
by

Gholam Reza Danesfahani AMIEE

Thesis submitted to the University of Surrey for the degree of Doctor of Philosophy

Centre for Satellite Engineering Research
University of Surrey
Guildford, Surrey
United Kingdom

© Gholam Reza Danesfahani 1995
Summary

For mobile satellite communication systems substantial gains in performance can be achieved by having a multichannel demultiplexer demodulator (MCDD) on-board the satellite. Fundamental operations in the demodulator are the carrier and bit timing recovery. The aim of this project was to study ways to improve the performance of the MCDD. The results of this study are as follows:

1. An efficient optimum recursive (IIR) demultiplexer (DMUX) has been designed and evaluated for use in OBP satellites [Dan94a, Dan94b]. Comparisons with a typical non-recursive demultiplexer [Goc88, Qi 92] show that the IIR approach:
   - requires about 80% less arithmetic operations;
   - has about 50% less delay;
   - has a superior noise performance for 5 bits filter coefficient quantisation;
   - is more attractive for ASIC implementation;
   - enables more optimum spectral utilisation.

2. A survey of timing error detectors (TEDs) has led to a choice between the Gardner [Gar86] and the modified Mueller & Müller (mM&M) [Moe88] algorithms. Although the mM&M algorithm has a larger self-noise, it requires only one sample per symbol to operate. An improvement has been made to the mM&M TED to cancel the self noise. Due to this optimisation, the new algorithm has fewer cycle slips and acquires the timing error in approximately 15 symbols [Dan95].

3. A study of different frequency error detectors (FEDs) has lead to a new FED which works using preambles. A low hardware and algorithmic complexity, a fast acquisition time and a good noise performance in medium to high signal-to-noise ratio (SNR) are the features of the detector developed.

4. A survey of phase error detectors has lead to the choice of the decision directed maximum-likelihood feed forward algorithm because of its fast acquisition which is important in mobile applications [Jes91]. An optimisation has been made to this algorithm to improve its noise performance without increasing the acquisition time. The noise performance of the new algorithm approaches the Cramer-Rao bound at medium to high SNR.
Acknowledgements

I would like to express my gratitude to Professor B. G. Evans and Mr T. G. Jeans for supervising my Ph.D. programme. I am indebted to Mr. Jeans for his help and encouragement. His hard work in looking after my day-to-day progress is greatly appreciated.

I would also like to thank Dr M. J. Willis, Dr J. Ahmad, Dr S. Shin, Mr R. Qi, and Mr J. Paffett for various discussions on fading channels, synchronization, error correction coding, transmultiplexing, and signal quality estimation. I am also grateful to Mr. J. Hibbitt for introducing Emacs and many advanced features of \LaTeX to me.

My parents, Esmail and Jahan, my wife, Una, and my brother, Hamid Reza, truly sacrificed many things in life to pay my tuition fees and living expenses so that I might study for a Ph.D. at University of Surrey, England. As a token of appreciation, I would like to dedicate this thesis to them.
## Contents

Summary ........................................... i
Acknowledgements ................................ ii
Table of Contents ................................ iii
List of Figures ................................... viii
List of Tables ................................... xii
Notations ......................................... xiii

1 Introduction and Overview ........................ 1
   1.1 Historical Notes on Satellite Communications .... 1
       1.1.1 Early Experimental Satellites .................. 1
       1.1.2 International Telecommunications Satellites (Intelsat) ... 2
       1.1.3 International Maritime Satellites (Inmarsat) and Standards ... 3
       1.1.4 Motorola's Iridium System ...................... 5
   1.2 Baseband Switching — The First Step Towards Fully OBP Satellite ... 5
   1.3 Access Schemes for OBP Satellites ................ 7
   1.4 Demultiplexing Approaches for OBP Satellites .......... 8
       1.4.1 Direct Method Demultiplexing ................. 9
       1.4.2 Tree Demultiplexer .......................... 9
           1.4.2.1 FIR Prototype Filter .................. 10
           1.4.2.2 IIR Prototype Filter .................. 11
       1.4.3 Polyphase Discrete Fourier Transform (DFT) Structure ... 11
   1.5 Modulation Schemes for Satellite Communications ..... 12
       1.5.1 BPSK, QPSK, OQPSK ........................ 12
           1.5.1.1 Spectral Efficiency .................... 14
           1.5.1.2 Immunity to Nonlinearities ............. 14
           1.5.1.3 Implementation Complexity ............. 16
1.5.2 π/4-QPSK: QPSK without 180° Phase Transition .......................... 16
1.6 Digital Demodulator Synchronisation ............................................. 17
1.6.1 Comments ................................................................. 18
1.7 Simulation Tool .............................................................. 19
1.8 Objectives and Organisation of This Thesis .................................... 20

2 Demultiplexing For OBP Satellites ............................................... 23
2.1 A Review of Sampling Rate Conversion ....................................... 23
2.1.1 Interpolation by Integer Factors ......................................... 24
2.1.2 Decimation by Integer Factors ........................................... 26
2.1.3 Sampling Rate Conversion by Rational Factors ........................ 27
2.2 Digital Structures for Decimation and Interpolation ...................... 28
2.2.1 FIR Sampling Rate Convertors .......................................... 28
2.2.2 IIR Sampling Rate Convertors .......................................... 30
2.3 Demultiplexing Terminologies .................................................... 31
2.4 Demultiplexing by Direct Form ............................................... 31
2.5 Half-Band FIR Filter (HBF) .................................................... 33
2.5.1 Tradeoffs in using the FIR Tree Structure .............................. 34
2.6 Polyphase-Matrix Generalised DFT (GDFT) Structure .................... 35
2.6.1 Disadvantages of the Polyphase Matrix GDFT Structure .......... 39
2.7 Weighted Overlap-Add GDFT Structure .................................... 39
2.7.1 Disadvantages of the Weighted Overlap-Add Structure ............ 41
2.8 IIR Approach to Demultiplexing .............................................. 41
2.9 A Package for Designing IIR Demultiplexers ............................... 41
2.9.1 Ansari and Liu Algorithm ................................................ 42
2.9.1.1 Design Method ...................................................... 43
2.9.1.2 Phase Response and the Group Delay ............................ 45
2.9.2 Fletcher and Powell Algorithm .......................................... 46
2.9.2.1 Minimisation Procedure .......................................... 47
2.9.2.2 Convergence Test of The Rosenbrock Function ............... 49
2.9.2.3 Convergence Test of The Powell Function ..................... 49
2.9.2.4 Convergence Test of Quadratic Equations ..................... 50
2.9.2.5 Comments on the Convergence Tests ............................ 51
2.9.3 Realisation of the Designed Filters ..................................... 51
2.9.3.1 Realisation of First-Order Sections .............................. 52
2.9.3.2 Realisation of Second-Order Sections ............................ 53
2.9.4 Evaluation of Multiplication Roundoff Errors .............. 56
  2.9.4.1 First-Order Sections ......................................... 56
  2.9.4.2 Second-Order Section ...................................... 57
2.9.5 Software Implementation ........................................... 59
2.10 Summary ............................................................................ 60

3 Frequency Synchronisation Algorithms 61
  3.1 Features of Doppler Shift ............................................... 62
  3.2 Frequency Search Techniques ......................................... 66
  3.3 Frequency Synchronisation by PLLs ................................ 68
  3.4 Acquisition Aiding by AFC Loops .................................... 68
    3.4.1 Gardner Frequency Error Detector ............................. 69
      3.4.1.1 Frequency Error Detector S-Curve ....................... 71
    3.4.2 Dual Filter Discriminator (DFD) .............................. 73
    3.4.3 Balanced Quadricorrelator (BQC) ....................... 79
    3.4.4 Cross-Dot Product Frequency Error Detector ............. 81
  3.5 Feedforward Carrier Synchronisation ............................... 81
    3.5.1 Rotational Frequency Error Detectors ....................... 82
    3.5.2 A New Rotational FED .......................................... 84
    3.5.3 Frequency Error Detection by DFT Techniques ............ 87
  3.6 Noise Performance of the FB and FF FEDs ...................... 90
  3.7 Summary ............................................................................ 91

4 Symbol Timing Synchronisation Algorithms 93
  4.1 Signal Representation .................................................... 95
  4.2 Maximum-Likelihood (ML) Estimation .............................. 97
  4.3 Digital Timing Tracking Algorithms ............................... 101
    4.3.1 Tracking Performance ........................................... 102
  4.4 Maximum-Likelihood Tracking Algorithm ....................... 102
  4.5 Two-Point Difference Tracker ....................................... 107
  4.6 Gardner Algorithm ....................................................... 109
  4.7 Decision-Directed Minimum Mean-Square Error Algorithm ... 113
  4.8 Data Transition and Tracking Loop (DTTL) .................... 115
  4.9 Self-Noise Cancellation of the mM&M Algorithm ............. 117
  4.10 Timing Error Correction .............................................. 124
    4.10.1 Polyphase Filters .............................................. 124
5 Phase Synchronisation Algorithms
5.1 FF and FB Configurations ......................................................... 130
5.2 Decision-Directed ML Algorithm .............................................. 131
5.3 Decision-Directed Arctangent Algorithm ............................ 136
5.4 Non Data-Aided Algorithm ......................................................... 137
5.4.1 Acquisition Performance ..................................................... 139
5.5 Viterbi and Viterbi (V&V) Algorithm ........................................... 141
5.6 Optimisation of the Decision-Directed ML FF Algorithm .... 144
5.7 Noise Performance of Phase Synchronisers ...................... 148
5.8 Design of the Loop Filter .............................................................. 149
5.8.1 FF Loops ........................................................................ 149
5.8.2 FB Loops ........................................................................ 149
5.9 Summary ....................................................................................... 151

