)})^2 + (x_k^{(2)})^2} + \nu_k^{(1)} \\
y_k^{(2)} = \arctan \left( \frac{x_k^{(1)}}{x_k^{(2)}} \right) + \nu_k^{(2)}
\]

The measurement vector is thus \(y = [y^{(1)} \ y^{(2)}] \in \mathbb{R}^2\), the measurement noise \(\nu = [\nu^{(1)} \ \nu^{(2)}] \in \mathbb{R}^2\) is taken to have variance:

\[
R_d = \begin{bmatrix} 0.6 & 0 \\ 0 & 0.06 \end{bmatrix} \Delta^{-1}
\]

The MDF tuning parameters are taken to be \(N_x = 49\), \(N_w = 9\), \(\zeta = 10^{-20}\) and \(\epsilon = 10\%\). For the particle filter 1000 particles were used. This approximately yields equal computational load per sample for MDF and particle methods. Both filters used the same initial condition for the state, i.e. a gaussian distribution with \(\hat{x}_0 = [35 \ 23]\) and covariance \(\hat{P}_0 = [100 \ 0; 0 \ 100]\).
Figures 4.11 and Figure 4.12 show the mean and variance of the state estimate. As can be seen, the MDF and Particle Filter give similar results. These results were computed using a very fine gridding of the state space.

Table 4.4 shows the root mean square error for the mean and variance estimates using MDF and PF algorithms. These results show that the performance of the MDF algorithm is better than that obtained by PF methods.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Mean $x_1$</th>
<th>Mean $x_2$</th>
<th>Variance $x_1$</th>
<th>Variance $x_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDF</td>
<td>4.043</td>
<td>4.5689</td>
<td>29.109</td>
<td>27.585</td>
</tr>
<tr>
<td>PF</td>
<td>6.644</td>
<td>6.870</td>
<td>38.023</td>
<td>46.6718</td>
</tr>
</tbody>
</table>
4.4 Systems Identification as a Nonlinear Filtering Problem

It is well known that parameter estimation can be combined with state estimation by augmenting the model with additional states for the parameters. Thus, if one begins with a discrete-time model, then the following model which explicitly includes the parameters is obtained.

\[
x_{k+1} = f_d(\Delta_k, \theta_k)x_k + B_d(\Delta_k, \theta_k)u_k + \omega_k \\
y_{k+1} = h_d(\theta_k)x_k + \nu_k
\]

(4.23)

(4.24)

where \( \theta_k \) is the unknown parameter vector. The time evolution of this vector is modelled by an additional state space model. For example a random-walk process leads to:

\[
\theta_{k+1} = \theta_k + \omega^{(2)}_k
\]

(4.25)
where $\omega_k^{(2)}$ is assumed to be white Gaussian noise with zero mean and variance $Q_\theta$. The covariance $Q_\theta$ models how much we expect the parameter $\theta$ to change over time. In turn, this affects the “memory” of the parameter estimator. For example, if $Q_\theta$ is large, then the model predicts rapid parameter variations. The filter should then automatically “discard” the data save very recent observations. Conversely, if $Q_\theta$ is small, then the model predicts slow parameter variations and then the filter should “retain” the data save for observations that are far removed from the present time.

Remark 4.4.1 Even if one believes that $\theta$ is constant, it is usually a good idea to use a value of $Q_\theta$ different from zero. The reason is that, otherwise, the parameter estimator will “lock-up” and not use the on-going data. This is acceptable under ideal conditions but can lead to erroneous estimates in practical cases, e.g. due to the influence of outliers.

4.5 Non-linear Filtering Applied to System Identification with Non-uniform Sampling period

This idea is used to illustrate the power of nonlinear filtering applied to parameter estimation. Most physical systems are described by continuous-time models. However, in practice, one needs to interact with these systems in some way. Thus, real implementations are subject to different constraints, such as sensors, actuators and communication channels. These add extra ingredients to the problem. As discussed in detail in Chapter 3, a key issue in almost all problems is that of sampling. Sampling provides the link between continuous-time systems and discrete-time models.

Discrete system theory for uniformly sampled linear systems has been studied over many decades using purely discrete methods [44]. However, our interest here is in non-uniform sampling. In this case, it is convenient to model the system in continuous time since the associated parameters are then invariant with respect to different sample periods. The situation differs from the one in Section 3.3 due to the fact that measurements are not triggered by the measured signal but by other issues, e.g. processing time, communication channels, etc.

For the linear case, assuming that a zero-order hold (ZOH) is used for the input signal $u(t)$. Then,
4.5 Non-linear Filtering Applied to System Identification with Non-uniform Sampling period

the corresponding discrete-time model with sample period $\Delta_k$ takes the form:

$$x_{k+1} = A_d(\Delta_k)x_k + B_d(\Delta_k)u_k + \omega_k$$  \hspace{1cm} (4.26)$$

where $k$ is the discrete-time index.

Also, as discussed in Chapter 2, assume the following discrete-time measurement equation:

$$y_{k+1} = C_dx_k + \nu_k$$  \hspace{1cm} (4.27)$$

In (4.26), (4.27), $\omega_k$ and $\nu_k$ are discrete-time white noise processes having covariance matrix

$$\Sigma_d(\Delta_k) = \begin{bmatrix} Q_d & 0 \\ 0 & R_d \end{bmatrix} = \begin{bmatrix} \Delta_k(Q_c + \frac{\Delta_k^2}{2}(A_cQ_c + Q_cA_c^T) + \ldots) & 0 \\ 0 & R_d(\Delta_k) \end{bmatrix}$$  \hspace{1cm} (4.28)$$

where $Q_d$ is a symmetric semi-positive definite and $R_d$ is positive definite.

The system matrices in (4.26) are given by

$$A_d(\Delta_k) = e^{A_c\Delta_k} = I + A_c\Delta_k + \frac{1}{2}A_c^2\Delta_k^2 + \ldots$$  \hspace{1cm} (4.29)$$
$$B_d(\Delta_k) = A_c^{-1}(e^{A_c\Delta_k} - I)B_c = (B_c + \frac{1}{2}A_cB_c\Delta_k + \ldots)\Delta_k$$  \hspace{1cm} (4.30)$$

where $A_c \in n \times n$, $B_c \in n$, $C_d \in p \times n$. The equations (4.28), (4.29) are similar to those presented in Section 2.3.1 with the addition of (4.30). The main difference is that in the discrete-time model described above the time-varying sampling sequence $\{\Delta_k\}$ is explicit, where

$$\Delta_k = t_{k+1} - t_k > 0 \quad \forall k \in \mathbb{N}$$  \hspace{1cm} (4.31)$$

Here, $\{t_k\}$ denotes the sample times. Assume that data is collected on the time interval $[0; T_f]$, where $T_f = \sum_{k=0}^{N_f-1} \Delta_k$ and $N_f$ is the total number of samples.
4.5.1 Non-Uniform Sampling in the Identification Problem

In this section the non-uniform sampling system identification problem is formulated. When the sampling rate is constant and the underlying continuous time system is time invariant, then the discrete-time model is time invariant. In this case, the parameters can be estimated directly on the discrete-time model. There are several possibilities. For example, one can first estimate the discrete-time model matrices. In a second step, a transformation can be applied to recover the continuous time model matrices. This operation may involve computation of the logarithm of the system matrix or the use of Padé like approximations \[99, 136\]. Different derivative approximations can also be used to discretize the model. However, the choice of the particular approximation may have a direct impact on the quality of the estimates (See, for example, \[86\]).

When the sampling rate is non-uniform, the discrete-time model becomes time varying. In this case, estimating the continuous parameters becomes more difficult.

Continuous-time system identification from non-uniform sampled-data has been considered from several different perspectives. For example, in \[54\] approximate output spectrum reconstruction is performed using B-spline functions. Another identification procedure for the non-uniform sampling was proposed in \[37\]. In \[37\] a least squares approach is used, where the states are estimated using a Kalman filter in shift operator form.

Here, a different approach is used. The sampling period is included explicitly in the discrete-time model\(^3\), retaining the continuous-time parameters. This leads to the following model which restates \[4.26\], \[4.27\):

\[
x_{k+1} = e^{A_c \Delta_k} x_k + A_c^{-1}(e^{A_c \Delta_k} - I) B_c u_k + \omega_k \quad (4.32)
\]

\[
y_{k+1} = C_d x_k + \nu_k \quad (4.33)
\]

where \(A_c, B_c\) and \(C_c\) are the associated continuous time matrices, \(\Delta_k\) is the sampling period between the \(k^{th}\) and \((k + 1)^{th}\) samples. The process noise \(\omega_k\) is gaussian and has a time varying covariance which is proportional to the sampling period \(Q_k = \Delta_k Q\). The measurement noise is

\(^3\)Note that \(A_c^{-1}\) need not be non singular since the singularity cancels in the following model.
4.5 Non-linear Filtering Applied to System Identification with Non-uniform Sampling period

considered gaussian and independent of $\omega_k$. For simplicity, it is assumed that the measurement noise has fixed covariance $R_d$.

**Remark 4.5.1** This formulation leads to a linear discrete-time model with respect to the states, but with significant nonlinearities with respect to the parameters.

4.5.2 Simulation Example

Here, a simple example to illustrate the ideas discussed above is used. Assume an underlying first order continuous-time system of the form:

$$dx(t) = ax(t)dt + bu(t) + d\omega$$  \hspace{1cm} (4.34)

where $d\omega$ is defined as in Chapter 2 and has variance $0.1dt$.

$$\Delta_k vs Sample$$

*Figure 4.13: $\Delta_k$ vs Sample*
Non-uniform sampling is used with a total of \( K = 100 \) samples. The upper bound \( \Delta_{\text{max}} = 0.5 \) on the sample period is defined. For estimation purposes the problem is assumed to have two parameters, namely \( a \) and \( b \). The true parameters are taken to be \( a = -1 \) and \( b = 5 \).

The model used for parameter estimation is an extension of (4.32), (4.33) using the technique described in Section 4.4, i.e., the augmented state equation is

\[
\begin{align*}
    x_{k+1} &= e^{a_k \Delta_k} x_k + a_k^{-1} (e^{a_k \Delta_k} - 1) b_k u_k + \omega_k^{(1)} \\
    a_{k+1} &= a_k + \omega_k^{(2)} \\
    b_{k+1} &= b_k + \omega_k^{(3)}
\end{align*}
\]

(4.36)

(4.37)

(4.38)

The measurement equation is taken to be

\[ y_k = x_k + \nu_k \]

(4.39)

Here the extended state vector is \( \bar{x}_k = [x_k; a_k; b_k] \); \( \omega_k^{(1)}, \omega_k^{(2)}, \omega_k^{(3)} \) and \( \nu_k \) are assumed to have variance \( 0.1 \Delta_k, 0.005 \Delta_k, 0.0005 \Delta_k \) and 0.1 respectively. Note that when one samples a continuous-time system, the variance of the process noise typically grows proportional to the sampling period (see Section 2.3.2).

Figure 4.13 shows one particular realization of the sampling period \( \Delta_k \). Note that this realization changes for each experiment.

**MDF**

The MDF algorithm is used to obtain an estimate of the continuous-time parameters. The density function of the augmented state is quantized on \( N_x = 27 \) points. Note that one is quantizing a 3 dimensional state vector. Another \( N_w = 27 \) points are used to quantize the process noise.
4.5 Non-linear Filtering Applied to System Identification with Non-uniform Sampling

period

The filter is initialized with $p(\bar{x}_0)$ having a Gaussian distribution with random initial value and variance $Q_0 = diag(3, 3, 3)$.

**Particle Filter**

The Particle Filter parameters chosen for this simulation are as described in Section 4.1.2 with 100,000 particles.

**Results**

Here, the results obtained using the MDF and Particle Filtering schemes are contrasted. Figure 4.14 shows the “true” state evolution and the estimate obtained by both algorithms. Clearly, both algorithms provide good estimates for the state, although the Particle Filter achieves a slightly better result. (This is perhaps not surprising given that the MDF uses 27 points in the approximation
of the posterior distribution for $\bar{x}$ whereas the Particle Filter uses 10,000 points.)

![Image](image_url)

**Figure 4.15**: Mean value of $p(a_k|Y_k)$ at each sample $k$. Particle Filter (black: solid) MDF (blue: dashed).

Figures 4.15 and 4.16 show the estimates for the parameters obtained using the MDF algorithm and Particle Filter for a particular data set. It is clear from the figures, that the MDF algorithm provides much better performance than the Particle Filter. Of course it is possible that better results could be obtained if the Particle Filter were to be refined. Thus no general claim from this example is made.

**Remark 4.5.2** It is worth noticing that the MDF algorithm uses only 27 points to quantize the distribution, yet provides excellent results.

Figure 4.17 and 4.18 shows the cumulative distribution at a particular sample time for the MDF algorithm on a particular data set. The distribution for the parameters can clearly be seen to be non Gaussian which confirms the nonlinear nature of the continuous-discrete time mapping. To reinforce this idea, also the discrete probability density function at the same sample time is shown in Figures 4.19 and 4.20.
4.5 Non-linear Filtering Applied to System Identification with Non-uniform Sampling period

**Figure 4.16:** Mean value of $p(b_k|y_k)$ at each sample $k$. Particle Filter (black: solid) MDF (blue: dashed)

**Figure 4.17:** Cumulative marginal distribution $p(a_k < A_k|y_k)$ at $k = 17$
4. Spatial Sampling and Numerical Issues

Figure 4.18: Cumulative marginal distribution $p(b_k < B_k|y_k)$ at $k = 17$

Figure 4.19: Density function $a$ at $k = 17$

Robustness

To illustrate the robustness of the MDF algorithm, the simulation was performed 20 times with a different seed. In particular, different initial conditions for the real state, $p(\bar{x}_0)$ and the sampling
period sequence \( \{ \Delta_k \} \) are used. The average of the estimates after a fixed number of samples are compared in Table 4.5. Note that the averages are after a fixed number of samples which will correspond to different time periods depending on the realization of the sampling period sequence.

It can be seen that the MDF algorithm provides consistent estimates. On the other hand, the Particle Filter has trouble finding the continuous-time parameters.

### 4.6 MDF applied to System Identification in Chemical Processes

The application of MDF to a standard model for a Continuous Stirred Tank Reactor (CSTR) \([123]\) is presented. This system has found widespread application in industry and embodies many features of other types of systems.
Consider a simple liquid phase, irreversible chemical reaction where chemical A reacts to form species B. The rate of the reaction is considered to be of first order with respect to A.