6 An Optimised 8-Channel Recursive (IIR) MCDD
6.1 Frequency Division Multiplexing ................................................ 155
6.1.1 FDM Rate .......................................................................... 156
6.2 Proposed Recursive (IIR) Tree Demultiplexer .................... 157
6.2.1 Design of the Prototype Recursive (IIR) Filter ................. 159
6.3 Nonrecursive (FIR) Tree Demultiplexer ............................. 160
6.3.1 Design of the Prototype Nonrecursive (FIR) Filter ......... 162
6.4 Simulation Results of Demultiplexing ................................ 163
6.5 Effect of Quantisation on the Performance of MCDDs .... 167
6.6 Comparison Between IIR and FIR Demultiplexers .......... 169
6.7 Phase Ambiguity .............................................................. 170
6.7.1 Frame Synchronisation ...................................................... 172
6.7.2 Bit Stuffing ..................................................................... 174
6.7.2.1 An Improved Bit Stuffing Technique ............................ 174
6.7.3 Detection of the Frame Overheads ........................................ 174
6.7.3.1 Simulation Results of Detecting the Frame Overheads .... 176
6.8 Simulation Results of Doppler Error Synchronisation .... 176
6.9 Joint Symbol Timing and Carrier Phase Synchronisation .... 179
List of Figures

1.1 Regenerative vs. transparent satellites. .......................................................... 6
1.2 The 3 principle multiple access schemes. .................................................... 7
1.3 Demultiplexing 4 FDM channel by a tree structure. ................................... 10
1.4 Polyphase DFT. ......................................................................................... 12
1.5 Generalised quadrature modulator (a) and demodulator (b). .................... 13
1.6 Scatter diagrams of BPSK and (O)QPSK. ................................................... 15
1.7 Scatter diagram of $\pi/4$-QPSK. .................................................................. 16

2.1 Block diagram of interpolation by a factor of $L$, and an illustration of the time domain representation of interpolation by a factor of $L = 3$. 25
2.2 Block diagram of decimation by a factor of $M$ and an illustration of the time domain representation of decimation by a factor of $M = 3$. 27
2.3 Sampling rate conversion by a rational factor $L/M$. ................................. 28
2.4 Direct implementation of FIR decimation filters. ....................................... 28
2.5 Sampling rate conversion by polyphase filters. ......................................... 30
2.6 Generalised direct per-channel approach to demultiplexing. ..................... 32
2.7 (a) Spectrum of the HBF, (b) Spectrum of the low-pass and the high-pass filters. ................................................................. 33
2.8 Polyphase matrix structure with $L = 4, M = 3, N = 47$. .......................... 36
2.9 Polyphase-matrix generalised discrete Fourier transform (GDFT) structure. ................................................................. 37
2.10 Weighted overlap-add GDFT structure. ................................................. 40
2.11 A simplified overview of the package developed. ................................... 42
2.12 The contour plot of the Rosenbrock function showing the search for the minimum. ................................................................. 49
2.13 The generalised first-order all-pass section. ............................................ 52
2.14 First-order all-pass networks. ................................................................. 54
2.15 The generalised second-order all-pass section. ........................................ 54
2.16 Second-order all-pass networks. ............................................................. 55
2.17 Noise model of first-order all-pass filters. ............................................. 56
2.18 Normalised output noise variance of first-order all-pass filters. ...... 58
2.19 Noise model of second-order all-pass filters. ..................................... 58
2.20 Output noise of the second-order sections. ......................................... 59
3.1 Instantaneous Doppler shift. ................................................................. 63
3.2 Rate of change of Doppler shift with time. ........................................... 63
3.3 Removing the predicted of the Doppler shift from the received signal. 65
3.4 Frequency error detection by parallel search. ..................................... 67
3.5 Use of the a second-order PLL for frequency acquisition. ............... 68
3.6 Gardner frequency discriminator and tracking loop. ......................... 69
3.7 PSD of the signal and frequency MFs. ................................................ 70
3.8 Variations in the Gardner ML FED S-curve with rolloff. ................. 72
3.9 Variations in the Gardner ML FED S-curve with timing error. ........ 72
3.10 Acquisition performance of the Gardner FED algorithm. .................. 74
3.11 Dual Filter Discriminator. ................................................................. 74
3.12 DFD spectra \( (f_s = 4R) \). ................................................................. 76
3.13 Simplified DFD. ................................................................................. 76
3.14 DFD characteristics. .......................................................................... 78
3.15 Discrete time equivalent of BQC. ....................................................... 80
3.16 Frequency acquisition with balanced quadricorrelator. ................. 80
3.17 Cross-dot product frequency error detector. ...................................... 81
3.18 S-curve of the CDP FED. .................................................................. 82
3.19 Feedforward frequency error correction. ......................................... 82
3.20 Block diagram of the feedforward frequency estimation structure. .... 83
3.21 Frequency error variance of the Classen et. al algorithm. .............. 84
3.22 Implementation of the proposed rotational FED. .............................. 85
3.23 Estimation performance of the proposed rotational FED. ............... 86
3.24 Acquisition performance of the sliding DFT FED. ......................... 88
3.25 Acquisition performance of the DFT FED. ....................................... 89
3.26 Noise performance of FB and FF FEDs. ......................................... 90
4.1 ISI free transmission of band-limited impulses. ................................. 96
4.2 Parallel search. .................................................................................. 99
4.3 Simulation results of the parallel search for QPSK with phase error of 40° ................................................. 99
4.4 The generic timing synchronisation loop. ......................... 101
4.5 The Kobayashi clock synchronisation algorithm. ......................... 103
4.6 Improved realisation of the Kobayashi clock synchroniser. .......... 104
4.7 Variations in $g(t)$ with time. ............................................. 105
4.8 Timing performance of the Kobayashi algorithm. ..................... 106
4.9 Normalised PSD at $fT = 0$ for DD ML tracking algorithm. ....... 107
4.10 Self-noise in two-point difference algorithm. ......................... 109
4.11 Performance of the Gardner algorithm. ................................. 110
4.12 Block diagram of the Gardner algorithm. ................................. 111
4.13 Performance of the Gardner algorithm. ................................. 112
4.14 Decision-directed MMSE algorithm. ...................................... 114
4.15 Data transition and tracking loop. ......................................... 115
4.16 Normalised PSD at $fT = 0$ for DTTL TED. ......................... 116
4.17 Acquisition and tracking performance of the DTTL TED. ......... 117
4.18 The mM&M clock synchroniser. ............................................. 119
4.19 Normalised PSD at $fT = 0$ for the mM&M TED. .................. 120
4.20 The block diagram of the optimised mM&M clock synchroniser. .. 122
4.21 Performance of the optimised mM&M clock synchroniser. .......... 123
4.22 Timing error correction by polyphase filters. ......................... 124

5.1 Simplified models of phase synchronisation loops. ................... 131
5.2 Decision-directed ML FB PED ............................................. 132
5.3 S-curve of the decision-directed ML PED. .............................. 133
5.4 Performance of the decision-directed ML PED. ....................... 135
5.5 decision-directed arctangent PED ......................................... 137
5.6 QPSK non data-aided PED. .................................................. 139
5.7 BER performance of the non data-aided algorithm. .................. 141
5.8 Viterbi & Viterbi FF PED .................................................... 142
5.9 Normalised PSD of the V & V algorithm. ................................. 143
5.10 Original decision-directed ML FF PED. .................................. 144
5.11 Proposed optimised decision-directed ML FF PED. ................. 145
5.12 Performance of the decision-directed ML FF PED algorithm before and after optimisation. ......................... 147
5.13 Normalised PSD of the optimised decision-directed ML FF, Costas, decision-directed arctangent, non-data aided ML, and V&V algorithms. .............................................................. 148
5.14 Second-order loop filter. ........................................................................ 150
5.15 Performance of the second-order loop with different $\xi$. ...................... 151

6.1 Frequency division multiplexing of $K$ channels. .................................... 155
6.2 FDM signal, $K = 8$. ........................................................................... 157
6.3 Block diagram of the 8-channel IIR tree demultiplexer ......................... 158
6.4 Theoretical spectra at different points of the 8-channel IIR tree demultiplexer .................................................................................. 159
6.5 Responses of the designed prototype IIR filter ..................................... 161
6.6 Block diagram of the 8-channel FIR tree demultiplexer ....................... 162
6.7 Theoretical spectrum at different points of the halfband FIR demultiplexer. .................................................................................. 163
6.8 Simulation results of demultiplexing the proposed IIR tree structure. 165
6.9 Simulation results of demultiplexing the FIR tree structure. ................. 166
6.10 Effect of quantisation on the performance of MCDDs. ......................... 168
6.11 Demonstration of the phase ambiguity .................................................... 171
6.12 Performance of a UW detector ............................................................. 173
6.13 Simulation results of the Doppler error synchronisation by the Alberty and Hespelt algorithm ....................................................... 177
6.14 Simulation results of the Doppler error synchronisation by the proposed algorithm. ................................................................. 178
6.15 Acquisition and tracking performance of the Gardner algorithm (GA) and the optimised mM&M algorithm. ............................. 181
6.16 Acquisition and tracking performance of the Costas loop and the proposed PED algorithm at $E_b / N_0 = 5$ dB. ......................... 182
6.17 Acquisition and tracking performance of the Costas loop and the proposed PED algorithm at $E_b / N_0 = 15$ dB. ......................... 183
6.18 BER performance of the the joint symbol timing and carrier phase synchronisation. ................................................................. 184
6.19 Performance of the optimised 8-channel recursive (IIR) MCDD at $E_b / N_0 = 6$ dB. ................................................................. 185
## List of Tables