To simplify the model assume the following:

- The CSTR is perfectly mixed
- The mass densities of the feed and product streams are equal and constant
- The liquid volume \( V \) in the reactor is kept constant by an overflow line
- The thermal capacitances of the coolant and the cooling coil wall are negligible compared to the thermal capacitance of the liquid in the tank
- All of the coolant is at a uniform temperature \( T_c \)
- The rate of heat transfer from the reactor content to the coolant is given by
  \[
  Q = UA(T_c - T)
  \]  
  where \( U \) is the overall heat transfer coefficient and A is the heat transfer area. Both of these model parameters are assumed to be constant and known
- The enthalpy change for the chemical reaction. (i.e. the heat of mixing) is negligible compared to the heat of reaction
- Shaft work and heat losses can be neglected

Under the above assumptions one has the following set of equations describing the reaction:

\[
\dot{C}_A = \frac{q}{V}(C_{Af} - C_A) - k_0 e^{-E/(RT)} C_A + \dot{\omega}_1(t) \quad (4.41)
\]
\[
\dot{T} = \frac{q}{V}(T_f - T) + \frac{\Delta H_R}{\rho \cdot C_p} k_0 e^{-E/(RT)} C_A + \frac{U \cdot A}{V \rho C_p} (T_c - T_A) + \dot{\omega}_2(t) \quad (4.42)
\]
where $\dot{\omega}_1, \dot{\omega}_2$ denote continuous “white noise” sources and where the states and control signal are,

$$\begin{bmatrix}
  x_1 \\
  x_2 
\end{bmatrix} = 
\begin{bmatrix}
  C_A \\
  T_A 
\end{bmatrix} ;
\quad u = T_c 
\quad (4.43)$$

The noise sequences are defined as in Chapter 2 and the constants of the model are given in Table 4.6. Assume that the constant $E$ is unknown (for the purpose of the estimation problem). Write

**Table 4.6: CSTR parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Nominal Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_A$</td>
<td>Concentration of A</td>
<td>$x_1$</td>
<td>$mol/m^3$</td>
</tr>
<tr>
<td>$T$</td>
<td>Temperature in CSTR</td>
<td>$x_2$</td>
<td>$K$</td>
</tr>
<tr>
<td>$T_c$</td>
<td>Temperature Cooling jacket</td>
<td>$u$</td>
<td>$K$</td>
</tr>
<tr>
<td>$q$</td>
<td>Flow rate</td>
<td>100</td>
<td>$m^3/sec$</td>
</tr>
<tr>
<td>$V$</td>
<td>Volume</td>
<td>100</td>
<td>$m^3$</td>
</tr>
<tr>
<td>$\rho$</td>
<td>Density</td>
<td>1000</td>
<td>$kg/m^3$</td>
</tr>
<tr>
<td>$C_p$</td>
<td>Heat capacity</td>
<td>0.239</td>
<td>$J/(Kg \cdot K)$</td>
</tr>
<tr>
<td>$\Delta H_R$</td>
<td>Heat of reaction</td>
<td>5e4</td>
<td>$J/mol$</td>
</tr>
<tr>
<td>$E/R$</td>
<td>Activation Energy over R</td>
<td>8750</td>
<td>$K$</td>
</tr>
<tr>
<td>$R$</td>
<td>Universal Gas Constant</td>
<td>8.31451</td>
<td>$(J/(mol \cdot K))$</td>
</tr>
<tr>
<td>$k_0$</td>
<td>Pre-exponential factor</td>
<td>7.2e10</td>
<td>$1/sec$</td>
</tr>
<tr>
<td>$U$</td>
<td>Overall Heat Transfer Coef</td>
<td>-</td>
<td>$W/(m^2 \cdot K)$</td>
</tr>
<tr>
<td>$A$</td>
<td>Area (specific for the U calculation)</td>
<td>-</td>
<td>$m^2$</td>
</tr>
<tr>
<td>$UA$</td>
<td>-</td>
<td>5e4</td>
<td>$W/K$</td>
</tr>
<tr>
<td>$C_{Af}$</td>
<td>Feed Concentration</td>
<td>1</td>
<td>$mol/m^3$</td>
</tr>
<tr>
<td>$T_f$</td>
<td>Feed Temperature</td>
<td>350</td>
<td>$K$</td>
</tr>
</tbody>
</table>

$\theta = E/R$ and adjoin the following equation:

$$\dot{\theta} = \dot{\omega}_3(t) \quad (4.44)$$

where $\dot{\omega}_3$ is another “white noise” source.

Assuming an integrate and reset anti-aliasing filter, the corresponding approximate discrete-time
model takes the form:

\[ x_{k+1} = f_d(x_k, u_k) + \tilde{\omega}_k \]  \hspace{1cm} (4.45)
\[ y_k = h(x_k, u_k) + \tilde{\nu}_k \]  \hspace{1cm} (4.46)

where \( h(x_k, u_k) = x_{2,k} \) and \( f_d(x_k, u_k) \) is given by

\[
\begin{bmatrix}
    x_{1,k} + \left( \frac{q}{V} \left( C_{Af} - x_{1,k} \right) - k_0 e^{-\theta_k/x_{2,k}x_{1,k}} \right) \Delta \\
    x_{2,k} + \left( \frac{q}{V} \left( T_f - x_{2,k} \right) + \frac{\Delta H_R}{\rho \cdot C_p} k_0 e^{-\theta_k/x_{2,k}x_{1,k}} + \frac{U \cdot A}{V \rho C_p} \left( u - x_{2,k} \right) \right) \Delta \\
    \theta_k
\end{bmatrix}
\]  \hspace{1cm} (4.47)

Simulation Results

Assume the system is in steady state with \( C_{Ass} = 0.87, T_{Ass} = 324.5 \) and constant input \( u_{ss} = 300 \). Then, at time \( t = 10 \) s, a step change \( u_{t=10} = 290 \) is applied. The simulated system response is as shown in Figure 4.21.

For the filtering problem, a sample period \( \Delta = 0.1 \) is chosen. Consider a multivariate Gaussian distribution for the initial state with mean \( \bar{x}_0 = E\{x_0\} = [0.87; 324.5; 8950] \) and variance

\[
E\{(x_0 - \bar{x}_0)(x_0 - \bar{x}_0)^T\} = \begin{bmatrix}
5 \cdot 10^{-4} & 0 & 0 \\
0 & 0.1 & 0 \\
0 & 0 & 225000
\end{bmatrix}
\]  \hspace{1cm} (4.48)

The process noise is considered independently distributed having variance \( E\{\omega_1 \omega_1^T\} = 5 \cdot 10^{-4}, E\{\omega_2 \omega_2^T\} = 0.1, E\{\omega_3 \omega_3^T\} = 0.1 \) and measurement noise \( E\{\nu \nu^T\} = 10 \). The measured temperature is as shown in Figure 4.22. Note that one has a scalar measurement and 3 unknown states.

Comparing Figures 4.21 and 4.22 it is seen that the measurement noise is large.
Two versions of the MDF algorithm were tested with different levels of approximation (i.e. $N_x = 125$ and $N_x = 27$). In both cases, one chooses $N_w = 27$.

**Case 1: $N_x = 125$**

Figure 4.23 shows the expected value of the estimated states provided by the MDF algorithm. Observe that the filter is able to recover the constant parameter $\hat{\theta}_{ss} = 8735.7$ with an error of 0.17%. Note that, satisfactory estimates for the Concentration and Temperature of $A$ have been obtained. Also, the use of the scaling factor $W$ in Lloyd’s algorithm is crucial to overcome numerical issues when the estimated $\theta$ is constant.
Note that the algorithm gives an excellent result. Also, the computation effort is small with the relative small number of points \( N_x = 125 \). Indeed, the whole algorithm takes approximately 2 second per sample.

**Case 2: \( N_x = 27 \)**

In an effort to test the viability of the algorithm on very high dimensional problems, \( N_x \) is reduced to 27 to give a very sparse set of quantisation points. Now the algorithm takes only a few milliseconds per sample. The mean of the estimated states are now as shown in Figure 4.24. A slight degradation in performance is seen but, the results still seem extremely acceptable in practical terms.
4.7 Conclusion

In this chapter, the classic particle filter was described and compared against a novel breed of deterministic filters based on vector quantization named Minimum Distortion Filtering (MDF). It has been shown that such filters provide very good performance with a low number of grid points.

**Figure 4.23:** Mean values for estimated states $N_x = 125$
Moreover, the MDF filters have shown outstanding performance in low dimensional cases and in systems which have a high sampling rate. The versatility of the MDF filter has been shown in a variety of problems, from systems with non-uniform sampling rate to chemical processes.

**Figure 4.24**: Mean values for estimated states $N_x = 27$
Part II

Networked Control
OVERVIEW OF NETWORKED CONTROL

This chapter presents a brief introduction to networked control. There is a vast, and growing, literature on this topic. Here the focus is on particular aspects which are motivated by a practical problem in wideband-code-division-multiple-access (WCDMA) broadband telecommunications. Brief background to the WCDMA broadband telecommunication system is provided.

The remainder of the chapter is organized as follows. Section 5.1 introduces communication constraints in the context of control theory. Section 5.2 describes WCDMA broadband communications systems, which is an example where communication constraints are ubiquitous. Section 5.3 describes, in detail, a simulator of a WCDMA system. Finally, Section 5.4 concludes the chapter.

5.1 Control with Communication Constraints

Traditionally control theory has ignored the fact that the measurements and control signals must be sent (transmitted) from the “controller” (usually placed in a central location) to the “plant”. However, recent research has expanded the horizons of control theory to capture the additional ingredient of the impact of communication constraints on control. These developments, inter-alia, link control theory, communication systems and information theory. The general title “Networked Control” has been used to describe this area [12, 90, 18, 11, 126, 22, 56, 91, 42, 43, 47, 107, 109, 120, 116, 121]. This thesis focuses on a particular class of networked control problems. These problems are motivated, in part, by control problems arising in WCDMA broadband telecommunication systems. Hence, a brief introduction to these systems will be presented later in this chapter.
A major focus in the thesis will be that of quantization and data rate limits. The usual approach, in classic control theory, has been to deal with quantization as an undesirable effect and not really as part of systems, see [16]. On the other hand, information theory has always considered quantization as part of the system [33].

In the literature, e.g. [126] and the reference therein, researchers have adopted a source coding perspective. This approach allows a variable length output symbol (bits). In this context, results have been established relying on the average data-rate (or average bit-rate). However, in practice, many systems (including WCDMA) have fixed rather than variable data-rates. Hence, it is important to study systems addressing this subclass of systems. In addition, it is clear that, in general, bounded average data-rate does not translate into fixed instantaneous data-rates, see [33] [66].

5.2 WCDMA Broadband Communications

As mentioned above, the WCDMA system will be used as a key motivational driver for the research outlined here. Hence, a brief outline of the WCDMA system is provided below.

WCDMA is part of the third generation of universal mobile telecommunications systems (UMTS). Its development depends on the third generation partnership project (3GPP), which is integrated by standard-developing organizations from all regions of the world.

WCDMA has opened the door to a wide variety of services. For example, T.V. broadcasting, video calls, large data transmissions, etc., which have been of great success in recent years. Despite the deployment of new communication networks, known as long term evolution (4G/LTE), there is a long predicted life ahead for 3G systems. (Surprisingly the major growth area in mobile telecommunications are in global system for mobile (GSM) and 3G and are expected to remain so for the foreseeable future). The growing number of subscribers and the vast deployment of such networks across the world are important driving forces to keep focusing important research effort on this technology.

WCDMA systems have many layers that can be related to control problems. In particular, the inner...
5.2 WCDMA Broadband Communications

loop power control presents many challenges. For example, the communication channel suffers from very aggressive noise, generally described by a Rayleigh distribution. In addition, the control commands can only be sent by a communication channel limited to one bit per sample\(^2\). Also, time delays, power saturation and backward compatibility make this component of the system a quintessential example of how control problems arise in communication systems.

The work in this Thesis has been strongly motivated by a partnership with Ericsson AB, and as a result of this collaboration, there has been significant time invested into this practical problem. To begin with, at the early stages of the project, a simulator of the system has been developed in order to test new ideas. This simulator is described below\(^3\).

Mobile communication systems have three main components: (i) Users, (ii) Base Station and (iii) the Air interface. Each one of these components plays several key roles in the operation of the overall system. In this thesis, the principal focus is on the control loops formed between these elements. In particular, one can identify two main control loops.

First, the control loop composed of the scheduler and the Transport Format Combination (TFC) selection. In this loop, the scheduler plays the role of an observer and a controller. It is an observer because, based on the available measurements, it has to estimate the load in the system (load estimator) to be able to determine how much resources are available (power headroom). Then, acting as a controller, it decides how these resources should be utilized. Next, it sends the control commands (absolute grants) via a wireless link. The latter that is subject to disturbances (relative grants) and time delays.

When the command is received (serving grant) by the plant (User Equipment) it has to be processed. Here, the UE determines what power (TFC) to use to send the data it has. This decision is subject to two constraints: power saturation and amount of data available to be sent. For instance, a UE will not select a power that cannot be used, neither will it choose more power than needed to send the data it has. Note that in this case, power translates to the amount of data sent per sample. Once this selection is over, the UE sends the data and reports back the TFC selected and a request for more power using the (Happy Bit). This closes the loop back to the scheduler.

---

\(^2\)This is constrained by international agreement.

\(^3\)One of the motivations for including these details here is that it provides an overview of the WCDMA system.
As can be inferred from the above description, this loop needs one extra component in the simulator to generate the data to be sent (Traffic model). The traffic model is in charge of simulating data in a buffer for each user, and once the buffer is emptied it replenishes it. In addition, the traffic model is also used to generate random calls in the system. These calls do not represent data in the system, but are an extra source of load for the system.

A second control loop of importance is the inner power control loop. This loop is a major focus of attention in the current Thesis because of its crucial impact on WCDMA performance. Indeed, performance, quality of service and efficiency depend critically on the performance of this loop. For example, if a user utilizes lower power than needed it will never be able to send anything. On the contrary, if too much power is used then no other user in the network will be able to perform correctly due to excessive interference.

In the inner power control loop, the Base Station sends control commands (TPC) to increase or decrease the power utilized by the User Equipment. Then, the user decodes the control signal and updates the power that it is using. This new power is measured at the Base Station after suffering interference and path loss, resulting in a measure of Signal to Interference Ratio (SIR). This measured SIR is then compared to some reference calculated by another layer of the system not described here.

Finally, note that all the signals sent in the system are digital signals therefore there is a finite codebook from which they can take values. In control theory, this is typically thought of as quantization. Quantization plays an important role in the work developed here and it will be appropriately described and addressed when necessary.

5.3 Simulator

This section describes a simple simulator of the uplink in WCDMA systems built by the author under guidance from Ericsson AB in Stockholm. The system has been built using a mixture of simulink and Mfunction. The simulink interface is used to provide a simple environment to measure and connect different components of the system. The main components of the simulator are:
5.3 Simulator

- Inner loop Power Control
- Scheduler
- Load estimator
- Traffic model
- Transport Format Combination (TFC) selection

The system can manage as many users as necessary. (It has been extensively tested with up to 25 users.)

In the network there exist two classes of traffic, namely voice and data. Voice traffic generates load in the system, however is not relevant to the decisions in the scheduler. Data users generate more load because they use more data channels. They are the users of prime interest in the current simulator. Also, note that each signal sent over the air interface belongs to a particular communication channel. There are four channels in the system:

- Dedicated Physical Control Channel (DPCCH): This is the channel measured by the inner loop power control loop. Also, the TPC commands are sent on this channel. Every user has this channel active.

- Dedicated Physical Data Channel (DPDCH): This channel is used for the voice calls. No signal of interest is transmitted on this channel. Non data users have this channel active when calls are in progress.

- Enhanced-Dedicated Physical Control Channel (E-DPCCH): This channel is used in the scheduler control loop. It carries the information reported back to the scheduler. Every user who is trying to send data has this channel active

- Enhanced-Dedicated Physical Data Channel (E-DPDCH): This channel is used to send the data. In the system, this channel is the result of the TFC selection process. This channel is active when a user is sending data.
Note that each one of these channels use part of the power available to the user and therefore generates load\textsuperscript{4} in the system.

5.3.1 Notation

The character $P$ denotes power in dBm, $\bar{P}$ denotes power on a linear scale ($T_x$ or $P'_x$ for the transmission power and $R_x$ for the received power in the Base Station). In addition, a subscript $x$ (ed, ec or c for the E-DPDCH, E-DPCCH and DPCCH) is used to denote to which channel the symbol refers to. The character $g$ and $I$ denote the channel gain and the interference, i.e. the symbol $T_xP_{ed}gI$ means: transmitted power of the E-DPDCH after the communication channel, including the effect of interference (i.e., the received SIR for channel x). The rest of the notation is intuitive, however it will be clarified if needed.

5.3.2 Inner loop

The inner power control loop represents how the UE behaves to control the power necessary for uplink transmission. The reference SIR target in the loop is given by the outer power control loop. The inner power control loop has a sampling period of one $TTI = 2/3[ms]$. The simulator block is shown in Figure 5.1.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{inner_loop.png}
\caption{Inner loop}
\end{figure}

\textsuperscript{4}"Load" is a term used to describe the total interference.
5.3 Simulator

**Inputs and outputs**

In the inner loop power control block the inputs are:

- Input 1: Interference from other UE’s
- Input 2: Channel gain model
- Input 3: SIR target from the Outer loop power control

The outputs are:

- Output 1: The Transmission power used on the control channel (DPCCH) of the UE before the channel and interference $T_x P_c$
- Output 2: Measured SIR in the control loop.
- Output 3: The Transmission power used on the control channel of the UE after the channel and before interference $T_x P_c g$

Output 1 is used to compute the transmission power that the UE will use for the E-DPDCH. (Note that this is not the received power in the Base Station.) Output 2 is used in the feedback calculation of the inner loop power control algorithm. Finally, output 3 is used to compute the interference that each mobile produces in the Base Station.

**Algorithm**

The inner power control loop has a quantizer in the communication channel and is controlled, in current implementations, using a proportional controller\(^5\).

**Remark 5.3.1** The measured SIR is the result of the measurement after the channel, the interference and recovered using channelisation. The resulting measured SIR is 256 times the received

\(^5\) A proposed new control law will be presented later in Section 6
SIR of the DPCCH i.e.,

\[
SIR_{\text{Measured}} = 256 \times T_x P_c \times \bar{g}/I \\
= 10 \log(256) + T_x P_c g I \quad [dB]
\]

where \(T_x P_c g I\) is the transmitted power of the UE after the channel and interference.

### 5.3.3 About the UE

The maximum power available for transmission in the Ue is \(21 [dBm]\) and the computation of the total transmission power used is:

\[
P_{TX} [dBm] = \left( \frac{\beta_{E-DPCCH}}{\beta_{DPCCH}} \right)^2 \times ncodes + \left( \frac{\beta_{E-DPCCH}}{\beta_{DPCCH}} \right)^2 + \left( \frac{\beta_{DPDCH}}{\beta_{DPCCH}} \right)^2 + 1) T_x P_c [dBm]
\]

where \(\left( \frac{\beta_{E-DPCCH}}{\beta_{DPCCH}} \right)^2 = 1\) and \(\left( \frac{\beta_{DPDCH}}{\beta_{DPCCH}} \right)^2 = \left( \frac{15}{11} \right)^2\) are fixed by convention and known.

**Remark 5.3.2** Typically users do not talk and web-browse (for example) at the same time. Hence, when the enhanced channel is used, then \(\left( \frac{\beta_{DPDCH}}{\beta_{DPCCH}} \right)^2 = 0\).

### 5.3.4 Interference

The model is shown in figure [5.2](#).

![Figure 5.2: Interference](#)
5.3 Simulator

Inputs and outputs

The inputs of this function are:

- The current power ratio factor of each UE
- The thermal noise in the cell
- The transmitted power used by each UE in the DPCCH

Output:

- The interference per each UE on a linear scale (subsequently scaled into dBm)

Algorithm

The interference block computes the interference for the received power of each mobile \( i \) in the Base Station, including E-DCH and non E-DCH users. The appropriate equation is,

\[
\bar{I}_i = \sum_{k \neq i}^{\text{nusers}} \left( T_x P_c g \right)_k \left\{ 1 + \left( \frac{\beta_{DPDCH}}{\beta_{DPCCH}} \right)_k^2 + \left( \frac{\beta_{E-DPDCH}}{\beta_{DPDCH}} \right)_k^2 \right\} + \bar{N}_0 + ICN + \alpha \left( T_x P_c g \right)_i [mW]
\]

\[
I_i = 10 \log(\bar{I}_i) [dBm]
\]  

(5.4)

(5.5)

where \( N_0 = -105 [dBm] \), ICN: inter cell noise (around 1-2 [dB] over thermal noise) and \( \alpha = 0.3 \) is the non orthogonality factor that describes self-interference.
5.3.5 Absolute, relative and serving grants

The absolute grant is a number transmitted by the serving cell that can take values between 0 and 31. The power ratio factor for each index is shown in table 5.1. Note that the relationship between index and grant is highly non-linear.

The relative grant is a number transmitted from the non-serving cell (also can be sent from the serving cell but this is not the case in the current system). This grant can only modify the serving grant by one index, i.e. if the current serving grant is 20 the relative grant can transform it into 19.

The resulting serving grant in the UE has a different granularity than the one of the absolute grant. Therefore Table 5.2 is used to choose the power ratio for the UE.

Remark 5.3.3 The table can be found in the document 25.213 of the standard in the following link: http://www.3gpp.org/ftp/Specs/html-info/25-series.htm.

Remark 5.3.4 An absolute grant can be sent every 2[ms].

Remark 5.3.5 An relative grant can be only sent, every 10[ms].
Table 5.1: Absolute Grant table

<table>
<thead>
<tr>
<th>Index</th>
<th>$(\beta_{ed}/\beta_c)^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>INACTIVE</td>
</tr>
<tr>
<td>1</td>
<td>Zero Grant</td>
</tr>
<tr>
<td>2</td>
<td>$(7/15)^2$</td>
</tr>
<tr>
<td>3</td>
<td>$(11/15)^2$</td>
</tr>
<tr>
<td>4</td>
<td>$(15/15)^2$</td>
</tr>
<tr>
<td>5</td>
<td>$(19/15)^2$</td>
</tr>
<tr>
<td>6</td>
<td>$(24/15)^2$</td>
</tr>
<tr>
<td>7</td>
<td>$(27/15)^2$</td>
</tr>
<tr>
<td>8</td>
<td>$(30/15)^2$</td>
</tr>
<tr>
<td>9</td>
<td>$(34/15)^2$</td>
</tr>
<tr>
<td>10</td>
<td>$(38/15)^2$</td>
</tr>
<tr>
<td>11</td>
<td>$(42/15)^2$</td>
</tr>
<tr>
<td>12</td>
<td>$(47/15)^2$</td>
</tr>
<tr>
<td>13</td>
<td>$(53/15)^2$</td>
</tr>
<tr>
<td>14</td>
<td>$(60/15)^2$</td>
</tr>
<tr>
<td>15</td>
<td>$(67/15)^2$</td>
</tr>
<tr>
<td>16</td>
<td>$(75/15)^2$</td>
</tr>
<tr>
<td>17</td>
<td>$(84/15)^2$</td>
</tr>
<tr>
<td>18</td>
<td>$(95/15)^2$</td>
</tr>
<tr>
<td>19</td>
<td>$(106/15)^2$</td>
</tr>
<tr>
<td>20</td>
<td>$(119/15)^2$</td>
</tr>
<tr>
<td>21</td>
<td>$(134/15)^2$</td>
</tr>
<tr>
<td>22</td>
<td>$(150/15)^2$</td>
</tr>
<tr>
<td>23</td>
<td>$(168/15)^2$</td>
</tr>
<tr>
<td>24</td>
<td>$(119/15)^2 \times 4$</td>
</tr>
<tr>
<td>25</td>
<td>$(150/15)^2 \times 2$</td>
</tr>
<tr>
<td>26</td>
<td>$(119/15)^2 \times 4$</td>
</tr>
<tr>
<td>27</td>
<td>$(134/15)^2 \times 4$</td>
</tr>
<tr>
<td>28</td>
<td>$(150/15)^2 \times 4$</td>
</tr>
<tr>
<td>29</td>
<td>$(168/15)^2 \times 4$</td>
</tr>
<tr>
<td>30</td>
<td>$(150/15)^2 \times 6$</td>
</tr>
<tr>
<td>31</td>
<td>$(168/15)^2 \times 6$</td>
</tr>
</tbody>
</table>

Table 5.2: Serving Grant table

<table>
<thead>
<tr>
<th>Index</th>
<th>$(\beta_{ed}/\beta_c)^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$(5/15)^2$</td>
</tr>
<tr>
<td>1</td>
<td>$(6/15)^2$</td>
</tr>
<tr>
<td>2</td>
<td>$(7/15)^2$</td>
</tr>
<tr>
<td>3</td>
<td>$(8/15)^2$</td>
</tr>
<tr>
<td>4</td>
<td>$(9/15)^2$</td>
</tr>
<tr>
<td>5</td>
<td>$(11/15)^2$</td>
</tr>
<tr>
<td>6</td>
<td>$(12/15)^2$</td>
</tr>
<tr>
<td>7</td>
<td>$(13/15)^2$</td>
</tr>
<tr>
<td>8</td>
<td>$(15/15)^2$</td>
</tr>
<tr>
<td>9</td>
<td>$(17/15)^2$</td>
</tr>
<tr>
<td>10</td>
<td>$(19/15)^2$</td>
</tr>
<tr>
<td>11</td>
<td>$(21/15)^2$</td>
</tr>
<tr>
<td>12</td>
<td>$(24/15)^2$</td>
</tr>
<tr>
<td>13</td>
<td>$(27/15)^2$</td>
</tr>
<tr>
<td>14</td>
<td>$(30/15)^2$</td>
</tr>
<tr>
<td>15</td>
<td>$(34/15)^2$</td>
</tr>
<tr>
<td>16</td>
<td>$(38/15)^2$</td>
</tr>
<tr>
<td>17</td>
<td>$(42/15)^2$</td>
</tr>
<tr>
<td>18</td>
<td>$(47/15)^2$</td>
</tr>
<tr>
<td>19</td>
<td>$(53/15)^2$</td>
</tr>
<tr>
<td>20</td>
<td>$(60/15)^2$</td>
</tr>
<tr>
<td>21</td>
<td>$(67/15)^2$</td>
</tr>
<tr>
<td>22</td>
<td>$(75/15)^2$</td>
</tr>
<tr>
<td>23</td>
<td>$(84/15)^2$</td>
</tr>
<tr>
<td>24</td>
<td>$(95/15)^2$</td>
</tr>
<tr>
<td>25</td>
<td>$(106/15)^2$</td>
</tr>
<tr>
<td>26</td>
<td>$(119/15)^2$</td>
</tr>
<tr>
<td>27</td>
<td>$(134/15)^2$</td>
</tr>
<tr>
<td>28</td>
<td>$(150/15)^2$</td>
</tr>
<tr>
<td>29</td>
<td>$(168/15)^2$</td>
</tr>
<tr>
<td>30</td>
<td>$(150/15)^2 \times 4$</td>
</tr>
<tr>
<td>31</td>
<td>$(168/15)^2 \times 6$</td>
</tr>
</tbody>
</table>
5.3.6 UE TFC selection

The matlab blocks for the UE TFC selection and serving grant is shown in figures 5.3 and 5.4.

![Figure 5.3: Ue beta selection and serving grant](image)

![Figure 5.4: Sub block ue beta selection](image)

Note that the absolute grant sent by the scheduler has an approximate 4 TTI delay, i.e. 8 msec. The scheduler knows the value of this delay. This means that the scheduler can predict exactly when a new absolute grant will be received from the moment that is sent.
Inputs and Outputs

The inputs of the UE beta selection process are:

- The serving grant
- The buffer length
- The power of the DPCCH
- A vector used as a flag to indicate when the buffer is empty

The outputs are:

- The selected beta factor
- The Happy Bit
- The new state of the buffer
- The buffer empty vector

Algorithm

The first step (shown in Figure 5.3) is to choose the maximum power ratio given the serving grant and the appropriate own power limitation. Since the power limitation is \( \hat{P}_{UE} = 21[dBm] \) for each UE, then the maximum possible beta that each UE can use is given by,

\[
\left( \frac{\beta_{E-DPDCCH}}{\beta_{DPCCH}} \right)_{\text{max}} = \hat{\beta}_{\text{ed}} = \frac{\hat{P}_{UE}}{T_x P_c} - 1 - \left( \frac{\beta_{E-DPDCCH}}{\beta_{DPCCH}} \right)^2
\]  

(5.6)

The second step (shown in Figure 5.4) in this procedure is to examine the state of the current buffer size. The UE has to choose a transport format (TFC) that allows it to send integer multiples of the minimum (and fixed, by now) packet size of 336[bits], see Table 5.3. Therefore, it is necessary to
know how many packets are in the buffer. In addition, the UE will not select a TFC that allows it to send more packets than the ones it has in its own buffer. Then, the UE will select the TFC subject to the following:

### Table 5.3: number of packet v/s power ratio

<table>
<thead>
<tr>
<th>number of packet</th>
<th>power ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>(42/15)^2</td>
</tr>
<tr>
<td>2</td>
<td>(53/15)^2</td>
</tr>
<tr>
<td>3</td>
<td>(67/15)^2</td>
</tr>
<tr>
<td>4</td>
<td>(95/15)^2</td>
</tr>
<tr>
<td>5</td>
<td>(106/15)^2</td>
</tr>
<tr>
<td>6</td>
<td>(106/15)^2</td>
</tr>
<tr>
<td>7</td>
<td>(119/15)^2</td>
</tr>
<tr>
<td>8</td>
<td>(120/15)^2</td>
</tr>
<tr>
<td>9</td>
<td>(120/15)^2</td>
</tr>
<tr>
<td>10</td>
<td>(134/15)^2</td>
</tr>
<tr>
<td>11</td>
<td>(134/15)^2</td>
</tr>
<tr>
<td>12</td>
<td>(150/15)^2</td>
</tr>
<tr>
<td>13</td>
<td>(150/15)^2</td>
</tr>
<tr>
<td>14</td>
<td>(150/15)^2</td>
</tr>
<tr>
<td>15</td>
<td>(150/15)^2</td>
</tr>
<tr>
<td>16</td>
<td>(150/15)^2</td>
</tr>
<tr>
<td>17</td>
<td>(150/15)^2</td>
</tr>
<tr>
<td>18</td>
<td>(164/15)^2</td>
</tr>
<tr>
<td>19</td>
<td>(164/15)^2</td>
</tr>
<tr>
<td>20</td>
<td>(184/15)^2</td>
</tr>
<tr>
<td>21</td>
<td>(184/15)^2</td>
</tr>
<tr>
<td>22</td>
<td>(184/15)^2</td>
</tr>
<tr>
<td>23</td>
<td>(184/15)^2</td>
</tr>
<tr>
<td>24</td>
<td>(184/15)^2</td>
</tr>
<tr>
<td>25</td>
<td>(206/15)^2</td>
</tr>
<tr>
<td>26</td>
<td>(206/15)^2</td>
</tr>
<tr>
<td>27</td>
<td>(206/15)^2</td>
</tr>
<tr>
<td>28</td>
<td>(206/15)^2</td>
</tr>
<tr>
<td>29</td>
<td>(206/15)^2</td>
</tr>
<tr>
<td>30</td>
<td>(206/15)^2</td>
</tr>
<tr>
<td>31</td>
<td>(206/15)^2</td>
</tr>
<tr>
<td>32</td>
<td>(233/15)^2</td>
</tr>
<tr>
<td>33</td>
<td>(233/15)^2</td>
</tr>
<tr>
<td>34</td>
<td>(233/15)^2</td>
</tr>
</tbody>
</table>