2.1 Convergence test of the Rosenbrock function. .................................. 50  
2.2 Convergence test of a quadratic function. ....................................... 50  
2.3 Transfer function of first-order sections. ........................................ 57  
2.4 Noise variance of first-order filters. ............................................. 57  
2.5 Noise variance of second-order sections. ........................................ 59  
3.1 Theoretical S-curve of the Gardner FED. ........................................ 71  
3.2 Operation of the proposed heuristic FED. ....................................... 85  
3.3 Summary of the parameters used in simulating the (S)DFT FEDs. ........ 89  
4.1 Algorithmic complexity of the timing synchronisers surveyed. ........... 128  
5.1 Computational complexity of the V&V phase synchroniser. ................. 143  
5.2 Algorithmic complexity of the phase synchronisers surveyed. ............. 152  
6.1 Phase ambiguity in QPSK demodulator .......................................... 171  
6.2 Possibilities of detecting UWs by the receiver ................................ 175  
6.3 Simulation results of detecting the frame overhead .......................... 176
# Notations

The following notations have been frequently used in this thesis:

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>Rolloff factor</td>
</tr>
<tr>
<td>$A_0$</td>
<td>Amplitude of the received signal</td>
</tr>
<tr>
<td>A&amp;H</td>
<td>Alberty and Hespelt</td>
</tr>
<tr>
<td>A&amp;L</td>
<td>Ansari and Liu</td>
</tr>
<tr>
<td>AWGN</td>
<td>Additive white Gaussian noise</td>
</tr>
<tr>
<td>$\beta$</td>
<td>Closed-loop gain factor</td>
</tr>
<tr>
<td>BER</td>
<td>Bit error rate</td>
</tr>
<tr>
<td>$B_L$</td>
<td>One-sided closed-loop noise bandwidth</td>
</tr>
<tr>
<td>BPSK</td>
<td>Binary phase shift keying</td>
</tr>
<tr>
<td>$\hat{c}(n)$</td>
<td>Decision on the complex data $c(n)$</td>
</tr>
<tr>
<td>CRB</td>
<td>Cramer-Rao bound</td>
</tr>
<tr>
<td>DMUX</td>
<td>Demultiplexer</td>
</tr>
<tr>
<td>$f$</td>
<td>Frequency in Hz</td>
</tr>
<tr>
<td>FB</td>
<td>Feedback</td>
</tr>
<tr>
<td>FDM</td>
<td>Frequency division multiplexing</td>
</tr>
<tr>
<td>FED</td>
<td>Frequency error detector</td>
</tr>
<tr>
<td>FF</td>
<td>Feedforward</td>
</tr>
<tr>
<td>FIR</td>
<td>Finite impulse response</td>
</tr>
<tr>
<td>FOS</td>
<td>First-order section</td>
</tr>
<tr>
<td>$h(n)$</td>
<td>$n$-th coefficient of the prototype filter</td>
</tr>
<tr>
<td>ICO</td>
<td>Intermediate circular orbit</td>
</tr>
<tr>
<td>$\Im[x]$</td>
<td>Imaginary part of $x$</td>
</tr>
<tr>
<td>IIR</td>
<td>Infinite impulse response</td>
</tr>
<tr>
<td>$K$</td>
<td>Number of FDM channels</td>
</tr>
<tr>
<td>LEO</td>
<td>Low earth orbit</td>
</tr>
<tr>
<td>$M$</td>
<td>$M = 2$ for BPSK, and $M = 4$ for (O)QPSK</td>
</tr>
<tr>
<td>MCDD</td>
<td>Multicarrier demultiplexer demodulator</td>
</tr>
<tr>
<td>MF</td>
<td>Matched filter</td>
</tr>
<tr>
<td>ML</td>
<td>Maximum-likelihood</td>
</tr>
</tbody>
</table>
## Notation Definition

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_i$</td>
<td>Number of filter coefficients in the interpolator sub-filters</td>
</tr>
<tr>
<td>OBP</td>
<td>On-board processing</td>
</tr>
<tr>
<td>(O)QPSK</td>
<td>(Offset) quadrature phase shift keying</td>
</tr>
<tr>
<td>PED</td>
<td>Phase error detector</td>
</tr>
<tr>
<td>$p(n)$</td>
<td>$n$-th sampled matched filter output</td>
</tr>
<tr>
<td>$R$</td>
<td>Symbol rate</td>
</tr>
<tr>
<td>$\Re[x]$</td>
<td>Real part of $x$</td>
</tr>
<tr>
<td>$S(f)$</td>
<td>Power spectral density at frequency $f$</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal-to-noise ratio</td>
</tr>
<tr>
<td>SOS</td>
<td>Second-order section</td>
</tr>
<tr>
<td>SPS</td>
<td>Number of samples per symbol</td>
</tr>
<tr>
<td>$t$</td>
<td>Time in seconds</td>
</tr>
<tr>
<td>$T$</td>
<td>Symbol period ($= 1 / R$)</td>
</tr>
<tr>
<td>TED</td>
<td>Timing error detector</td>
</tr>
<tr>
<td>$T_s$</td>
<td>Sampling period ($= 1 / f_s$)</td>
</tr>
<tr>
<td>$\tau / T$</td>
<td>Normalised timing error</td>
</tr>
<tr>
<td>$u_q(n)$</td>
<td>$n$-th mean $\Psi$ error detector output</td>
</tr>
<tr>
<td>V&amp;V</td>
<td>Viterbi and Viterbi</td>
</tr>
<tr>
<td>$vT$</td>
<td>Normalised frequency error</td>
</tr>
<tr>
<td>$x^*$</td>
<td>Complex conjugate of $x$</td>
</tr>
<tr>
<td>$\hat{\Psi}$</td>
<td>Estimated value of $\Psi$</td>
</tr>
<tr>
<td>$\tilde{\Psi}$</td>
<td>Trial value of $\Psi$</td>
</tr>
<tr>
<td>—</td>
<td>Complex signal</td>
</tr>
<tr>
<td>—</td>
<td>Real signal</td>
</tr>
</tbody>
</table>
Chapter 1
Introduction and Overview

This chapter provides a brief history of satellite communication from the early experimental satellites to the proposed on-board processing Iridium and Inmarsat-P systems. The optimisation of demultiplexing and synchronisation algorithms for OBP satellites will be discussed. This chapter will end with a guide to the rest of the thesis.

It is over three decades since the first communication satellite was launched, and began the evolution of the global satellite communication of today. Since that launch many satellites have been put into orbit, each one developing new technologies. The uninterrupted services offered by satellites have become an integral part of our everyday life. It has been mankind’s dream to, one day, be able to communicate via satellites from anywhere at anytime. This dream is about to be realised through the use of on-board processing (OBP) mobile satellites, constellations of which will encircle the Earth. Two of today’s most ambitious OBP projects are the Iridium system, and the Inmarsat-P standard which offer worldwide communication from pocket-sized, hand-held terminals. Before discussing the details of OBP satellites it is useful to describe the history of satellite communication, so that the reader will understand the context in which we conducted our research.

1.1 Historical Notes on Satellite Communications

1.1.1 Early Experimental Satellites

The first well-known article on communication satellites was published in 1945 [Cla45]. The article pointed out that three satellites spaced equally in the geosta-
1.1 Historical Notes on Satellite Communications

A stationary (circular) orbit could provide communications coverage for the whole world, apart from regions near the poles. These satellites were envisaged as picking up signals from Earth, boosting them and retransmitting them anywhere within view on the globe below. It took more than a decade before the first steps were taken in starting the satellite communications. In 1957 the first artificial satellite, Sputnik I, was launched into a Low Earth Orbit (LEO) by Russians [Dav90]. In January 1958 Americans launched their first satellite called Explorer 1 [Sel93]. The first voice communication via a satellite was achieved in late 1958 with Score (Signal Communication by Orbiting Relay Equipment) [Bro60]. This satellite received messages from the Earth and stored on tape for retransmission.

In 1960 Echo I was launched to test propagation and transmission techniques. Echo I was a passive satellite which merely reflected the received signals. The success of Echo I led to the design and launch of Courier which could record a message for play back at a later time [Imb61].

Telstar [Hol62] was launched in 1962. This was the first active satellite, a signal being transmitted to it in the 6 GHz band and downconverted on-board before retransmission to Earth in the 4 GHz band. Soon after this satellite, Relay [Met63] was put in orbit. The basic objectives of using Relay were to transmit telephone and TV signals across the Atlantic. The satellite gave visibility across the Atlantic Ocean of less than one hour during each pass.

The breakthrough in trans-Atlantic communications came in 1963 when Syncom II [Ben64] became the first satellite in near geostationary orbit, Syncom I having failed before achieving its orbital position. Syncom II was not truly geostationary as it had a 33° inclination. In 1964 Syncom III achieved a true geostationary orbit. These early experimental satellites paved the way for the subsequent fully commercial satellites.

1.1.2 International Telecommunications Satellites (Intelsat)

In 1965 Early Bird (later re-designated Intelsat I) [Vot66] was launched. This satellite, with 240 circuits, tripled the existing trans-Atlantic telephony capacity and made possible the first trans-Atlantic transmission of live television signals.

Intelsat II [Arn68] was a follow-on to Early Bird. Intelsat II satellites had additional capacity for commercial traffic. An objective of the Intelsat III [Fei68] program was to develop satellites with greater capacity than the previous ones, which had a multiple access capability allowing communications between any pair
of terminals within view of the satellite. The Intelsat III program was the first to provide global services, with satellites serving each of the three ocean areas of the world.

The Intelsat III satellites were a significant improvement over the previous Intelsat satellites. However, it was recognised that the continually increasing demand for communication satellite services would shortly require even larger satellites in orbit. Therefore, design work began on Intelsat IV [Wil71]. The main requirements for the Intelsat IV satellites were to provide increased capacity and operational flexibility while remaining compatible with existing ground terminals. Intelsat IV-A [Pil74] was a follow-on to Intelsat IV satellites to provide more capacity over the North Atlantic area.

Further capacity increase in 4 and 6 GHz bands were not so practical with a simple stretching of the Intelsat IV/IV-A design, so development of Intelsat V [Hay78] began. The Intelsat V satellite also had the new feature of using the 11 and 14 GHz bands for nations with the largest traffic volumes.

Intelsat V-A [Mar83] was a modification of the Intelsat V design. Increased capacity and higher reliability were the features of Intelsat V-A satellite.

The trend in increasing the capacity was continued in the design of Intelsat VI satellites. An innovation of Intelsat VI satellites is the introduction of satellite switched time division multiple access (SS-TDMA). The need for the on-board switching arises when frequency re-use is introduced by using several uplink and downlink beams. While this practice increases capacity, traffic flow between beams can be more difficult; flexible connectivity between beams is only obtainable by on-board switching. A microwave switching matrix (MSM) is used for dynamic switching. For Intelsat VI satellites, six beams are switched [Ben84].

1.1.3 International Maritime Satellites (Inmarsat) and Standards

Inmarsat's own satellites are located around the Earth in geostationary orbit to provide communications coverage of all areas of the globe, with the exception of the extreme polar regions. Four overlapping satellite coverage regions mean that large areas of the world's surface have access to more than one satellite [Bel93, Bel92, Ber86].

The Inmarsat-A service was launched in 1982 and offers telephone, telex, data or facsimile services in areas where local networks or terrestrial communications are unavailable or unreliable.
As a follow-on to Inmarsat-A, Inmarsat-B provides all of the capabilities of Inmarsat-A but uses digital transmission techniques for improved efficiency in terms of power and bandwidth utilisation.

In 1991 Inmarsat began commercial operation of Inmarsat-C, a system that provides a low speed, store and forward data communication. Inmarsat-C system can be used as a mobile, transportable or fixed Earth stations.

Inmarsat-M is an all-digital mobile satellite communication system providing telephony, data and the International Telegraph and Telephone Committee (CCITT) Group III facsimile (up to 2.4 kbits/sec) with any terrestrial subscriber via the international public switched network. The modulation is filtered offset quadrature phase-shift keying (OQPSK) [Ahm92]. Forward error correction (FEC) convolutional coding has been used to improve the error rate.

The latest Inmarsat standard, known as Inmarsat-P, is a pocket-sized hand-held terminal which will provide communications via satellites from every square inch of the Earth at any time. The satellite provides the link between the terminal and one of the Land Earth Stations (LESs) located around the world. The LESs, in turn, provide the connection to the public switched telecommunications network and the public mobile land network [Inm94, Inm92]. The Inmarsat-P design incorporates a dual-mode feature; operating as a normal cellular phone when within range of a cellular system or, otherwise, as a satellite phone. To support Inmarsat-P services, a new generation of Inmarsat satellites will be required. This will be in addition to the second and the next generation of Inmarsat satellites. After extensive feasibility studies, Inmarsat has opted for intermediate circular orbit (ICO) satellites to provide worldwide coverage. The choice of ICO for Inmarsat-P satellites has the following advantages:

- Fewer satellites are required (9-15 satellites compared to 66 satellites for the Iridium constellation\(^1\)).

- The satellites will have a longer in-service life [Inm94].

- Although the relative motion of the satellite in ICO orbits with respect to the Earth station gives rise to Doppler shift, this shift in frequency is considerably less than that for the Iridium system.

\(^1\)See the next section.
1.1.4 Motorola’s Iridium System

The name Iridium was chosen for Motorola’s proposed global digital wireless communications network, because the original concept required 77 satellites to span the globe — and 77 is the atomic number of the element Iridium. The 77 satellites have been reduced to 66 by further feasibility studies [Mor92, Iri93]. The main features of the Iridium system and Inmarsat-P are:

- Dual mode portable instruments, designed to work in conjunction with existing cellular systems
- A worldwide coverage
- Support of any type of telephone transmission – voice, data, facsimile and paging.

One major difference between how Inmarsat-P and Iridium are implemented is the choice of orbits and the number of satellites. As mentioned earlier, Inmarsat-P satellites will be in intermediary circular orbits. Iridium satellites will be in polar orbits at an altitude of 780 km. The choice of LEO satellites for the Iridium system will allow more tightly focused beams to be projected on the ground, ensuring strong signals and communication quality. Echo will be minimised due to the satellites’ low orbit, and the receiving antenna can be small enough to be carried on a hand-held subscriber unit. Furthermore, the round-trip delay with the Iridium satellites is less than that with the Inmarsat-P satellites.

1.2 Baseband Switching — The First Step Towards Fully OBP Satellite

The success of Intelsat VI in employing a microwave switching matrix (MSM) has paved the way to use baseband switching in Inmarsat-P and Iridium systems. To have access to baseband data, demultiplexing of the received channels and demodulation of the individual channels must be performed. After switching, the data from \( N \) demodulators are time division multiplexed to form a high rate bit stream. Remodulation of this bit stream onto a single carrier and its amplification for retransmission to ground completes the scenario of using single channel per carrier frequency division multiple access (SCPC/FDMA) on the uplink and time division multiplex (TDM) on the downlink.
1.2 Baseband Switching — The First Step Towards Fully OBP Satellite

The ability to demodulate and remodulate signals provide a degree of decoupling between the mobile link and the feeder link. The most obvious advantage is an improvement in the overall link error rate for a given carrier-to-noise ratio on each link [Eva87]. This effect is shown in Figure 1.1 for an error probability of $10^{-4}$ and QPSK modulation with coherent demodulation.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure1.1.png}
\caption{Regenerative vs. transparent satellites.}
\end{figure}

In conventional satellites the total signal-to-noise ratio $(E_b / N_o)_T$ is

$$(E_b / N_o)_T^{-1} = (E_b / N_o)_U^{-1} + (E_b / N_o)_D^{-1}$$ (1.1)

and therefore the total BER is a function of the uplink SNR, $(E_b / N_o)_U$, and the downlink SNR, $(E_b / N_o)_D$. In links with OBP satellites, the error probability is expressed as the probability of having an error on the uplink ($BER_U$) and no error on the downlink ($1 - BER_D$) or no error on the uplink ($1 - BER_U$) and an error on the downlink ($BER_D$), hence [Mar93]

$$BER = BER_U(1 - BER_D) + (1 - BER_U)BER_D$$ (1.2)

As $BER_D \ll 1$ and $BER_U \ll 1$

$$BER = BER_U + BER_D$$ (1.3)

Therefore with OBP links, the error rates add. An important point which emerges from comparison between transparent and OBP satellites in Figure 1.1 is that with
OBP satellites at a given downlink power, a smaller uplink power is required. This is an important reason why OBP satellites have been chosen for Iridium system and Inmarsat-P standard.

1.3 Access Schemes for OBP Satellites

The problem of accessing the satellite by several users simultaneously is addressed by multiple access schemes. There are three principle multiple access schemes:

- Frequency division multiple access (FDMA)
- Time division multiple access (TDMA)
- Code division multiple access (CDMA)

and are shown in Figure 1.2.

In FDMA, each user can continuously transmit at a pre-assigned frequency band. It is necessary to separate the users by guard bands to reduce the adjacent channel interference (ACI) and to allow for imperfection in multiplexing/demultiplexing of the user channels. FDMA has the advantage of simplicity and relies on the use of proven equipment. A variation of FDMA is SCPC/FDMA in which there is a

Figure 1.2: The 3 principle multiple access schemes.
1.4 Demultiplexing Approaches for OBP Satellites

Use of SCPC/FDMA is the most efficient access scheme in reducing the mobile station complexity and cost [Eva87].