### Table 5.4: Maximum number of packet per power ratio

<table>
<thead>
<tr>
<th>Power ratio</th>
<th>maximum number of packet</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>(42/15)^2</td>
</tr>
<tr>
<td>2</td>
<td>(53/15)^2</td>
</tr>
<tr>
<td>3</td>
<td>(67/15)^2</td>
</tr>
<tr>
<td>4</td>
<td>(95/15)^2</td>
</tr>
<tr>
<td>5</td>
<td>(106/15)^2</td>
</tr>
<tr>
<td>6</td>
<td>(119/15)^2</td>
</tr>
<tr>
<td>7</td>
<td>(120/15)^2</td>
</tr>
<tr>
<td>8</td>
<td>(134/15)^2</td>
</tr>
<tr>
<td>9</td>
<td>(150/15)^2</td>
</tr>
<tr>
<td>10</td>
<td>(150/15)^2</td>
</tr>
<tr>
<td>11</td>
<td>(150/15)^2</td>
</tr>
<tr>
<td>12</td>
<td>(164/15)^2</td>
</tr>
<tr>
<td>13</td>
<td>(184/15)^2</td>
</tr>
<tr>
<td>14</td>
<td>(206/15)^2</td>
</tr>
<tr>
<td>15</td>
<td>(206/15)^2</td>
</tr>
<tr>
<td>16</td>
<td>(233/15)^2</td>
</tr>
<tr>
<td>17</td>
<td>(233/15)^2</td>
</tr>
</tbody>
</table>
\[ \text{Power ratio} = \min \{\text{Maximum power ratio, power ratio due to number of packets}\} \quad (5.7) \]

If the selected power ratio is constrained due to the number of packets to be transmitted, then the number of packets transmitted in one TTI is the same number of packets as in the buffer. On the other hand, if the selected power ratio is constrained due to the maximum power ratio, then the number of packets sent is the maximum possible number of packets for the selected power ratio, see Table 5.4.

### 5.3.7 Traffic model

The traffic model generates different packet sizes to be transmitted per user. The corresponding Simulink file is separated in two blocks as shown in Figure 5.5.

![Figure 5.5: Traffic model](image)

**Figure 5.5: Traffic model**

**Inputs and outputs**

The inputs:

- The buffer status
A buffer flag vector indicating when a UE has an empty buffer

The outputs:

- The buffer new status
- The buffer flag vector reset to zero.

Algorithm

When the buffer flag vector is active one generates a random time for the arrival of a new-packet of data, see equation (5.8). Also, at what time this happened is recorded.

\[
t_{\text{new session}} = -\log_{10}(\text{rand}()) \times \text{readtime};
\]  

(5.8)

where \( \text{readtime} = 5[\text{s}] \).

Then, when the waiting time has finished, we generate a new packet, in bits. The following function sets the size of the new buffer size.

\[
\text{buffer} = 8 \times \text{round}(\min(\maxsize, 10^{\text{randn()}\times\sigma + \mu} + \text{minsize}))
\]  

(5.9)

where \( \maxsize = 200 \times 10^3 \), \( \text{minsize} = 1 \times 10^3 \), \( \sigma = 0.6145 \) and \( \mu = 3.605 \).

For DCH users

The traffic model for the DCH users has two timers. The first one is used to simulate the time needed to generate a new session and the second one is used to simulate the finish of the session. Both of these use equation (5.8) to generate each time.
5.3 Simulator

5.3.8 Scheduler

The scheduler is the engine behind any communication system. It determines who transmits and how it does it at any given time. There are many classes of scheduler, for example Round Robin, proportional fair, etc. [35].

The scheduler implemented in the current simulator works in the following way. First, it determines the current state of the system, that is, if there exist an overload. In order to do so, it uses the TFC of each user and the transmit power to compute the total received wideband power (RTWP) and the rise over thermal noise (RoT). When the RoT is above, or below, a given threshold it triggers the overload mechanism (or non-overload mechanism) in the system.

When in overload, the scheduler will disconnect users (assign zero absolute grant) in the following order:

1. A user with the highest grant is given a minimum grant.
2. Do step 1 until all users have minimum grant or until the overload is resolved. If not, move to step 3.
3. Give zero grant to users in the same priority order as in step 1 until the overload is resolved. If not, move to step 4.
4. Shut down the cell.

When the system is not in overload the scheduler works as follows:

1. Any user requesting a grant for the first time is given a minimum grant.
2. Check the inactivity timer of those users with an active grant who are not requesting more. Decrease the corresponding grants to a minimum grant and initiate disconnection timers.
3. Check disconnection timers of those users with minimum grant and not requesting power. Disconnect users as required.
4. A user in step 1 obtains the maximum possible increment (this value is upper bounded).
5. Increase the active grant of those users with active grant but still unhappy\textsuperscript{6}.

6. Increase again the grant of users in step 4 and 5 in the same priority order.

The above steps should be carried until the system reaches maximum load or until there is no possibility of increasing the grants further.

**Load Estimator**

There is one more element in the scheduler, namely the load estimator. This unit is in charge of estimating the load coming from neighbouring cells ($L_{other}$). The algorithm is relatively simple. When the scheduler measures the load, in the cell, it compares this measured value against an estimate based on the actual TFC received from each users and their powers. If the estimated value is larger than the measured value then the algorithm decreases the value of $L_{other}$ by a fixed increment. When the opposite happens, the algorithm increases the value of $L_{other}$ by the same fixed increment.

**5.3.9 Other Relationships Used in the Simulator**

The following relationships are also deployed in the simulator. SIR in the control channel:

\[
SIR_i = \frac{\bar{P}_{ci} \times 256}{\sum_{k=1}^{\text{numusers}} P_{ck}(1 + \beta_d + \beta_{ed} + \beta_{ec}) + \bar{N}o + ICN - (1 - \alpha)\bar{P}_{ci}(1 + \beta_d + \beta_{ed} + \beta_{ec})}
\]

ROT (Rise over thermal):

\[
Rot = \frac{\left(\sum_{k=1}^{\text{numusers}} P_{ck}(1 + \beta_d + \beta_{ed} + \beta_{ec})\right) + \bar{N}o + ICN}{\bar{N}o}
\]

RTWP (Received total wide-band power):

\[
RTWP = \left(\sum_{k=1}^{\text{numusers}} P_{ck}(1 + \beta_d + \beta_{ed} + \beta_{ec})\right) + \bar{N}o + ICN
\]

\textsuperscript{6}It would be desirable if the scheduler could know more about the buffer sizes relevant to each user. A new scheme for achieving this goal is described later in Chapter 8.
5.4 Conclusions

In this chapter, WCDMA systems have been explained with a focus on readers with a control systems background. Each component of this complex system has been described to a sufficient level and more details were given for those components relevant to the current thesis. Also, control elements have been identified in this system. Specifically, two control loops were made explicit and several issues related to networked control were explained.
6

ADAPTIVE QUANTIZER

It can be seen from the earlier analysis in Chapter 5 that Inner Loop Power Control is implemented with a the bit rate constraint. It is known\(^1\) that this is a major limiting factor in determining closed loop performance. This leads to the following question “If one limits oneself to one bit per sample, are there better ways of using this restriction than to deploy a simple nearest neighbour quantizer?”. This is the subject of the following chapter. It will be shown that significant performance improvements are possible by use of an adaptive quantizer. A major potential application of these ideas will be to the problems of inner loop power control. Hence, additional information about this loop will be provided in Section 6.1. Then Section 6.2 describes the application of non-linear MPC to make more effective use of the adaptive quantizer in the inner loop power control problem. Section 6.3 gives performance comparisons between the adaptive quantizer and the traditional inner loop power controller used in WCDMA systems. Section 6.4 studies the robustness of the proposed scheme to missing bits in the communication systems. Conclusions are drawn in Section 6.5.

6.1 Inner Loop Power Control

Here further information is provided about inner loop power control. The motivation for providing this greater detail is that the inner loop power control problem will subsequently be used to motivate some of the research results presented here.

As seen in Chapter 5, inner loop power control (also called fast closed loop power control) lies

\(^1\)Simulations presented later will confirm this claim.
at the core of WCDMA mobile communication systems. This loop adjusts the UE transmitter power in accordance with one or more Transmit Power Control (TPC) commands received in the downlink. The goal of this loop is to maintain the received uplink Signal-to-Interference Ratio (SIR) at a given SIR target.

The standard model considered for the UE is an integrator plus delay. Signals are expressed in the logarithmic domain so that gain and interference ratios appear in an additive fashion. The various components in the inner loop power control loop are illustrated in Figure 6.1.

A simple state space model for the UE part of the inner loop power control system is given by,

\[ x_{k+1} = x_k + u_{k-d} \]  \hspace{1cm} (6.1)
\[ y_k = x_k \]  \hspace{1cm} (6.2)

where \( x_k \in \mathbb{R}^1 \) denotes the state of the UE and the signal \( y_k \in \mathbb{R}^1 \) corresponds to the Transmitted Power ("Tx") of the UE. No error appears in (6.1) since DPCCH coding is relatively strong\(^2\).

In Figure 6.1, \( g \) denotes fading of one form or another. Fading may be due to multipath propagation or to shadowing, and is sometimes modeled as a random process having a Rayleigh distribution

\(^2\)In other words, we assume no lost bits. Later, the impact of lost bits on performance will be explored - see Section 6.4.
Fading can, at times, result in very deep dips in power (over 10 dB).

In Figure 6.1 “I” denotes interference. Interference arises from three different sources: neighboring cells, all other users within the same cell and self-interference. (Self-interference can arise due to RAKE and G-RAKE finger misplacement [21, 49].)

The controller, located at the BS in the serving cell, estimates SIR of the received uplink DPCCH and generates TPC commands \( TPC_{cmd} \). In standard implementations, the controller at the BS, issues commands according to the following rule: if \( SIR_{est} > SIR_{target} \) then the TPC command transmitted is "0" (−1 dB power step), while if \( SIR_{est} < SIR_{target} \) then the TPC command is "1" (+1 dB power step).

The standard control law can be thought of as issuing power increments \( u_k \) of the form

\[
u_k = \text{sign}(y_k + g_k - I_k).
\]

where \( y_k + g_k - I_k \) corresponds to \( SIR_{est} \), i.e. the estimated SIR at the BS.

The dynamic range of uplink power control can be of the order of tens of dB. In this context, the standard bit quantization in the (DPCCH) of ±1dB used for power control is a major restriction. As a result, there has been substantial interest during the past two decades in different mechanisms to compensate for the ±1dB constraint. Recurring themes have been to use either advanced control methods, such as minimum variance control, with a simple decoder or some form of adaptive step size decoder with a simple control law, e.g, see [80, 38, 4, 88, 89, 113, 112]. The issue of time delay compensation has been addressed in [68]. Several authors, e.g., in [117, 29], have studied adaptive minimum variance controllers. In [11] a three degree of freedom controller was suggested. In all of this past literature the principal focus has been on the encoders and decoders of the control signal. For further background in power control problems see [102, 103, 104].

The content of this chapter is subject to intellectual property issues. Part of this chapter has been patented [59].
6.2 An MPC Based Technique for Channels with Limited Data Rate

In this section, the idea is to fix a nonlinear decoder in the UE and then redesign the controller to take account of the modified decoder. The decoder considered here is dynamic and uses a scaling technique to deal with the trade-off between dynamic range and idling errors resulting from the ±1dB bit limitation. The key contribution is to utilize a special form of Model Predictive Control (MPC) [97] in the Base Station to best capitalize on the scaling decoder.

6.2.1 Model Predictive Control

Here, a very brief introduction to MPC is given. (See [97, 60] for further details of nonlinear MPC).

Consider a time-invariant discrete-time model of the form:

\[ x_{k+1} = f(x_k, u_k), \]
\[ y_k = h(x_k), \]

where \( k \) is the discrete-time index, \( x_k \in \mathbb{R}^n \) denotes the state at time \( k \), \( u_k \in \mathbb{R}^m \) denotes the input and \( y_k \in \mathbb{R}^p \) denotes the measured outputs. The function \( f(\cdot) : \mathbb{R}^{n \times m} \to \mathbb{R}^n \) describes the state evolution and \( h(\cdot) : \mathbb{R}^n \to \mathbb{R}^p \) describes the output function.

An objective function is used to quantify the predicted future performance. The choice of the cost function plays a fundamental role in ensuring stability and performance of the system. A typical cost function takes the form:

\[ V_N(x_k, u_k) = \sum_{k=0}^{N-1} \left( \frac{1}{2} x_k^T Q x_k + \frac{1}{2} u_k^T R u_k \right) + \frac{1}{2} x_N^T P x_N, \]

where \( N \) is the optimization horizon, \( Q, R \) and \( P \) are tuning matrices with appropriate dimensions.

The MPC problem is formulated as finding the control sequence \( u^o = [u_0, u_1, \ldots, u_{N-1}] \) which
minimizes the cost function $V_N(\cdot)$, i.e.

$$u_0^* = \arg_{u_0} \min \{ V_N(x_k, u) \}$$

(6.7)

Then the control sequence $u_0 = (1, 0, \ldots, 0)$ is applied and one advances one time step. The optimization in (6.7) is carried out recursively subject to state and control constraints.

### 6.2.2 A New Paradigm for Inner Loop Power Control

In this approach, rather than fixing the controller and designing an encoder-decoder, one fixes the decoder in the UE as a nonlinear scheme with finite memory and then one redesigns the controller in the BS using nonlinear MPC. Thus, the controller will send the control signal which makes the best use of the decoder capability. The goal of this paradigm shift is to improve performance by allowing faster transient response and lower steady state errors. For simplicity of exposition, the focus is on a single UE. However, the methodology can be extended to multiple interacting UEs. Indeed, the simulations presented later, cover multiple UE’s with practical forms of fading.

**Nonlinear UE Decoder**

The first component in the mechanism is that the decoder in the UE includes a dynamic scaling mechanism. The Base Station can send only one $TPC_{cmd}$ per slot. In the basic WCDMA ILPC scheme used presently the UE interprets the control sequence as $\pm 1$ dB increments. This limits the rate of change of the transmit power. To overcome this constraint, the new adaptive decoder advocated here interprets two equal consecutive $TPC_{cmd}$ as $m$ times the previous value. That is, for the sequence $\{1, 1, 1\}$ the UE will increment its power by $\{1, 1 \cdot m, 1 \cdot m^2\}$ dB respectively. On the other hand, when differing $TPC_{cmd}$ are received, the UE will decrease its previous power step by $1/m$ and reverses the sign. That is, the sequence $\{1, 0, 1\}$ will cause the UE to use the power steps $\{1, -1/m, +1/m^2\}$ dB.