In TDMA, each user can transmit at a pre-assigned time-slot. Unlike the FDMA, in TDMA there is a single carrier which occupies all of the channel bandwidth. Therefore, the high power amplifiers (HPA) can be operated at saturation point. This is an advantage over FDMA in which the HPA must be operated away from its saturation point to avoid intermodulation products as the number of accesses increases. A disadvantage of TDMA is that a high transmitting peak power is required. As the mobile has a small transmitting power, the use of TDMA from them mobile to the satellite is ruled out on the grounds of the transmitter power requirement. Time division multiplexing is more suitable on the satellite – Land Earth Station (LES) link. The satellite or the LES can afford the high power required to employ this access scheme.

In CDMA, a unique code is assigned to the transmitter and the receiver. Each user transmits continuously on the available channel bandwidth. CDMA operates on the principle of spread spectrum transmission. The receiver recovers the useful information by de-spreading the spectrum of the carrier transmitter to its original bandwidth. By this operation, other users will remain as noise of low spectral density with respect to the user of interest. Application of CDMA to mobile OBP satellites is still under research, and will not be covered in this thesis.

1.4 Demultiplexing Approaches for OBP Satellites

In an OBP satellite, it is necessary to recover the individual channels so that they may be switched and recombined for TDM downlink channel. The device that recovers the channels is called a demultiplexer. There are a number of approaches to implement the demultiplexer. An overview of some of the proposed demultiplexing methods is given in the following sections. The following assumptions are made

1. The channels are uniformly spaced, i.e. if the centre frequency of the $k$-th channel is $f_k$, the following condition applies:

   \[ f_{k+1} - f_k = f_k - f_{k-1}, \quad f_{k-1} < f_k < f_{k+1} \]  \hspace{1cm} (1.4)

2. All the channels have the same bandwidth $B$ and sampling frequency $f_s$.  

---

Chapter 1 Introduction and Overview
1.4 Demultiplexing Approaches for OBP Satellites

1.4.1 Direct Method Demultiplexing

In this method, the spectrum of the multiplexed channels is shifted to baseband before the channel is low-pass filtered. Shifting a channel to baseband requires a complex multiplication between each complex sample of the received FDM signal and \( e^{-j2\pi fkT} \), where \( n \) is the \( n \)-th sample of the FDM signal. For a complex multiplication 2 real additions and 4 real multiplications are performed.

Representing the real or the imaginary components of the input signal by \( x[n] \), the low-pass filtering operation on each of these components is simply

\[
y[m] = \sum_{n=0}^{N-1} h[n].x[m - n] \quad m = 0, 1, \ldots
\]

where

- \( h[n] \) is the \( n \)-th coefficient of the low-pass filter with a length \( N \).
- \( x[m] \) is the \( m \)-th real or imaginary component of the shifted FDM signal.

Performing the convolution in (1.5) requires \( N - 1 \) additions and \( N \) multiplications. For demultiplexing \( K \) channel, the following operations must be performed:

\[
2K + \frac{MK(N - 1)}{2}, \quad \text{real additions}
\]

\[
4K + \frac{MKN}{2}, \quad \text{real multiplications}
\]

where \( M = 4 \) for (O)QPSK and \( M = 2 \) for BPSK. Furthermore, \( \frac{MKN}{2} \) memory locations are required to store the filter coefficients. The above computation burden and the memory size, make the direct method demultiplexing inefficient for practical applications. In the next two sections more attractive demultiplexers will be discussed.

1.4.2 Tree Demultiplexer

The simplified block diagram of a 4 channel tree demultiplexer is shown in Figure 1.3. There are 2 stages to recover 4 FDM channels. In general, to demultiplex \( K \) channels, a tree structure with \( \log_2 K \) stages is required. At every stage half of the channels are recovered and the sampling rate of the FDM signal is successively halved. An attraction of using a tree structure is that the same prototype filter is

---

2A baseband channel is a channel whose centre frequency is at DC.
Figure 1.3: Demultiplexing 4 FDM channel by a tree structure.

used at every stage of the tree. Therefore, only \( \frac{M}{2} (N - \frac{1}{2} + 1) \sum_{k=1}^{\log_2 K} 2^{k-1} \) memory locations are required. This is a substantial saving in memory size compared to the direct approach demultiplexing. In practice the saving is even further than this, because the prototype filter of the tree structure has a long transition band, while the prototype filter of the direct approach demultiplexer has a narrow transition band. The impact of the width of the transition band on the design of the filter can be deduced from [Bel80]:

\[
N = \frac{2}{3} \log_{10} \left[ \frac{1}{\delta_1 \delta_2} \right] \frac{f_s}{\Delta f}
\]  

(1.7)

where

- \( N \) is the number of coefficients.

- \( \delta_1 \) and \( \delta_2 \) are the pass band and stop band ripples, respectively.

- \( \Delta f \) is the width of the transition band.

It is clear that \( N \) and \( \Delta f \) are inversely proportional. Therefore the prototype filter length in the direct method demultiplexer is larger than that required in the tree structure. This subject is treated in more details in Chapter 2.

The prototype filter used in the demultiplexer can be nonrecursive (FIR) or recursive (IIR). In the next section the choice of the prototype filter is discussed.

### 1.4.2.1 FIR Prototype Filter

If FIR filters are used, the spectrum is divided into two halves; the lower half includes channels multiplexed in the frequency range from \( f = 0 \) to \( f = f_s / 4 \), and the upper half includes channels multiplexed in the range from \( f = f_s / 4 \) to \( f = f_s / 2 \). This is then followed by decimating the sampling rate by 2 [Göc88, Qi 92]. Two of
1.4 Demultiplexing Approaches for OBP Satellites

the main reasons in using FIR filters in tree structures are the linear phase response of these filters and the design algorithms, such as the Park and McClellan equiripple filter design [McC73], which are widely available. As will be shown in Chapter 2, the prototype filters require a small number of coefficients to design.

1.4.2.2 IIR Prototype Filter

In Section 1.1, it was noted that with every new satellite the communication capacity was increased. Furthermore, it was pointed out that FDMA was proposed for the mobile uplink. Although the FIR approach has its advantages, its implementation is in contradiction to the idea of increasing the capacity; 50% of the spectrum from \( f = f_s / 2 \) to \( f = f_s \) must be left empty. Due to the nature of the designed FIR filters, any channels in this region will be distorted. This side effect of using FIR filters will be demonstrated fully in Chapter 6. To overcome this problem, recently the Ansari and Liu [Ans83] IIR algorithm to design filters with approximately linear phase was implemented. The performance of a tree structure with these IIR filters was evaluated for OBP satellites [Dan94a, Dan94b]. The designed IIR filters allow a different tree structure and that in turn gives a better spectral efficiency. With IIR tree structure, the spectrum is divided into two halves; the side half and the middle half:

\[
\begin{align*}
\text{side half:} & & f = 0 \text{ to } f = f_s / 4 \text{ and from } f = 3f_s / 4 \text{ to } f = f_s \\
\text{middle half:} & & f = f_s / 4 \text{ to } f = 3f_s / 4
\end{align*}
\]

(1.8)

Therefore, apart from guard-bands\(^3\) between channels, the spectrum is fully utilised. Under the motivation of doubling the capacity in the available spectrum, other important advantages in using IIR demultiplexer have been achieved. These advantages such as lower algorithmic complexity, lower demultiplexing delay and a better noise performance will be presented in Chapter 6.

1.4.3 Polyphase Discrete Fourier Transform (DFT) Structure

The tree structure presented in the previous section is a multi-stage demultiplexer. To perform single-stage demultiplexing, polyphase DFT can be used. This structure has been analysed in [Vai90, Cro83]. The block diagram is shown in Figure 1.4, and the operation is as follows:

A prototype filter is decomposed into sub-filters. For each sub-filter, there is a shift register. For a \( K \) channel demultiplexer, there are \( K \) shift registers. By a

---

\(^3\)The spectral gaps between FDM channels to minimise adjacent channel interference.
serial to parallel conversion, one sample of the received signal is put in every shift register. By convolution, $K$ output samples are generated on which a DFT operation is performed to extract the channels. The above process is repeated for all the received FDM signal samples.

When designing an FIR prototype filter for polyphase structures, the transition bands of the filter must be short. A long filter length is a disadvantage of using polyphase structures (see expression (1.7)).

1.5 Modulation Schemes for Satellite Communications

1.5.1 BPSK, QPSK, OQPSK

Binary phase shift keying (BPSK), quadrature phase shift keying (QPSK), and offset QPSK (OQPSK) are the most often used PSK variations for satellite modems (modulators/demodulators) [Skl88].

Figure 1.5(a) shows the generalised quadrature modulator that is applicable for all three schemes. The difference is in the baseband generator, and is as follows:

- For BPSK, the baseband generator is not required, and only the upper half of the modulator is used.

- For QPSK, the baseband generator is a serial to parallel converter.

- For OQPSK, the baseband generator consists of a serial to parallel converter followed by a quadrature channel delay of $T/2$. 

---

Chapter 1 Introduction and Overview
Figure 1.5: Generalised quadrature modulator (a) and demodulator (b).

Figure 1.5(b) shows the generalised quadrature demodulator. It is applicable for all schemes. The difference is in the detector and combiner. The detectors for the above modulation schemes operate on

\[ q(kT): \quad \text{The matched filter output sampled at time } kT \text{ (quadrature channel)} \]
\[ p(kT): \quad \text{The matched filter output sampled at time } kT \text{ (in-phase channel)} \]
\[ \hat{b}_m: \quad \text{Receiver’s decision about the } m\text{-th bit (quadrature channel)} \]
\[ \hat{a}_m: \quad \text{Receiver’s decision about the } m\text{-th bit (in-phase channel)} \]

where

\[ q(kT) = \begin{cases} 0, & \text{BPSK} \\ p(kT), & \text{QPSK} \\ p(kT + T/2), & \text{OQPSK} \end{cases} \]  

(1.9)

and \( \hat{b}_m = 0 \) for BPSK. Therefore, BPSK does not need the lower half of the circuit and the combiner.

All of the above modulation schemes have approximately the same power.
efficiency. That is, for coherent detection the probability of error is \([\text{Mar93}]\)

\[
P_e = \frac{1}{2} \text{erfc} \left( \sqrt{\frac{E_b}{N_0}} \right)
\]

where

- \text{erfc} is the complementary error function and is given by

\[
\text{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} e^{-t^2} dt \tag{1.11}
\]

- \(E_b\) is the bit energy

- \(N_0 / 2\) is the double-sided noise power spectral density (PSD) at the receiver input.

The BER performance is the same for BPSK and (O)QPSK. What differentiates them from each other are the spectral efficiency, the immunity to nonlinearity, and the hardware complexity. Each of these will be discussed next.

### 1.5.1.1 Spectral Efficiency

The spectral efficiency \(\eta\) of a modulation scheme is defined as the ratio of the bit rate, \(f_b\), to the bandwidth, \(B\), expressed in bits/s/Hz. As the bandwidth of the signal is \((1 + \alpha) / T\), where \(\alpha\) is the rolloff factor of the prototype filter, the spectral efficiency is

\[
\eta = \frac{f_b}{B} \tag{1.12}
\]

or

\[
\eta = \begin{cases} 
\frac{2}{1 + \alpha}, & \text{\((O)\)QPSK} \\
\frac{1}{1 + \alpha}, & \text{BPSK}
\end{cases} \tag{1.13}
\]

Therefore, BPSK has the poorest spectral efficiency. QPSK and OQPSK have the same spectral efficiency.

### 1.5.1.2 Immunity to Nonlinearities

Figures 1.6(a), 1.6(b) and 1.6(c) show the scatter diagrams for BPSK and (O)QPSK. The phase variations between two consecutive symbols are:

\[
\Delta \theta = \begin{cases} 
0, 180^\circ & \text{BPSK} \\
0, \pm 90^\circ, \pm 180^\circ & \text{QPSK} \\
0, \pm 90^\circ & \text{OQPSK}
\end{cases} \tag{1.14}
\]
The phase variation $\Delta \theta$ affects the performance of the modem in the presence of nonlinearities. To achieve a maximum power efficiency, high power amplifiers (HPA) are operated near to or at saturation point. One of the damaging effects of these nonlinearities is the spectral spreading of the transmitted signal, which can cause adjacent channel interference (ACI). As $|\Delta \theta_{\text{max}}| = 180^\circ$ for BPSK signals, when they pass through a nonlinear amplifier operating at saturation, there is a large spreading of side-lobes. For QPSK, $|\Delta \theta| = 90^\circ$ occur as well as $|\Delta \theta| = 180^\circ$. Therefore, the spectral spreading is less than that for BPSK. As $|\Delta_{\text{max}}| = 90^\circ$ for OQPSK, the performance of OQPSK modem is better than that of QPSK and BPSK modems in the presence of nonlinearities.
1.5.1.3 Implementation Complexity

BPSK has the lowest hardware complexity; it has half the complexity of a QPSK modem. The OQPSK and QPSK modulators are the same. The difference is in using delay elements in the quadrature component of the complex data (see section 1.5.1). The OQPSK demodulator is slightly more complex than the QPSK demodulator. In OQPSK demodulator, demultiplexers are required to separate samples which occur at times $T$ and $T/2$.

Some examples of the implementation of the discussed modems are BPSK in Inmarsat-B, QPSK in Intelsat IV and OQPSK in Inmarsat-M.

1.5.2 $\pi/4$-QPSK: QPSK without $180^\circ$ Phase Transition

$\pi/4$-QPSK is another modulation suitable for satellite communications, especially mobile satellite communications where differential detection is often needed [Feh91]. It was proposed by Baker in 1962 [Bak62].

![Figure 1.7: Scatter diagram of $\pi/4$-QPSK.](image)

Figure 1.7 shows the $\pi/4$-QPSK signal constellation. It is seen that the carrier phase transitions are restricted to $\pm 45^\circ$ and $\pm 135^\circ$. As the carrier phase of $\pi/4$-QPSK does not undergo instantaneous $\pm 180^\circ$ transitions as in QPSK, the envelope fluctuation due to bandlimiting filtering and subsequent spectral restoration due to nonlinear amplification is significantly reduced.

Either coherent or noncoherent detection can be applied to $\pi/4$-QPSK. As to BER performance, the differentially detected $\pi/4$-QPSK is about 3dB inferior to QPSK [Aka87]; the coherently detected $\pi/4$-QPSK has the same error performance as that of QPSK [Liu91].
The spectrum of the $\pi/4$-QPSK signal is the same as that of QPSK. The difference between a $\pi/4$ - QPSK modulator and a QPSK modulator is that in $\pi/4$ - QPSK modulator the phase of every other symbol is shifted by 45°.

$\pi/4$-QPSK has been chosen as the standard modulation technique for the American Digital Cellular second generation standard radio system and is also considered for the second generation Japanese standard [Feh91].

1.6 Digital Demodulator Synchronisation

A problem in communication via satellite channels is the change in the characteristics of the signal due to conditions such as fading, Doppler shifts, nonlinearities, imperfect synchronisation between the transmitter and the satellite clocks. The characteristics of interest are timing, phase and the frequency of the signal. Assuming that the AWGN channel model holds, the complex MPSK (BPSK: $M = 2$, (O)QPSK: $M = 4$) envelope of the received signal is given by [Jes91]

$$X_L(t) = \left[ \sum_m a_m h_i(t - mT - \tau) + j \sum_m b_m h_q(t - mT - \tau) \right] e^{j(\theta + 2\pi v\tau)} + n(t) \quad (1.15)$$

where

- $a_m$ and $b_m$ are the in-phase and quadrature binary data symbols, which randomly take the values ± 1 (BPSK: $b_m = 0$)

- $h_i(t)$ and $h_q(t)$ are the in-phase and quadrature baseband pulses (BPSK: $h_q(t) = 0$, OQPSK: $h_q(t) = h_i(t - T/2)$), normalised to unit energy

- $n(t)$ is the complex valued Gaussian white noise process, with statistically independent real and imaginary parts, each having a two-sided spectral density of $N_0 / 2$

- $\theta$, $\tau$ and $v$ are the unknown carrier phase, symbol timing and carrier frequency.

The detection and correction of $\theta$, $\tau$ and $v$ is called synchronisation. Many synchronisation algorithms are available in the literature. The algorithms investigated in this thesis are categorised according to the following characteristics:

- Decision-directed (DD) or non data-aided (NDA)
1.6 Digital Demodulator Synchronisation

- Derived from the maximum-likelihood (ML) criterion or from some ad hoc principle; the ad hoc synchronisers are named after the authors who introduced the synchroniser (Viterbi and Viterbi (V&V), Alberty and Hespelt, and Gardner)

- Feedforward (FF) or feedback (FB) configuration

- Dependency of the operation of the error detector on the prior acquisition of other errors, for example whether the timing synchroniser is phase independent.

In selecting a suitable algorithm, several features have to be considered. For the purpose of this thesis, in the context of application to a scenario in which either Inmarsat-P standard or Iridium system are used, the following features have been taken into account:

1. A fast acquisition and correction of $\theta$, $\tau$ and $\nu$;
3. A rare occurrence of cycle slips;
4. A low algorithmic complexity;
5. A large acquisition range.

1.6.1 Comments

Conditions 1 and 2 are in contradiction; feedforward configurations of a synchronisation algorithm result in a fast acquisition of error, but the BER can be poor. On the other hand, feedback configurations result in an improved BER performance at the expense of a longer acquisition time. Any attempt to decrease the acquisition time in feedback configurations, by increasing the closed-loop gain, results in a higher noise. Therefore, a compromise must be found.

Cycle slips have crippling effect on the performance of modems, because each cycle slip gives rise to many decision errors. Symbol period slipping is an example of a type of cycle slipping which may happen in the demodulator. If the actual timing

---

4The Cramer-Rao bound (CRB) is a plot of the error variance at the output of the synchroniser as a function of $E_b / N_0$ or $E_s / N_0$. The CRB is the theoretical lower bound on the performance of a synchroniser. The performance of a practical synchroniser cannot be better than the CRB.
error is $\tau$, the timing error synchroniser will try to acquire an error of $nT \pm \tau$, where $n$ is an integer and $T$ is the symbol period; $n = 0$ means no period slipping, $n = 1$ means that the demodulator has slipped one symbol period and so on. One factor which has the potential of causing cycle slipping is the self-noise of the synchronisation algorithm. Therefore, an algorithm which is self-noise free is desirable.

One of the concerns in using OBP satellites is their complexity. Every possible avenue must be searched to find a system with the lowest possible algorithmic complexity. Therefore, satisfying condition 4 is important to avoid unnecessary complexity on-board the satellite and, hence, increase reliability.