As explained in Section 6.1, similar decoding schemes have been proposed by other authors. The novel contribution made here is to change the control law to make best use of the zooming
quantizer. Note that the delay in (6.1) is unimportant in the optimization since it only represents an off-set in the control sequence [60]. Also, it is assumed, for the sake of simplicity of exposition, that future values of the interference and gain remain constant. (More sophisticated formulations would include a prediction for future values given the available observations [97, 60]). Under these conditions, one can simply redefine the origin as that state which achieves the required SIR for a given disturbance. Under these conditions, the state equation for the new model (see (6.4)) of the UE can be expressed as:

\[ x_{k+1}^{(1)} = x_k^{(1)} + x_k^{(2)}, \]
\[ x_{k+1}^{(2)} = b_k x_k^{(2)}, \]
\[ x_{k+1}^{(3)} = u_k, \]
\[ b_k = x_k^{(3)} u_k m^2 + 1 + \frac{m^2 - 1}{2m}. \]

Note that the computation of \( b_k \) is equivalent to:

\[ b_k = \begin{cases} 
  m & \text{if } u_k = x_k^{(3)}, \\
 -1/m & \text{if } u_k \neq x_k^{(3)}. 
\end{cases} \]

where \( x_k^{(1)} \) is the current transmitted power by the UE, \( x_k^{(2)} \) denotes the power increment at time \( k \) and \( x_k^{(3)} \) is the previous received TPC\(_{cmd}\). The factor \( b_k \) takes the values \(-1/m\) or \( m\) at each sample depending upon the current and previous control command. Thus \( b_k \) is \( m \) if the current TPC\(_{cmd}(u_k)\) is equal to the previous TPC\(_{cmd}(x_k^{(3)})\) and is \(-1/m\) when the current value for \( u_k \) is chosen different to \( x_k^{(3)}\).

Note that the model given by (6.8), (6.9), (6.10) has two extra states. The state \( x_k^{(2)} \) is used to record the size of the power step currently being utilized and the state \( x_k^{(3)} \) is used to remember the previous TPC\(_{cmd}\).
6.2 An MPC Based Technique for Channels with Limited Data Rate

Cost Function

The cost function, (6.11), used in the MPC design is chosen to have the following specific form:

\[
V(x_k) = \sum_{i=0}^{N-1} \left\{ (x_{k+i}^{(1)})^2 + (x_{k+i}^{(2)})^2 + (x_{k+i}^{(3)})^2 \right\} + P(x_{k+N}^{(1)})^2
\]  

(6.11)

where \(N\) is the control horizon. Subject to \(u_k \in \{-1, 1\}\).

In the next subsection it is shown that the choice \(N = 1, P = m\) leads to a closed loop having bounded signals.

6.2.3 Properties of the Control Law

An important property of the scheme is as follows:

**Theorem 6.2.1** The choice \(N = 1, P = m\) ensures that, for constant interference and constant channel gain, the trajectories of the state are bounded.

**Proof.** Note that for \(P = m, N = 1:\)

\[
V(x_k) = (x_k^{(1)})^2 + (x_k^{(2)})^2 + (x_k^{(3)})^2 + P(x_k^{(1)} + x_k^{(2)})^2
\]

(6.12)

Hence,

\[
V(x_k) \geq \|x_k\|^2
\]

(6.13)

Consider the change of variables defined by \(\bar{u} = b_k(u_k).\) Then, \(\bar{u}\) can only take values from the set \(\bar{u} = \{-1/m; m\}\). It is readily seen that the change in the cost function is exactly determined
by:

\[
\Delta V(x, \bar{u}) = V(f(x_k, u_k)) - V(x_k) = 2x_k^{(1)}x_k^{(2)}(1 + m\bar{u}) + (x_k^{(2)})^2(\bar{u}^2 + 2m\bar{u} + m^2) \tag{6.14}
\]

Directly from the previous equation one sees that when one chooses the control input \(\bar{u} = -1/m\) (\(u_k = -u_{k-1}\)) one has that

\[
\Delta V(x_k, -1/m) = -(x_k^{(2)})^2 \cdot \frac{2}{m^2}(m^2 - 1/2m - 1/2) \tag{6.15}
\]

which is negative for all \(m > 1\). Moreover, due to the operation of the minimization in the optimization, one only chooses \(\bar{u} = m\) when the change in cost is less than when \(\bar{u} = -1/m\) is used. Hence, the optimal policy ensures that \(\Delta V(x_k, \bar{u})\) is negative for all time. Hence \(V(x_k) \leq V(x_0)\) for all \(k\). Hence using (7.16), one has that

\[
\|x_k\|^2 \leq V(x_k) \leq V(x_0) \tag{6.16} \tag{6.17}
\]

The result follows.

**Remark 6.2.1** It is worth noticing that the current scheme does not fit into the classical framework of nonlinear MPC (NMPC) \[5, 30, 45, 97\]. In NMPC the usual approach is to linearize the system once it enters a final region and then to find a control law which steers the states into an invariant set. In order to establish this, it is necessary that the errors introduced by the quantizer are bounded. However, due to the adaptive nature of the decoder, it is impossible to find a finite bound on the quantization error.

The result in Theorem 6.2.1 treats an idealized case. The examples presented later consider fading and other practical issues. For example, in the above result, it is allowed for the power increments to decrease to zero. In practice, it is desirable to introduce a limit on the zooming capability.
6.3 Performance Comparison

of the algorithm. The reason is that, although it seems desirable to reduce the power increments arbitrarily for a fixed set point and disturbances, this is undesirable if the disturbances and set point change since in the latter case it will take time to build the power increments back up. Hence, there is a practical trade-off between the steady state oscillation produced by the minimum allowable power increment and the time needed to build the power increments back up. In the simulations presented in the next section, the minimum power increment is chosen to be $|1/m|$.

6.3 Performance Comparison

Several examples are presented to illustrate the performance of the algorithm. First, the nominal case consisting of a single UE is addressed.

**Nominal Case: Horizon 1**

This case is used to illustrate the advantages of the scheme under the simplest scenario. In addition, the channel gain and interference are taken to be constant and no saturation levels are considered.

Figure 6.2 shows the performance of the MPC controller with horizon 1 and three different zooming factors $m = \{1, 2, 3\}$.

The factor $m = 1$ is equivalent to the case when the traditional controller plus a delay compensation mechanism is utilized [67]. To compensate for a change of 10 dB this configuration will require at least 10 slots with this control law. Moreover, in steady state, there will be an oscillation of magnitude $\pm 1$ dB around the target value. When the factor $m$ is increased it is seen that a significant improvement in performance occurs. For $m = 2$ the transient time is similar to $m = 1$ (i.e. about 10 slots) but the steady state oscillation is half that achieved with $m = 1$. The factor $m = 3$ reduces the transient time to 4-5 slots, which is less than half the transient with $m = 1$. Also, the oscillation in steady state has amplitude $1/3$ of that seen with a zooming factor of 1. Also, it is worth noticing that because of the choice of $P$ the cost function of the MPC algorithm decreases in all cases.
Figure 6.2: State regulation: upper plot: Cost function $V(x)$. middle plot: State evolution $x_k^{(1)}: T x$. lower plot: $SIR$
In the previous example, the performance of the proposed technique has been illustrated. The performance of the existing schemes with the addition of delay compensation [67] has been illustrated in Figure 6.2 since it corresponds to the choice $m = 1$. The case without delay compensation (as is usual in practice) is shown in Figure 6.3.

Comparing the results in Figure 6.3 with those in Figure 6.2 shows that the UE in the current scheme experience longer transient time and increased steady state oscillations. (The current scheme yields a $2\, dB$ oscillation whereas with the new scheme gives an oscillation of $1/(2m)$).
Figure 6.4: Upper plot: $T_x$ power. Middle plot: $SIR$ UE 1. Lower plot: $SIR$ UE 2. Rural fading.
6.3 Performance Comparison

Two UEs with Rural Fading

Here, the performance of the system when, two users are interacting within the same serving cell in the presence of typical fading is compared. The algorithms are tested using standard rural fading data supplied by Ericsson AB. This brings a new ingredient to the problem, namely interference between users while they each tries to maintain their own SIR targets. The results are compared with the new scheme and the current scheme.

Figure 6.4 shows outstanding performance for the new scheme. In this case, it is clear that the performance is dramatically improved by using the new zooming strategy. Oscillations in the system are dramatically decreased. The transmitted power transient of the UEs with the new scheme are significantly less than for the UEs in the current scheme.
6.4 Robustness

Even though the previous sections showed promising results in the nominal case, it is desirable to explore the performance in more realistic scenarios. Hence, in this section, the Nonlinear Decoder Model Predictive Control (NDMPC) performance is studied under non ideal conditions. In particular, the NDMPC performance in the presence of bit errors is examined. Also, different types of fading\(^3\) will be tested.

The remainder of the section is organized as follows. Section 6.4.1 explains the general framework for the simulations. In Section 6.4.2 the case when the BS does not know about bit errors is studied. Section 6.4.3 explains why it is desirable to have an upper bound for power increments. Section 6.4.4 shows the performance of all four configurations with fading and bit error rates of 5%. Section 6.4.5 summarizes results for all configurations with different bit error rates under transient conditions and in steady state. In Section 6.4.6 the same analysis as in Section 6.4.5 is shown but considering the case when the Base Station knows about the bit errors.

6.4.1 Experimental Setup

Four configurations are studied. Consider the NDMPC using a scaling (zooming) factor of \(m = 1, 2, 3\) and the conventional scheme. Each case is simulated with a single UE (to focus on the bit error problem exclusively). The bit errors are modelled using a uniformly distributed random process. Thus, one can simulate a percentage of bit errors. In particular, bit error rates of 0%, 5% and 10% are used. Finally, when a bit is lost (error) the UE will use the opposite command i.e., when the UE should do \(-1/m\) times its previous power increment it will use \(m\) times instead and vice-versa.

First, the case when the controller in the BS has no information about the errors is studied. Then, the case when the BS knows about errors is analysed and the performance is compared with previous results.

---

\(^3\)The fading used in the experiments is real fading supplied by Ericsson AB.
6.4 Robustness

6.4.2 Erroneous bits, no information in the Base Station

Here, the results obtained under three different scenarios are shown. First, nominal performance (system without fading and 0% of bit errors) is considered so as to provide a benchmark. Second, bit error rates are introduced. This leads to the idea of adding saturation to the power increments in the system. Finally, the performance of the system using an upper bound for power increments is studied.
Nominal Performance

Figure 6.5 shows the performance of the system under ideal conditions for 4 configurations namely (i) $m = 1$, (ii) $m = 2$, (iii) $m = 3$ and (iv) conventional scheme. Using the NDMPC scheme with a moderate zooming factor yields a large improvement when compared to the conventional system. Note that in Figure 6.5, the line corresponding to $m = 1$ (black) coincides with the conventional...
scheme up to time $k = 20$.

Figure 6.6 shows the square error of the tracking of $SIR_{target}$. Here, the conventional scheme has the worst performance compared to the NDMPC schemes. This is as expected since the NDMPC scheme can utilize a much smaller power increment when in steady state.

\[ \text{Figure 6.6: Square error } (SIR - SIR_{target})^2. \]
6.4.3 Upper bound for power increments

In this section, the desirability of having a limit on the maximum power increment that a UE can use is shown. Here, the probability of bit error is taken to be 10%.

![Graph 1](image1.png)

**Figure 6.7**: Performance for $m = 1, 2, 3$ and conventional scheme (legacy).

Figure 6.7 considers the case without power limits and shows that, when bit errors are present, large peaks in power can occur. Under these conditions, a bit error is never corrected and hence,
the controller loses synchronization.

Figure 6.8 confirms the conclusion above. Also, it is clear that the larger the scaling factor \( m \), the worse the problem becomes.

![Graph showing square error](image)

**Figure 6.8**: Square error \((SIR - SIR_{target})^2\).

One possible solution is to enforce an upper saturation level for the power increments. This helps to control the peaks in power and also allows the controller to re-establish synchronization with the UE. When the system is tracking a step change, it will request to increase the power constantly allowing the system to saturate in both, the controller and the UE. Also, when the system is in steady state, the controller will send alternate control commands that will allow the controller and the UE to saturate at their lower level.

Figure 6.9 shows that, after introducing an upper limit equal to \( m^2 \), the system peak power is much lower. (Compare with Figure 6.7).
6.4.4 Performance with bit errors

Here the performance of the system (including an upper bound for power increments of $|m^2|$ and lower bound of $1/m$) with no fading, rural fading and Rayleigh fading for different bit error rates are studied.

Figure 6.9: Performance for $m = 1, 2, 3$ and conventional scheme (legacy).
Figure 6.10: Performance for $m = 1, 2, 3$ and conventional scheme with probability of bit error 5%. No fading.

Figure 6.10 shows that, when $m = 3$, the system has larger power peaks. However, the frequency and amplitude decreases with smaller scaling factors. Also, the scaling factor of $m = 2$ and $m = 1$ perform better than the conventional scheme.

Figure 6.11 reinforces what is seen in figure 6.10.
Figure 6.11: Square error \((SIR - SIR_{\text{target}})^2\) with probability of bit error 5%.

No fading.

Rural Fading

Figure 6.12 shows that when \(m = 3\) the system has larger power peaks. Here, the scaling factors of \(m = 2\) and \(m = 1\), clearly, perform better than the conventional scheme. The conventional scheme signals is worse.

Figure 6.13 again shows the large peaks when \(m = 3\) is used.
6.4 Robustness

Figure 6.12: Performance for $m = 1, 2, 3$ and conventional scheme (legacy) with probability of bit error 5%. Rural fading.

Rayleigh Fading

Figure 6.14 shows the difference between the schemes under Rayleigh fading. The performance of all of the configurations look similar and there are almost no peaks in power due to bit loss. However, again $m = 3$ gives the worst performance. Figure 6.15 shows that the errors are similar between all of the other configurations.
Figure 6.13: Square error \((SIR - SIR_{target})^2\) with probability of bit error 5%.

6.4.5 Overall results

Here, the results for the four scenarios for different fading and bit error probabilities are summarized. The simulation length has been reduced to 50 samples so as to focus on the performance during transients.

Table 6.1 summarizes the results obtained for 3 different bit error rates with no fading. Mean Square error, is defined here to be the average value of the measured square error of the SIR for each configuration. In particular, configurations \(m = 2\) and \(m = 3\) performed exceptionally well compared with the conventional scheme, except for \(m = 3\) when the bit error probability was 10%. This result is encouraging and shows that NDMPC is reliable when no fading is present.

<table>
<thead>
<tr>
<th></th>
<th>(m = 1)</th>
<th>(m = 2)</th>
<th>(m = 3)</th>
<th>Conventional</th>
<th>Probability bit Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Square error</td>
<td>43.34</td>
<td>23.7</td>
<td>23.80</td>
<td>43.64</td>
<td>0%</td>
</tr>
<tr>
<td>Mean Square error</td>
<td>57.11</td>
<td>23.73</td>
<td>23.83</td>
<td>43.84</td>
<td>5%</td>
</tr>
<tr>
<td>Mean Square error</td>
<td>53.91</td>
<td>24.43</td>
<td>95.68</td>
<td>54.6784</td>
<td>10%</td>
</tr>
</tbody>
</table>

Table 6.2 considers rural funding and shows that \(m = 2\) provides excellent performance which is
Figure 6.14: Performance for $m = 1, 2, 3$ and conventional scheme (legacy) with probability of bit error 5%. Rayleigh fading.

significantly better than the performance achieved by the conventional scheme.

Table 6.3 shows that Rayleigh fading has a significant impact on the performance of all of the configurations.
Figure 6.15: Square error \((SIR - SIR_{target})^2\) with probability of bit error 5%. Rayleigh fading.

Table 6.2: Comparison with rural fading

<table>
<thead>
<tr>
<th></th>
<th>(m = 1)</th>
<th>(m = 2)</th>
<th>(m = 3)</th>
<th>Conventional</th>
<th>Probability bit Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Square error</td>
<td>52.70</td>
<td>27.19</td>
<td>22.58</td>
<td>53.02</td>
<td>0%</td>
</tr>
<tr>
<td>Mean Square error</td>
<td>82.98</td>
<td>40.90</td>
<td>25.70</td>
<td>58.84</td>
<td>5%</td>
</tr>
<tr>
<td>Mean Square error</td>
<td>117.53</td>
<td>43.51</td>
<td>28.83</td>
<td>60.36</td>
<td>10%</td>
</tr>
</tbody>
</table>

Results in steady state

Here, the results using 1000 samples after the system has reached steady state are presented. This allows one to study the steady state performance of the NDMPC.

Table 6.4 shows that the NDMPC configuration performs better than the conventional configuration with no bit errors. With 5% bit errors the performance with \(m = 2\) is comparable to the conventional scheme.

Table 6.5 shows that, with rural fading, the NDMPC configuration with \(m = 2\) performs exceptionally well with no bit errors. However, when 5% errors occur then the performance is degraded but is still comparable with the conventional scheme.
6.4 Robustness

Table 6.3: Comparison with Rayleigh fading

<table>
<thead>
<tr>
<th>$m = 1$</th>
<th>$m = 2$</th>
<th>$m = 3$</th>
<th>Conventional</th>
<th>Probability bit Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Square error</td>
<td>67.81</td>
<td>30.98</td>
<td>26.73</td>
<td>68.02</td>
</tr>
<tr>
<td>Mean Square error</td>
<td>100.94</td>
<td>30.97</td>
<td>26.73</td>
<td>68.02</td>
</tr>
<tr>
<td>Mean Square error</td>
<td>83.89</td>
<td>68.73</td>
<td>37.09</td>
<td>78.00</td>
</tr>
</tbody>
</table>

Table 6.4: Comparison with no fading

<table>
<thead>
<tr>
<th>$m = 1$</th>
<th>$m = 2$</th>
<th>$m = 3$</th>
<th>Conventional</th>
<th>Probability bit Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Square error</td>
<td>0.36</td>
<td>0.41</td>
<td>0.11</td>
<td>0.98</td>
</tr>
<tr>
<td>Mean Square error</td>
<td>0.43</td>
<td>1.25</td>
<td>2.35</td>
<td>1.087</td>
</tr>
<tr>
<td>Mean Square error</td>
<td>0.99</td>
<td>3.43</td>
<td>17.25</td>
<td>1.25</td>
</tr>
</tbody>
</table>

Table 6.6 shows that Rayleigh fading is very detrimental to the performance of the NDMPC. The conventional scheme performance seems to be the best in this case. However, this is not surprising since the conventional scheme has very low capacity to react and compensate for fast fading and is thus insensitive to bit errors. This is another manifestation of robustness/performance trade-offs.
Table 6.5: Comparison with rural fading

<table>
<thead>
<tr>
<th>m = 1</th>
<th>m = 2</th>
<th>m = 3</th>
<th>Conventional</th>
<th>Probability bit Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Square error</td>
<td>0.35</td>
<td>0.23</td>
<td>0.19</td>
<td>0.99</td>
</tr>
<tr>
<td>Mean Square error</td>
<td>0.63</td>
<td>1.24</td>
<td>1.40</td>
<td>1.08</td>
</tr>
<tr>
<td>Mean Square error</td>
<td>0.90</td>
<td>4.48</td>
<td>10.42</td>
<td>1.13</td>
</tr>
</tbody>
</table>

Table 6.6: Comparison with Rayleigh fading

<table>
<thead>
<tr>
<th>m = 1</th>
<th>m = 2</th>
<th>m = 3</th>
<th>Conventional</th>
<th>Probability bit Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Square error</td>
<td>3.52</td>
<td>5.02</td>
<td>6.62</td>
<td>4.34</td>
</tr>
<tr>
<td>Mean Square error</td>
<td>6.40</td>
<td>8.18</td>
<td>10.17</td>
<td>5.47</td>
</tr>
<tr>
<td>Mean Square error</td>
<td>10.06</td>
<td>15.99</td>
<td>15.68</td>
<td>7.25</td>
</tr>
</tbody>
</table>

6.4.6 Erroneous bits with information at the Base Station

Here, the benefit of having information in the BS about bit errors but, of course, not in the UE is explored. Note that this would require transmission of 1 bit from the UE to BS so as to inform the BS of how the UE interpreted the power command.

Performance with bit errors in steady state

Table 6.7 shows that information at the BS regarding bit loss is very beneficial. The performance of the system is outstanding when one looks at the mean square error.

Table 6.7: Comparison with no fading

<table>
<thead>
<tr>
<th>m = 1</th>
<th>m = 2</th>
<th>m = 3</th>
<th>Conventional</th>
<th>Probability bit Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Square error</td>
<td>0.36</td>
<td>0.41</td>
<td>0.11</td>
<td>0.98</td>
</tr>
<tr>
<td>Mean Square error</td>
<td>0.50</td>
<td>0.42</td>
<td>0.42</td>
<td>1.16</td>
</tr>
<tr>
<td>Mean Square error</td>
<td>0.62</td>
<td>0.52</td>
<td>0.92</td>
<td>1.14</td>
</tr>
</tbody>
</table>

Table 6.8 shows that, in the presence of rural fading, knowledge of the bit errors at the BS is also largely beneficial. In this case, m = 2 shows an outstanding performance with very low mean square error even when the bit error rate is high.

Table 6.9 shows that in the presence of fast fading the performance of NDMPC configuration leads to very similar performance compared with the performance of the conventional scheme. This shows a big improvement relative to the case in Table 6.6.
Table 6.8: Comparison with rural fading

<table>
<thead>
<tr>
<th>$m = 1$</th>
<th>$m = 2$</th>
<th>$m = 3$</th>
<th>Conventional</th>
<th>Probability bit Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Square error</td>
<td>0.35</td>
<td>0.23</td>
<td>0.19</td>
<td>0.99</td>
</tr>
<tr>
<td>Mean Square error</td>
<td>0.46</td>
<td>0.36</td>
<td>0.42</td>
<td>1.13</td>
</tr>
<tr>
<td>Mean Square error</td>
<td>0.61</td>
<td>0.70</td>
<td>1.97</td>
<td>1.30</td>
</tr>
</tbody>
</table>

Table 6.9: Comparison with Rayleigh fading

<table>
<thead>
<tr>
<th>$m = 1$</th>
<th>$m = 2$</th>
<th>$m = 3$</th>
<th>Conventional</th>
<th>Probability bit Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Square error</td>
<td>3.52</td>
<td>5.02</td>
<td>6.62</td>
<td>4.34</td>
</tr>
<tr>
<td>Mean Square error</td>
<td>4.31</td>
<td>5.58</td>
<td>7.14</td>
<td>5.42</td>
</tr>
<tr>
<td>Mean Square error</td>
<td>7.05</td>
<td>7.81</td>
<td>10.89</td>
<td>7.85</td>
</tr>
</tbody>
</table>

6.4.7 Summary

To summarize, the performance achieved with the adaptive quantizer is usually not significantly worse and often much better than the conventional scheme. It has been shown that the scaling mechanism must have an upper bound (as well as a lower bound) for power increments. Otherwise, it is very hard to recover synchronization after a transmission error occurs. Finally, the simulations revealed that, for aggressive fading scenarios, it is helpful to inform the BS about errors in the transmission of the control commands.

Thus despite the initial recommendation of using $m = 3$, it was shown that $m = 2$ provides better overall results. This is not surprising because bit errors add uncertainty to the problem. Hence, $m = 3$ does not provide good robustness properties since, in some sense, is too aggressive. Whereas, $m = 2$ provides a good compromise between robustness and performance.

Second, to have an upper/lower bound for power increments is crucial. When a bit error occurs, the synchronization between Base Station (BS) and User Equipment (UE) is lost. This is a potential problem in regard to the power increment used e.g., with $m = 3$ the difference between power increments can go from 9dB to 27dB or even to 81dB. The saturation levels allow synchronization to be recovered when the system is in steady state and, because plus and minus ones are transmitted, or during transient when the UE is trying to reach a power level. In the former case, the lower limit aids synchronization. Whilst, in the latter case, the upper limit aids synchronization. In particular, a lower bound of $|1/m|$ and upper bound $|m^2|$ have been chosen, leading to four ab-
solute levels (positive and negative). Note that this is also helpful to handle soft-handover because, without initial synchronization, the system will need only about 4 to 5 samples to synchronize.

Finally, the NDMPC performs similarly to the current algorithm in steady state. However, it provides faster convergence speed under transient conditions.

6.5 Conclusions

This chapter has developed a new technique, called NDMPC, to be applied to systems with a fixed data rate. This technique, takes advantage of combining a fixed nonlinear decoder in the plant with an appropriate controller, based on MPC, to best capitalize on the nonlinear decoder. This novel algorithm exhibits good performance in practical examples. The scheme was first tested successfully on an idealized case of the inner loop power control problem in WCDMA. Also, the robustness of the technique was also analyzed when bit errors occurred. The results obtained show good performance. Some practical adjustments were added to ensure robust behaviour of the technique.

Motivated by this positive results, the NDMPC idea is extended to cover unstable systems in the next chapter.
An interesting research question arising from the results presented in the previous chapter is whether, or not, such a scheme would be useful to stabilize open loop unstable plants. (The WCDMA inner power loop is actually marginally stable). This is the question that is pursued in the current chapter.

The work here is based on the feedback system of Figure 7.1 which is a special type of Networked Controlled Architecture.

Many authors have studied necessary and sufficient conditions for stabilization of linear time-invariant unstable systems over bit rate constrained networked controlled architectures. In particular, it has been shown \[137, 133, 108, 72, 11, 12, 61, 106, 109, 91, 42\], under a variety of assumptions, that the required channel data rate must satisfy

\[ R \geq \sum_{\lambda_i \in U} \log_2(\lambda_i) \]  

(7.1)

where \( U \) denotes the set of unstable poles \( \{\lambda_i\} \). This constraint gives an upper bound on the location of the unstable pole consistent with closed loop stability. In the current context, this result means that with 1 bit one can stabilize systems having an unstable pole less or equal to 2. The main difference between the work reported here and the main body of work in the literature is that here a fixed bit rate rather than variable bit rate is considered.
All of the available results utilise a quantizer of some form or another and many of the results implicitly depend upon recursive optimal quantization of the initial state [109]. These results are conceptual in the sense that the schemes could not be used in practice since they depend on the application of a recursive optimal quantizer with infinite memory. Also, the control law is typically fixed to be a (variant of) a standard controller, e.g. linear quadratic optimal control [109].

Early work in this area [22, 91, 115] utilized an adaptive zoom-in and zoom-out strategy. The basic idea behind this class of strategies is that one expands the quantization step size until the state is captured and then one reduces the quantization step to give successively finer control as equilibrium is approached. Similar results hold for memoryless quantizers with no a-priori bound on the number of quantization levels [47].

Our proposed scheme differs in several respects as detailed below:

(i) The communication channel lies between the controller and the plant. (This restricts the set of allowable control commands.)

(ii) A fixed nonlinear decoder having one internal state is used. This decoder translates current and past commands into the plant input signal.

(iii) The controller is redesigned (using MPC) to make best use of the combination of the fixed nonlinear decoder and the plant.

(iv) A fixed bit rate of 1 bit/sample is used.

Note that the decoder is easy to implement in practice. Indeed, the algorithm described here is, a generalization of the one described in Sections 6.2, 6.3 and 6.4 to inner loop power control in WCDMA. Also, recall the algorithm in Section 6.1 combined several features including a fixed communication bit rate of 1 bit per sample, a nonlinear decoder and an MPC controller. Thus, the term “Nonlinear Decoder Model Predictive Control” was used for this class of strategies.
7.1 A Class of Open Loop Unstable Systems

Consider a networked control architecture\textsuperscript{1} as shown in Figure 7.1. The architecture differs from that normally considered in the networked control literature \cite{137, 133, 108, 72, 11, 12, 61, 108, 109, 91, 42} where the bit rate constraint is typically considered to be in the measurement channel. Here the communication link is in the control channel. Consider a single input plant having state space model of the form:

\[
\dot{x} = Ax + Bu
\]  

(7.2)

where \(A\) has one unstable pole (i.e. one eigenvalue in the right half plane). Assume that \(x\) is measured.

Note that one can apply a similarity transformation, as also used in \cite{109}, to write the system as

\[
\dot{x}' = \begin{bmatrix} \lambda & 0 \\ 0 & A' \end{bmatrix} x' + \begin{bmatrix} b_c \\ B' \end{bmatrix} u
\]  

(7.3)

where \(\lambda > 0\) and \(\text{det}(sI - A')\) is non zero for \(s\) in the closed right half plane.

The focus here is on the question of, whether or not, bounded trajectories exist for \(x\) and \(u\). Hence, one can limit the analysis to the open loop unstable mode \(x'_1\). The model for this mode satisfies

\[
\dot{x}'_1 = \lambda x'_1 + b_c u
\]  

(7.4)

\textsuperscript{1}Note that this is a generalization of the inner power loop shown in Figure 6.1.
The control law uses sampled data and issues one command per sample. Also, assume that the plant input is implemented via a zero order hold. Assume a sampling period of $\Delta$. In this case, the sampled values of $x'_1$ satisfy

$$x'_{1,k+1} = ax'_{1,k} + bu_k$$  \hspace{1cm} (7.5)

where $a = e^{\lambda \Delta}$ and $b = \frac{1}{\lambda} [e^{\lambda \Delta} - 1]b_c$. Note that since $\lambda > 0$ then $a > 1$.