The maximum acquisition ranges, for example for QPSK synchronisers, are $\pm \pi/4$, $\pm T/2$ and $\pm R/2$ for phase, timing and frequency error detection algorithms. Each type of synchronisation will be dealt with in detail in this thesis with the aim of finding a solution which satisfies most or all the above conditions.

### 1.7 Simulation Tool

An important issue in telecommunications research is the performance of the proposed system under conditions that make the transmission channel imperfect. However, actual experimentation with the system may not be possible. This is particularly true when the system of interest does not yet exist. In such cases, analysis and simulation are the only methods which can be used to verify the performance. There are many applications in which deriving an exact analytical model for the system is not possible. Some approximations must be made so that a model can be developed whose performance closely matches that of the real system. The developed model can then be implemented in a hardware prototype or in a computer-based simulation package. The former approach is generally cumbersome, expensive, time-consuming and relatively inflexible [Bal84]. Due to these reasons software packages are widely used in research establishments. One of these packages is called COSSAP\(^6\) which is used for simulating the physical layer of communication systems. An important feature of COSSAP is the self-scheduling. Self-scheduling deals with how many signal elements are processed each time a model is activated. This is important in multirate digital filtering, in which the input and output sampling rates of one or more models are different. The input sampling rate of models is set

\(^{5}\text{It is possible to acquire frequency errors which are larger than } |R/2|. \text{ However, the acquisition time will be prohibitively long for fast applications.}\)

\(^{6}\text{Communication System Simulation and Analysis Package, Version 6.6 from Synopsys.}\)
automatically by the package. An overview of the features of COSSAP has been presented in Appendix E.

1.8 Objectives and Organisation of This Thesis

The concept of advanced OBP satellites has been established for many years. The literature is rich in covering different aspects of on-board processing. The aim of our research was to see what the state-of-the-art in OBP was, and to make an improvement to contribute towards the evolutionary process of these satellites. After an extensive literature survey, it was clear to us that there was one corner which had not been covered yet; a recursive MCDD. From the analysis and simulations performed, it was shown that the recursive MCDD developed outperformed the typically existing ones.

The next objective of our research was to find the most suitable carrier frequency, carrier phase and clock timing synchronisation algorithms for OBP satellites. The results of this part of our research were

- A new carrier frequency synchronisation algorithm
- Two optimised carrier phase and symbol timing synchronisation algorithms.

All the results are covered in this thesis. An outline of the contents of each chapter is as follows:

Chapter 2: Demultiplexing for OBP Satellites

Details of the design of a new IIR demultiplexer by the Ansari and Liu algorithm will be presented. FIR tree structure, the polyphase DFT structure, and the weighted overlap-add DFT structure will be analysed. The analysis will cover the computational complexity, the memory size to store the prototype filter, and the demultiplexing delay. The development of a complete filter design package which includes the optimisation of the designed filters by the Fletcher and Powell algorithm will be reported.

Chapter 3: Frequency Synchronisation Algorithms

Different Doppler correction algorithms will be described. Frequency error detection by search methods is presented first. The use of phase-locked loops (PLLs) to detect very small frequency errors is discussed
next. For medium to large frequency offsets, automatic frequency control (AFC) loops are surveyed. The major problem with AFC loops is their extended acquisition time. Feedforward (FF) implementation of frequency error detectors can be used for the fast acquisition of the Doppler shift. A problem with FF algorithms is their relatively small acquisition range. To overcome this problem and also reduce the high computation rate of AFC loops, a new heuristic rotational frequency error detector algorithm will be presented which works using unmodulated carrier. Simulation results will be presented.

Chapter 4: Digital Timing Synchronisation Algorithms

The decision-directed (DD) and the non data-aided (NDA) search timing error detection (TED) methods will be presented. To overcome the high computation rate of the search techniques, a survey of tracking TED algorithms will be made to find the suitable candidates which have a good noise performance and a low algorithmic complexity. The tracking algorithms surveyed are the Kobayashi’s maximum-likelihood (ML) algorithm, the two-point difference tracker, the Gardner algorithm, the DD mean square algorithm, the data transition and tracking loop (DTTL), and the modified M&M algorithm. The survey shows that the modified M&M algorithm has the desirable feature that it works on one sample per symbol, which makes its application attractive for the proposed Iridium and Inmarsat-P satellites. However, this algorithm has a large self-noise. The self-noise cancellation analysis and simulations will be presented.

Chapter 5: Phase Synchronisation Algorithms

Different carrier phase recovery algorithms for the coherent demodulation of QPSK signals will be presented. As before, the search technique is considered first. Unlike the timing error detection search algorithm, which requires a smaller hardware complexity, the search technique used to detect the phase error requires an excessively large number of arithmetic computations. The tracking techniques considered maximise the log-likelihood function and are the DD and the NDA ML algorithms, the digitised version of the Costas loop, the Viterbi and Viterbi (V&V) algo-
Algorithm, and the DD arctangent algorithm. The analysis and simulations show that the DD ML FF algorithm has a fast acquisition of the phase error with a poor noise performance. To improve the noise performance, the algorithm will be optimised.

Chapter 6: An Optimised 8-Channel IIR MCDD

By the end of Chapter 5, suitable candidate algorithms for demultiplexing and synchronisation will be identified. Final comparisons will be made to select the algorithms which have the lowest possible algorithmic complexity and the fastest possible performance. Unlike the previous chapters, in which the demultiplexing and synchronisation algorithms were studied independently, in Chapter 6 an optimised 8-channel MCDD will be developed in which various algorithms interact.

Chapter 7: Conclusions and Future Work

Conclusions about the overall results will be made, and suggestions for future research will be presented.
Chapter 2
Demultiplexing For OBP Satellites

In this chapter, the fundamental operations performed in sampling rate conversion are reviewed. The details of designing an IIR filter design package by the Ansari and Liu algorithm for a tree demultiplexer are presented. An FIR multi-stage demultiplexer will be analysed. The analysis will be extended to cover the polyphase matrix generalised DFT (GDFT) structure and the weighted overlap-add GDFT structure.

One of the important building blocks in the proposed on-board processing satellites is the demultiplexer whose function is to separate the channels in the received FDM signals. The choice of the type of the demultiplexer depends on factors such as filtering delay, hardware complexity, design modularity and the arrangement of the channels. In the last few years some demultiplexing algorithms have emerged [Ver71, Cro81, Cro83, Ans83, Ans85, Gar85b, Goc88, Vai90, Yim92b, Qi92].

Before analysing different demultiplexing algorithms, it is useful to review the principles of sampling rate conversion.

2.1 A Review of Sampling Rate Conversion

One of the fundamental issues in designing a communication system is the operating sampling frequency at different sub-systems. The following are some of the examples of the need for sampling rate conversion in a simple QPSK modem:

1. A Gray encoding is performed on the source bits to generate symbols. The symbols are then mapped to a QPSK constellation. There is only one sample per symbol. The signal must be band-limited before transmission. A practical
band-limiting filter is designed to have a rolloff, \( \alpha \), in the stopband transition region. Denoting the symbol rate by \( R \), the one-sided bandwidth of the designed filter will be \( (1 + \alpha)R / 2 \). With this bandwidth, the mapped QPSK symbols must be sampled at a rate \( f_s \geq (1 + \alpha)R \) in order to satisfy the Nyquist's sampling rate criterion [Pro89]. Therefore the sampling rate of the QPSK symbols must be increased by a factor of at least two before being band-limited.

2. Assume there are \( K \) of the above transmitters to be multiplexed and each of the transmitters has an output sampling rate of \( F \). Before multiplexing, the sampling rate of each transmitter must be increased by a factor of \( 2^n K \) [Cro83], where

- \( 2^n K F \) is the sampling rate of the FDM signal.
- \( 2^n \) is the oversampling factor.
- \( n = 0, 1, \ldots \) is the oversampling factor. Selecting \( n = 0 \) or \( n = 1 \) results in the input signal to be sampled at the Nyquist rate, or oversampled by a factor of 2 respectively.

3. In the receiver, each of the above channels must be recovered. The process of recovering the channels, reduces the sampling rate from \( 2^n K F \) to \( F \).

4. Before demodulating the individual channels, the sampling rate must be reduced to only 1 sample per symbol to match the rate of the signal after mapping at the transmitter side.

The process of digitally up converting the sampling frequency is called interpolation since samples of the original physical process are created from a reduced set of samples [Cro81]. Similarly, The process of digitally down converting the sampling frequency is called decimation. By down conversion some of the samples are discarded. In the next section the decimation and interpolation are discussed in more detail.