The focus in this section will be on (a modified form of) the Nonlinear Decoder Model Predictive Control (NDMPC) algorithm described in detail in Section 7.2. This utilizes 1 bit per sample. Here, the focus is on what values of $a$ are associated with systems that can be stabilized by the NDMPC algorithm for a given sampling period $\Delta$. It is shown that there exists a maximal value of the discrete pole, $\alpha_n$, for systems that can be stabilized. Note that $\alpha_n$, $\lambda$ and $\Delta$ are related via $\alpha_n = e^{\lambda \Delta}$ or $\Delta = \log_e \alpha_n / \lambda$. Hence, for a given continuous time pole $\lambda$, the associated minimal bit rate (bits/second) will satisfy.

$$\text{minimal bit rate} = \text{bits/samp} \ast \text{samp/sec}$$  \hspace{1cm} (7.6)

$$= 1 \ast \frac{1}{\Delta}$$  \hspace{1cm} (7.7)

$$= \frac{\lambda}{\log_e \alpha_n}$$  \hspace{1cm} (7.8)

### 7.2 Nonlinear Decoder Model Predictive Controller (NDMPC)

The algorithm described in this section is unique in the sense that, instead of focusing on decoder/encoder schemes to deal with bit rate constraint, it focuses on optimizing the use of the decoder. One advantage of this scheme is that it allows stability conditions to be developed for fixed bit rate scenarios. The algorithm will include an upsampling strategy. However, before describing the upsampling strategy, first the state space model for the plant and nonlinear decoder are summarized.
7.2 Nonlinear Decoder Model Predictive Controller (NDMPC)

\[ x_{1,k+1} = ax_{1,k} + c(u_k)x_{2,k} \]  
\[ x_{2,k+1} = c(u_k)x_{2,k} \]  
\[ c(u_k) = \begin{cases} 
  m & \text{if } u_k = u_{k-1} \\
  -1/m & \text{if } u_k = -u_{k-1} 
\end{cases} \]

where \( u_k \in \{-1, 1\} \) and \( m \) is a fixed scaling factor. (Typically set to \( m=2 \).)

Note that there is a third implicit state that remembers \( u_{k-1} \). However, this state is always \( \pm 1 \) and therefore has no impact on our (practical) stability analysis.

The cost function used by the NDMPC algorithm is given by

\[ V(x_k, u_k, \ldots, u_{k+N-1}) = x_{1,k+N}^2 + x_{2,k+N}^2 \]

\[ := V_f(x_{k+N}) \]

Also denote

\[ V^*(x_k) = \min_{u_k, \ldots, u_{k+N-1}} V(x_k, u_k, \ldots, u_{k+N-1}) \]

Note that, in carrying out the above optimization, the control commands are planned over a horizon of \( N \) steps. The full \( N \) step control signal is implemented. (This contrasts with the usual rolling horizon update commonly used in MPC [60, 97].) Then design the next \( N \) step sequence over the modified sampling period \( \Delta' = N \Delta \). Note that the control signal is not constant on the period \( \Delta' \) but is composed of \( N \) piecewise constant segments. In this case, the period \( \Delta \) can be thought of as generating a form of upsampling. The above description can be summarized in the following state space model which applies at period \( \Delta' \). \(^3\)

---

\(^2\)Recall the discussion on choice of \( m \) in Chapter 6.

\(^3\)Without loss of generality \( b = 1 \) is fixed since it only represents a scaling of the input.
Lemma 7.2.1 \textit{The upsampled model obtained at sampling period }\Delta' \textit{is given by,}

\begin{align*}
x_{1,k+N} &= a^N x_{1,k} + \alpha x_{2,k} \quad (7.14) \\
x_{2,k+N} &= \beta x_{2,k} \quad (7.15)
\end{align*}

where

\begin{align*}
\alpha &= \sum_{i=0}^{N-1} a^i \prod_{j=0}^{i} c(u_{N-1-j}) \quad (7.16) \\
\beta &= \prod_{j=0}^{N-1} c(u_{N-1-j}) \quad (7.17)
\end{align*}

and \( c(u_k) \) as defined in (7.11).

\textbf{Proof.} \textit{The evolution of the state equation over two samples is}

\begin{align*}
x_{1,k+2} &= a(x_{1,k+1}) + c(u_{k+1})x_{2,k+1} \quad (7.18) \\
x_{2,k+2} &= c(u_{k+1})x_{2,k+1} \quad (7.19)
\end{align*}

Substituting (7.9), (7.10) into (7.18), (7.19):

\begin{align*}
x_{1,k+2} &= a(ax_{1,k} + c(u_k)x_{2,k}) \\
&\quad + c(u_{k+1})c(u_k)x_{2,k} \\
&= a^2 x_{1,k} + (ac(u_k) \\
&\quad + c(u_k)c(u_{k+1}))x_{2,k} \quad (7.20) \\
x_{2,k+2} &= c(u_{k+1})c(u_k)x_{2,k} \quad (7.21)
\end{align*}

where it is clear that if one iterates up to time \( k+N \) one obtains the desired result given by (7.14)-(7.17).

The stability properties of the above scheme are analysed in Section 7.4. However, to motivate the subsequent analysis, the performance of the NDMPC algorithm is first illustrated via an example.
7.3 Motivational Example

To motivate the subsequent analysis of the algorithm, a simulation result is presented. One extra caveat on the algorithm is that a lower limit, $s$, is enforced on the control signal that can be used. This limit is justified in the next section. The limit is assumed to be known to both, the MPC control law and to the decoder.

In the simulations, the scaling factor is fixed as $m = 2$. Also, the prediction horizon is chosen as $N = 7$. A typical trajectory beginning from the initial state $(1.2, 1)$ for $a = 1.5$, $s = 0.01$ and $N = 7$ is shown in Figure 7.2.

Note that the trajectory shown in Figure 7.2 does not converge to the origin but, instead, finally enters an invariant set. Here, the invariant set corresponds to the bold blue lines in the figure. Thus, there is a form of “practical stability” (see Definition 1 below) as opposed to “asymptotic stability”. This feature will also be reflected out in the analysis presented in the next section.

![Figure 7.2: Typical trajectory with $a = 1.5$, $m = 2$, $N = 7$ starting at the red dot.](image)

Many other cases have been studied and it has been found, empirically, that the NDMPC can
stabilize open loop unstable plants where the unstable pole lies between 1 and 1.5. With some extra effort, it has been possible to stabilize systems with pole up to 1.7. Note that the 1.7 result is near the upper bound of 2 given by (7.1).

7.4 Analysis

In the sequel, the following notion of practical stability is deployed.

**Definition 1 (Practical Stability)** The system \( x^+ = f(x, u) \), with control law \( u(x) \), is practically stable with region of attraction \( X_0 \) if and only if there exist regions \( X_e \subset X_f \subset X_0 \) such that:

(i) \( \forall x_0 \in X_0 \setminus X_e := \{ x_i : x_i \in X_0, x_i \notin X_e \} \), the cost increment is negative.

(ii) Once \( x_n \) enters \( X_f \), then \( x_m \in X_f, \forall m > n \).

(iii) Furthermore \( X_e \) is a rejection region which has the property that \( \forall x_k \in X_e \) then \( x_{k+N} \in X_f \).

The optimal cost function (7.13) is used as Lyapunov function. The conditions under which the cost function decreases for each sample period \( \Delta' = N \Delta \) are examined.

First, the following is established:

**Lemma 7.4.1** The difference in the cost function over period \( N \) satisfies

\[
V(x_0, u_0, \ldots, u_{N-1}) - V_f(x_0) = \\
= V_f(x_N) - V_f(x_0) \\
= (a^{2N} - 1)x_{1,0}^2 + 2a^N x_{1,0} \alpha x_{2,0} \\
+ (\beta^2 + \alpha^2 - 1)x_{2,0}^2
\]

\[
(7.23)
\]

\[4\]In the sequel, \( x^+ \) denotes \( x_{k+N} \) and \( f() \) is used as a general nonlinear function.

\[5\]Due to time invariance it is possible to set \( k = 0 \) without loss of generality.
Proof. Follows by substituting (7.14), (7.15), (7.16) and (7.17) into (7.12).

A consequence of Lemma 1 is that there exists a region in the state space in which it is impossible to reduce the cost for horizon $N$. (Heuristically, this region is where the initial control effort $(x_{2,0})$ is relatively small in comparison with the initial plant state $(x_{1,0})$. In this case, it is intuitively clear that one cannot reduce the cost even with the nonlinear decoder.) The region in which the cost cannot be reduced is quantified in the next result.

**Lemma 7.4.2** For horizon $N$, and for each choice of $\alpha$ and $\beta$ (7.16), (7.17) the cost cannot be reduced in the following regions:

For $x_2 > 0$:

\[
x_1 \leq \frac{-2\alpha a^N x_2 + |x_2| \Theta}{2(a^{2N} - 1)} \tag{7.24}
\]

and

\[
x_1 \geq \frac{-2\alpha a^N x_2 - |x_2| \Theta}{2(a^{2N} - 1)} \tag{7.25}
\]

For $x_2 < 0$:

\[
x_1 \geq \frac{-2\alpha a^N x_2 + |x_2| \Theta}{2(a^{2N} - 1)} \tag{7.26}
\]

and

\[
x_1 \leq \frac{-2\alpha a^N x_2 - |x_2| \Theta}{2(a^{2N} - 1)} \tag{7.27}
\]

where

\[
\Theta = \sqrt{4\alpha^2 a^{2N} - 4(a^{2N} - 1)(\beta^2 + \alpha^2 - 1)}
\]

Proof. Solving the quadratic inequality from Lemma 7.4.1 one finds that

\[
x_1 = \frac{-2\alpha a^N x_2 \pm |x_2| \Theta}{2(a^{2N} - 1)} \tag{7.28}
\]

Then, where $V_f(x_0) - V_f(x_N) \leq 0$ is determined, which leads to the results (7.24) to (7.27).

Note that $\alpha$ and $\beta$ are determined by the available options for the control commands over the optimization horizon $N$. Since 1 bit/sample is used, then there are exactly $2^N$ options. Taking
account of all possible commands, yields a region in which the cost cannot be reduced for all possible commands. This region is the intersection of the regions for each possible $\alpha$ and $\beta$. Moreover, the region where the cost can be reduced is the union of the complements of the regions given in (7.24) to (7.27).

The region in which the cost can be reduced is illustrated in Figure 7.3 for the case $m = 2$, $N = 2$ and $a = 1.1$ with no lower limit on the input signal.

![Figure 7.3](image)

**Figure 7.3**: Light green area: Region where the cost function can be decreased. Dark red area: Region where the cost cannot be decreased.

The exclusion regions (where the cost cannot be reduced) coincides with the intuition given after equation (7.23) i.e. where there is a small $x_{2,0}$ relative to $x_{1,0}$. This motivates one to put a lower limit, $s$, on the available input, i.e. a limit on the smallest value of $x_{2,0}$. Then one incorporates this limit into the nonlinear decoder, then the state space model for the composite system becomes:
\[ x_{1,k+1} = ax_{1,k} + \text{sat}(c(u_k)x_{2,k}) \] (7.29)
\[ x_{2,k+1} = \text{sat}(c(u_k)x_{2,k}) \] (7.30)

\[ \text{sat}(c(u_k)x_{2,k}) = \begin{cases} 
  mx_{2,k} & \text{if } c(u_k) = m \\ 
  -1/mx_{2,k} & \text{if } c(u_k)x_{2,k} \geq s \\ 
  s \cdot \text{sign}(c(u_k)x_{2,k}) & \text{if } c(u_k)x_{2,k} < s 
\end{cases} \] (7.31)

Given this modified system description, the region in which the cost function is able to be decreased over horizon \( N \) is examined. The result is shown in Figure 7.4, again for the case \( N = 2 \), \( m = 2 \), \( a = 1.1 \) and now for \( s = 0.01 \).

**Figure 7.4:** Light green area: Region where the cost function can be decreased. Dark red area: Region where the cost cannot be decreased. White area: unfeasible region.
Note that, when the system initial state lies in the region G (light Green) in Figure 7.5, then the trajectory can be “thrown” into the area R (dark Red). Empirically all of these trajectories are examined in order to define an invariant set, $\mathcal{X}_f$, having the property that once the trajectory enters this region, the trajectories do not leave it. This region is illustrated in Figure 7.5.

**Figure 7.5**: Outer circle $\mathcal{X}_0$: Radius 0.1433. Inner circle $\mathcal{X}_\epsilon$: Radius 0.0215. Middle circle $\mathcal{X}_f$: Radius 0.022. Red points inside $\mathcal{X}_\epsilon$ are mapped to the black points.

From Figure 7.5 one can conclude the following result regarding practical stability.

**Theorem 7.4.1** For any $x_0 \in \mathcal{X}_0$, then the system (7.14), (7.15), with $a = 1.1$, $N = 2$, $m = 2$, $s = 0.01$ is practically stable in the sense of Definition 1. The regions $\mathcal{X}_0$, $\mathcal{X}_f$ and $\mathcal{X}_\epsilon$ are circular regions with outer radius $r_0 = 0.1433$, $r_f = 0.022$ and $r_\epsilon = 0.0215$ respectively.

**Proof.** The cost function measures the distance of the states from the origin. Also the region inside $\mathcal{X}_0$ as having a circular boundary is defined. Hence, if the cost decreases within that region, then one can guarantee that the state will move closer to the origin and will ultimately enter a smaller circular region. Hence, it will eventually enter $\mathcal{X}_f$ where the same argument holds. However, when one enters $\mathcal{X}_\epsilon$ the worst case scenario is that it will kick the state back into $\mathcal{X}_f$. The radius...
of each region are directly determined from Figure 7.5.

In the next section the effect of varying the horizon $N$ is examined.

## 7.5 A Non-uniform Horizon Version of the NDMPC Algorithm

The results shown in the previous section are encouraging. They were based on $N = 2$. However, as $a$ increases, the regions where the system can be stabilized, for fixed horizon $N$, become smaller. This motivates a further modification to the algorithm so as to utilize a non-uniform upsampling horizon $N$.

The non-uniform horizon mechanism operates as a trial and error procedure: One begins with horizon $N = 1$ and examines if the cost function can be decreased. When this is not possible, the horizon is increased to $N = N + 1$ until one finds an optimization horizon $N$ that succeeds. In practice, it is necessary to place an upper limit on the horizon. Thus, define an arbitrary maximum horizon $\hat{N}$. The results shown below illustrate that this technique allows one to achieve practical stability when the unstable pole $a$ is closer to 2.

### 7.5.1 Examples

Here, the use of the non-uniform horizon version of the NDMPC algorithm is explored via simulations. The upper limit is set on the horizon as $\hat{N} = 10$.

First, the example in Section 7.3 with $a = 1.1$ is repeated. Figure 7.6 shows that, when the multi horizon version of the algorithm is utilized, then the size of the region of attraction grows dramatically and the size of the final invariant set is reduced. Specifically, for the non-uniform horizon algorithm with $\hat{N} = 10$, $s = 0.01$, the region of attraction has radius $r_0 = 4.58$ and the invariant set has radius $r_f = 0.0141$. If one increases the lower saturation limit, $s$, to 0.1, then $r_0$ grows to 45.8 and the invariant set radius goes to 0.14. Also, the region of attraction of the NDMPC with non-uniform horizon increases as the maximum horizon $\hat{N}$ increases. Similar results were found utilizing a system having an unstable pole at $a = 1.2$. An important point is that the longer horizons (near 10) are infrequently utilized. Indeed, their use is limited to the near
vicinity of the invariant set.

![Diagram](image_url)

**Figure 7.6**: Outer circle $X_0$: Radius 4.58. Inner circle $X_e$: Radius 0.01. Middle circle $X_f$: Radius 0.0141

Next, consider the more difficult case $a = 1.5$. For this case, using the non-uniform horizon NDMPC algorithm leads to the results in Figure 7.7. Figure 7.7 shows sets $X_0$, $X_f$ and $X_e$ as in the definition of practical stability. Even though these sets are relatively small, one needs to remember that the system is highly unstable ($a=1.5$) and that a fixed bit rate constraint holds. The crosses in the figure represent initial states involved in determining the regions.
7.6 Conclusions

In this chapter, a new technique, called NDMPC, has been developed and applied to unstable systems where the controller is implemented using a fixed data rate. As seen in the previous chapter, this technique takes advantage of combining a fixed nonlinear decoder in the plant with an appropriate controller based on MPC. Using the original NDMPC algorithm for unstable systems allows one to stabilize such systems. However, the stability regions were very conservative. This problem has been mitigated by introducing an adaptive horizon technique. This technique allows one to obtain larger stability regions, smaller final invariant sets and to reduce the computational time by not having to choose an excessively large horizon N all the time, but using a longer horizon only when needed.

Figure 7.7: Outer circle $X_0$: Radius 0.0310. Inner circle $X_\epsilon$: Radius 0.0101. Middle circle $X_f$: Radius 0.0246
The work described in Chapters 6 and 7 was primarily concerned with examining the impact of bit rate constraints on the implementation of control systems. The results have been shown to have direct relevance to practical systems including the inner loop power control loop in WCDMA telecommunication systems. Actually bit rate constraints are ubiquitous in telecommunication problems because the control and information signalling is performed over communication channels. This chapter addresses a different bit rate problem namely the use of the “Happy Bit” to indicate that a UE in WCDMA has a large file to send. This is an example of a bit rate constrained sensor since only 1 bit is being used to signal to the scheduler (in the Base Station) that the user has data to send. Obviously, using 1 bit (for the Happy Bit) sends only limited information. Motivated by the work in Chapters 6 and 7, this chapter explores ways of making best use of the one bit constraint in the Happy Bit to signal important facts about the amount of data that the user wants to send.

8.1 Overview of the Happy Bit

The Happy Bit is used to signal a request from a UE to the Base Station that it wishes to have additional power (i.e. a larger grant) to send data over the network. In this context, the Happy Bit is crucial to determine who should be allowed to transmit at a given time (and at what rate). In this chapter it is suggested that the Happy Bit can be utilized to transfer additional information. In particular, the Happy Bit can be used to estimate buffer sizes, when scheduling information is not available.
In the current implementation the Happy Bit is sent every TTI to the UE. This bit is set to “unhappy” when:

- The UE is transmitting as much scheduled data as allowed (E-TFC) by the current serving grant, and
- UE has enough power to transmit at a higher data rate, and
- the time it would take to empty the buffer is greater than $nTTI \times E - TFC$.

In the proposed implementation the Happy Bit can be used, not only for its original purpose, but also to estimate the buffer length in UE. Some advantages of having a more accurate buffer size estimate are:

1. It provides potential to better schedule the UEs. The availability of a better buffer size estimate potentially allows the scheduler to plan how a UE will empty its buffer. Hence, it should provide enhanced capability to finish and disconnect some UEs quickly.

2. It could help to classify UEs based on their buffer sizes and this respective capacity to send data. (This could provide more information to deal with bursty traffic).

3. If $nTTI$ is chosen appropriately, then undetected small buffers are of no importance since they will be emptied quickly anyway.

### 8.2 Using the Happy Bit for Buffer Size Estimation

The scheduler could perform more effectively if it has more information about the actual amount of data a UE wishes to send. Inherently, the Happy Bit conveys information regarding buffer size. However, it is very limited as outlined in section 5.2. Here, a novel mechanism is proposed to extract more information. In particular, the Happy Bit is used to provide an estimate of the buffer size ($\hat{B}$). The principle of this mechanism lies on the fact that the quantity $E - TFC$, used by the UE, is known in the Base Station, and that, the quantity $nTTI$ is set by the Base Station and is thus known once a UE is connected. Therefore, it is possible to estimate the minimum buffer ($\hat{B}$)
8.2 Using the Happy Bit for Buffer Size Estimation

in the UE by using

\[ \hat{B}_{k+1}^a = \hat{B}_k = nTTI \times E-TFC_k \]  

(8.1)

each time an unHappy Bit is received. When the user is happy the buffer estimate should be updated by

\[ \hat{B}_{k+1}^b = \hat{B}_k - E-TFC_k \]  

(8.2)

There is one more issue to be addressed. The E-TFC can change from one TTI to the next one. This means that, at a given time \( k \), \( \hat{B}_k \) can be significantly less than \( \hat{B}_{k-1} \). Therefore, one needs to add one extra step in the buffer estimation mechanism when the UE is unhappy, namely

\[ \hat{B}_{k+1} = \max(\hat{B}_{k+1}^a, \hat{B}_{k+1}^b) \]  

(8.3)

Note that, the time varying nature of the E-TFC allows for estimation of buffer sizes of different lengths. In particular, it seems that one can distinguish between large buffers in UEs with good channel conditions and small buffers in UEs with poor channel conditions.

In the next section, the performance of the estimation mechanism for short and long \( nTTI \) is studied.
8.3 Simulation Studies

8.3.1 Short number of TTI's

As shown in Figure 8.1, when nTTI is set to be small\(^1\), then large buffer sizes are hardly identifiable. Furthermore, there is little potential to utilize this information in a meaningful way in the scheduler.

\(^1\)As is typically in most operational WCDMA systems.
8.3 Simulation Studies

8.3.2 Long number of TTI

Figure 8.2 shows that when a longer nTTI is used then this helps better estimate the buffer size for the majority of users. However, some of the small buffers are missed, e.g. at $t = 11$, because, after the UE has requested a grant the UE becomes happy immediately upon receiving its first serving grant. On the other hand, small buffers of UEs with poor channel conditions are also identifiable,
e.g. at $t = 11.8$ or $t = 19$.

\begin{figure}
\centering
\includegraphics[width=\textwidth]{figure8.2}
\caption{Buffer Estimation $n_{TTI} = 50$}
\end{figure}

8.3.3 Recommendation

Based on the above analysis, it is recommended that the value $n_{TTI}$ used in setting the Happy Bit be set to a larger value so that it conveys more relevant information about buffer sizes to the Base Station.
8.4 Adaptive nTTI in Happy Bit

8.4.1 Motivation

It has been seen above that performance gains are achieved when the factor nTTI, used in setting the Happy Bit is set, to a value which lies somewhere in the “middle” of the anticipated range. This leads to a follow up research question namely, could one adaptively select nTTI so as to better signal the actual buffer length. This idea is explored below.

8.4.2 A Possible Algorithm

To design an adaptive strategy to change the nTTI factor is not straightforward because there are several factors influencing the decision. For example, at the beginning of the transmission of a packet, a UE can have a very low serving grant, (because the transmission is just starting or the channel conditions might be poor at that time) but, the serving grant could rapidly change in the next time slot. Therefore, one cannot be too aggressive in changing the nTTI factor if there are other factors changing in the system. Consequently, care must be exercised when adapting the nTTI variable. These conditions have lead to the algorithm given in Table 8.1.

<table>
<thead>
<tr>
<th>step</th>
<th>Do</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Verify that the E-TFC has not changed, i.e $E - TFC_{k-1} = E - TFC_k$. If it has not changed, go to 2, otherwise do not change nTTI.</td>
</tr>
<tr>
<td>2</td>
<td>If the UE is unhappy, then increase nTTI otherwise decrease nTTI</td>
</tr>
</tbody>
</table>

It will be verified below that this scheme leads to an enhanced buffer size estimate using the Happy Bit.

Before proceeding to the simulation results, the values used for the granularity of nTTI are as follows:
Table 8.2: nTTI granularity

<table>
<thead>
<tr>
<th>level</th>
<th>value (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>50</td>
</tr>
<tr>
<td>5</td>
<td>100</td>
</tr>
<tr>
<td>6</td>
<td>200</td>
</tr>
<tr>
<td>7</td>
<td>500</td>
</tr>
<tr>
<td>8</td>
<td>1000</td>
</tr>
</tbody>
</table>

8.4.3 Simulation Results

In this section, two cases are examined. First, the adaptive scheme is tested. Second, a fixed nTTI scheme is used to contrast the performance against the adaptive scheme.

![Buffer Estimation adaptive nTTI](image)

**Figure 8.3: Buffer Estimation adaptive nTTI**

(i) Performance achieved with the adaptive scheme: The adaptive scheme performance is shown in Figure 8.3. It can be seen that the performance of the adaptive nTTI technique is extremely
good. It can be seen from Figure 8.3, that the adaptive scheme is able to very accurately estimate the buffer size at different times regardless of the channel conditions or the length of the buffer. The scheme tracks the size of the buffer in real time as the buffer size changes.

**Figure 8.4**: Happy Bit
Figures 8.4, 8.5 and 8.6 show the variables involved in this system. Figure 8.4 shows the Happy Bit, which allows the scheme to check condition 2 in the algorithm in Table 8.1. Figure 8.5 shows the number of packets being sent. This is useful because if the same number of packets is sent then the same E-TFC is used, which allows one to check condition 1 in Table 8.1. Finally, Figure 8.6 illustrates the evolution of the nTTI chosen. As expected, it increases and decreases depending on the E-TFC being used and on the length of the buffer. In addition, Figure 8.6 shows that, when large buffers are present, the nTTI is adapted so that it varies in the 100 to 250 (ms) range. Conversely, in the presence of small buffers, the range of nTTI is adjusted automatically to lie between 25 to 50 (ms). This illustrates the adaptive nature of the scheme.

(ii) Performance achieved with a non-adaptive (fixed nTTI) scheme: Based on the results presented in section 8.3, it is clear that the performance of the fixed nTTI scheme will depend critically on exactly how the fixed nTTI is chosen. We see from the results for the adaptive scheme that the nTTI variable is, on average, chosen to be about 100 (ms). Thus, so as to give the fixed nTTI scheme its best chance, a value of either 50 and 100 (ms) were chosen for the fixed nTTI scheme.
Figure 8.7 and 8.2 respectively show the performance achieved when a fixed nTTI of 100 and 50 (ms) is used. As expected, based on the results in section 8.3, these values allow for relatively good buffer size estimation for large buffers although not as good as with the adaptive technique. However, the fixed nTTi scheme misses all of the small buffers. This is in sharp contrast to the adaptive technique which provides a good estimate for all scenarios. Furthermore, note that in Figure 8.7, the buffer was poorly estimated at around 10 seconds, due to poor channel conditions. This is despite the large value of nTTI chosen.
8.5 Conclusion

This chapter has shown how the Happy Bit can be utilized to convey information about buffer sizes. In particular, an adaptive technique has been developed to estimate buffer sizes. It has been shown that the adaptive scheme allows one to deal with problems in buffer size estimation arising from small buffers and poor channel conditions. In general, all the buffers are satisfactorily estimated and tracked accordingly. Finally, some synchronization is needed between the UE and Base Station since the latter needs to know the value of $n_{TTI}$ used by the UE to compute its Happy Bit.

Note that the work in this chapter has recently been prepared by Ericsson AB for submission as a provisional patent.
CONCLUSIONS

This thesis has been concerned with the impact of temporal and spatial quantization on the performance of estimation and control algorithms.

The thesis has been presented in two parts. Part one has addressed nonlinear filtering. Part two has addressed control problems in broadband telecommunication systems.

In nonlinear filtering, the emphasis has been on quantization issues in both the temporal and spatial domain. Upsampling, downsampling and event based sampling have been explored. Also, a novel deterministic filter based on vector quantization has been developed and successfully applied to a wide variety of problems.

In networked control, a strong link to WCDMA systems in mobile telecommunications has been established. It has been shown that WCDMA systems provide a quintessential example of networked control. Typical problems have been identified and successfully addressed by developing a novel class of control strategies. It has been shown that the combination of a nonlinear quantizer with MPC theory offers improved performance for inner loop power control. Performance and stability of the system have been studied. The algorithm has been extended, by the use of a variable horizon MPC scheme, to the stabilization of unstable systems.

9.1 Summary of Contributions

The principal contributions of each chapter are briefly summarized.
9.1.1 Chapter 2

Chapter 2 describes different classes of models. The links between continuous time systems and discrete time models have been addressed. Also, the topic of nonlinear filtering has been discussed and the main sources of errors have been identified. In summary, this chapter provides a basis for the subsequent treatment of nonlinear filtering in the thesis.

9.1.2 Chapter 3

Chapter 3 addresses the issue of temporal sampling in nonlinear filtering. New upsampling and downsampling methods have been developed. The novelty lies at a conceptual level and contributes to a better understanding of the impact of the sampling period on the filtering process. Also, the use of irregular sampling in nonlinear filtering has been studied. In particular, the application of Lebesgue sampling to nonlinear filtering has been developed.

9.1.3 Chapter 4

Chapter 4 addresses spatial sampling issues in a structured fashion. A novel breed of deterministic nonlinear filters have been developed. In particular, ideas from vector quantization and clustering were used to develop new filtering algorithms. Also, connections between spatial and temporal sampling period have been explored. The performance of the resultant, so called, Minimum Distortion Filters, has been studied by comparing it against other well known approaches to nonlinear filtering including particle filters.

9.1.4 Chapter 5

Chapter 5 provides a brief overview of networked control problems. Also, a practical example of a networked control system with a very restricted data rate is given. This system is the WCDMA broadband telecommunication network. This is used as motivation for the tools developed in the rest of the thesis. A model of a WCDMA network is presented, which is used to test the results later in the thesis.
9.2 Future Work

9.1.5 Chapter 6

A novel architecture is described, aimed at dealing with communication channels having a fixed (low) data rate. The architecture combines model predictive control with a fixed nonlinear quantizer. The new approach has been shown to lead to improvements compared with traditional techniques used for the inner loop power control problem. Transient times have been improved and steady state errors made smaller. Stability has also been studied and conditions for boundedness provided. The techniques develop in this chapter have lead to two patent applications in conjunction with Ericsson AB.

9.1.6 Chapter 7

Chapter 7 applies the NDMPC technique from Chapter 6 to unstable systems. Practical stability has been established for the scheme. Also, the basic scheme has been embellished by the use of a variable horizon length. This idea was shown to be useful in decreasing computational time whilst making the region of attraction larger and the final invariant set smaller.

9.1.7 Chapter 8

Chapter 8 describes a related problem in broadband telecommunications, namely that of signalling the buffer size of a user based on a one bit per sample communication channel. A novel adaptive scheme for signalling buffer size has been developed. This work is also the subject of a patent application in collaboration with Ericsson AB.

9.2 Future Work

The results of this thesis have potential to be extended in several directions:

Minimum Distortion Filters: It was shown that Minimum Distortion Filters give very good performance using very low computational resources compared with other techniques. Future work on further theoretical support for this class of filters would be interesting. More analysis of the convergence properties would also be interesting. Analysis of high dimensional cases would
be important since there is a vast literature on this topic across several fields.

**Upsampling and Downsampling:** These topics touch on deep theoretical issues in the field of the mathematics. Convergence is very interesting in this case. In the thesis, it has been shown that convergence to the continuous time probability density functions occurs. However the nonlinear case was not developed in full. In addition, the study of the exact numerical errors introduced by the downsampling technique is an interesting question worth exploring.

**Adaptive Quantizer:** The adaptive quantizer has been shown to be successful in improving performance in systems having a constrained communication channel. Stability analysis has been performed, however further exploration is needed. It is suspected that the stability properties can be further improved by using novel techniques from the nonlinear MPC literature. However, these techniques are not easily applied to this case due to the special structure of the adaptive quantizer.