### 2.1.1 Interpolation by Integer Factors

The sampling rate interpolation by an integer factor \( L \) together with an illustrative example are shown in Figure 2.1. The process of interpolation includes sampling rate expansion and low-pass filtering.
For a real input signal $x[n]$, the output of the sampling rate expander is given by [Cro83]:

$$w[m] = \begin{cases} x[m / L], & m = 0, \pm L, \pm 2L, \cdots \\ 0, & \text{otherwise} \end{cases}$$

(2.1)

The side effect of the sampling rate expansion is the replicas of the spectrum of the original signal. With an expansion factor of $L$, $L - 1$ replicas will appear between $f = 0$ and the expanded sampling frequency $f_s$ at $f = f_s / L, f = 2f_s / L, \text{etc}$. To remove these images, an anti-imaging (AI) post-filter with the following ideal...
2.1 A Review of Sampling Rate Conversion

A low-pass characteristic is used:

\[ H_{AI}(\omega) = \begin{cases} L, & |\omega| \leq 2\pi F T / 2 = \pi / L \\ 0, & \text{otherwise} \end{cases} \]  

(2.2)

The implementation of the above ideal filter is infeasible. In practice a filter is designed whose frequency response has a good approximation to the specifications in (2.2). The combination of sampling rate expander and the AI filter is known as interpolator. The relation between the input and the output of the AI filter is

\[ y[m] = \sum_{n=0}^{N-1} h[n] w[m - n] \]  

(2.3)

By combining Equations (2.1) and (2.3) the relation between the input and the interpolated output is of the form:

\[ y[m] = \sum_{n=0}^{N-1} h[n] x\left(\frac{m - n}{L}\right) \]  

(2.4)

2.1.2 Decimation by Integer Factors

The sampling rate decimation by an integer factor \( M \) together with an illustrative example are shown in Figure 2.2. The process of decimation includes low-pass low-pass filtering and sampling rate compression.

By compressing the sampling rate of a signal by an integer factor of \( M \), the \((kM)\)th samples are retained, where \( k = 0, 1, \ldots \). The sampling rate compressor is characterised by the input-output relation [Vai90]:

\[ y[n] = w[Mn] \]  

(2.5)

which says that the output at time \( n \) is equal to the input at time \( Mn \). The side effect of sampling rate compression is aliasing (spectral fold-over) which must be prevented. To avoid aliasing, the sampling rate compressor is preceded by an anti-aliasing (AA) pre-filter whose ideal low-pass characteristic is:

\[ H_{AA}(\omega) = \begin{cases} 1, & |\omega| \leq 2\pi F T / 2 = \pi / M \\ 0, & \text{otherwise} \end{cases} \]  

(2.6)

The design of the AA filter is the same as that of the AI filter, except that the output of the AA filter is not scaled. If the anti-aliasing filter unit impulse response at time \( n \) is denoted by \( h[n] \), then the filtered output \( w[n] \) is expressed by:

\[ w[n] = \sum_{k=0}^{N-1} h[k] x[n - k], \quad n = 0, 1, \ldots \]  

(2.7)
2.1 A Review of Sampling Rate Conversion

![Diagram of decimation by a factor of M and an illustration of the time domain representation of decimation by a factor of M = 3.]

By combining Equations (2.5) and (2.7), the relation between the input and the decimated output is of the form:

\[ y[m] = \sum_{k=0}^{N-1} h[k]x[Mm - k] \]  

(2.8)

2.1.3 Sampling Rate Conversion by Rational Factors

A block diagram of the rational sampling rate convertor is shown in Figure 2.3. Performing interpolation before decimation guarantees the width of the baseband of the intermediate signal is greater than or equal to the width of the baseband of the input or output signals. Rather than using separate anti-imaging and anti-aliasing
2.2 Digital Structures for Decimation and Interpolation

2.2.1 FIR Sampling Rate Convertors

Figure 2.4: Direct implementation of FIR decimation filters.

Figure 2.4 shows the direct implementation of FIR decimation filters in which the output \( y[n] \) is found from

\[
y[m] = h[0]x[m] + h[1]x[m - 1] + \cdots + h[N - 1]x[m - (N - 1)]
= \sum_{n=0}^{N-1} h[n]x[m - n]
\]
Whenever $y[mM]$ is computed, the system is working at $N$ multiplications and $N - 1$ additions per unit time. At other intervals, the system is sitting idle because no output need be computed. An improvement to the system implementation can be obtained if the symmetry property of FIR filters is employed. Assuming there is an odd number of coefficients in the prototype filter $h[n]$, the output is determined from:

$$y[m] = h[0](x[m] + x[m -(N - 1)]) + h[1](x[m - 1] + x[m - (N - 2)])$$
$$+ \cdots + h \left[\frac{N-1}{2}\right] x \left[m - \frac{N-1}{2}\right]$$

$$= \sum_{n=0}^{N-1} h[n](x[m - n] + x[m - (N - 1) + n])$$
$$+ h \left[\frac{N-1}{2}\right] x \left[m - \frac{N-1}{2}\right]$$

(2.12)

Although the filtering delay reduces to $(N - 1)T / 2$, there is still the problem of operating the filter at a high sampling rate. The problem can be overcome by either performing the sampling rate conversion in a multi-stage configuration, or by decomposing $h[n]$ into sub-filters and performing the sampling rate conversion in a single stage. A simple decomposition algorithm which results in a polyphase configuration is [Vai90]

$$h_q[n] = h[nM + q], \quad q = 0, 1, \cdots M - 1$$

(2.13)

in which each of the sub-filters works at $1 / M^th$ of the rate at which the filter $h[n]$ was working, i.e. one sample in every $M$ units of time is received. Therefore, only $(N - 1)M$ additions per unit time (on average), and $N / M$ multiplications per unit time are performed. The polyphase representation is thus an efficient method to minimise the computation load per unit time. This configuration is illustrated in Figure 2.5(a).

As mentioned earlier, an anti-imaging filter must follow the sampling rate expander in order to remove the images of the original baseband signal. The problem with this configuration is that the anti-imaging filter operates at a high sampling frequency. To optimise the interpolator, the filter must be placed before the sampling rate expander. Once again, a polyphase approach gives a more efficient design. The configuration is shown in Figure 2.5(b).

Finally, the rational sampling rate conversion is considered. Direct implementation is inefficient due to the following reasons [Vai90]:

---

Chapter 2 Demultiplexing for OBP Satellites
2.2 Digital Structures for Decimation and Interpolation

(a) Decimation

(b) Interpolation

(c) Rational Sampling rate conversion

Figure 2.5: Sampling rate conversion by polyphase filters.

1. One out of a block of $L$ samples is possibly nonzero. That is, one out of $L$ multipliers is working for every $L$ interpolated samples.

2. One out of a block of $M$ samples is retained at the output of the filter.

This configuration can be improved by a polyphase structure as shown in Figure 2.5(c). A more efficient implementation of the rational sampling rate convertor by polyphase structures will be described in Section 2.6.

2.2.2 IIR Sampling Rate Convertors

The general difference equation for an IIR filter is

$$y[i] = N_0x[i] + N_1x[i - 1] + N_2x[i - 2] \cdots N_{Q-1}x[i - (Q - 1)]$$

$$+ D_1y[i - 1] + D_2y[i - 2] \cdots D_{P-1}y[i - (P - 1)]$$

(2.14)

where $x[n]$ is the input sequence, and $y[n]$ is the filter output sequence. The coefficients $D_1, D_2 \cdots$ govern the pole positions, and the coefficients $N_0, N_1 \cdots$
control the position of zeros in the $z$-plane. The unfactorised transfer function of this filter is [Cro83]

$$\frac{Y(z)}{X(z)} = \frac{\sum_{q=0}^{Q} N_q z^{-q}}{1 + \sum_{p=1}^{P} D_p z^{-p}} \quad (2.15)$$

where $Q$ is the number of zeros, and $P$ is the number of poles. For IIR systems, the cascaded structure is generally preferred over the direct-form structure because of its improved finite-word-length properties: reduced round-off noise, less coefficient sensitivity, and so on [Rab75]. For the purpose of this chapter, the cascaded configuration consists of first- and second-order all-pass sections.

### 2.3 Demultiplexing Terminologies

The following terminologies have been used in describing the demultiplexing algorithms in this chapter.

1. **Odd** and **even** channel stacking: For even and odd channel stacking the centre of channels $k = 0, 1, \cdots K - 1$ are at [Cro83]

   $$f_k = \begin{cases} 
   kW, & \text{Even stacking} \\
   W(2k + 1), & \text{Odd stacking}
   \end{cases} \quad (2.16)$$

   where $W$ is the channel spacing. By default, all channels are of even stacking arrangement. To convert an even channel stacking to an odd channel stacking, the spectrum of the FDM signal is shifted by half the channel spacing, i.e. $W / 2$.

2. **Critical** and **over sampling**: Assuming the sampling rate ($f_s$) of the FDM signal is $IKW$, with $I = 1$ the sampling is critical, otherwise the signal is oversampled by a factor $I$.

### 2.4 Demultiplexing by Direct Form

Figure 2.6 shows the direct per-channel approach to demultiplex the FDM signal. The input and output rates of the demultiplexer are $F$ and $(L/M)F$. In this approach, the channel stacking is even and the sampling frequency is critical. The centre of the spectrum of the desired channel $k$ to be demultiplexed is shifted back to baseband. The sampling rate is then increased by inserting $L - 1$ zeros between consecutive
2.4 Demultiplexing by Direct Form

To remove the images from the resulting spectrum the filter $h[n]$ is used. This filter also acts as an anti-aliasing filter for the sampling rate reduction at the next stage which removes $M - 1$ samples for every block of $M$ samples. Direct method demultiplexing has the disadvantages of:

1. Separate anti-imaging/anti-aliasing (AIAA) filters, and separate complex multipliers are required for each channel.

2. The AIAA filters operate at the high sampling frequency of $(L/M)F_{in}$.

Therefore, the direct-form implementation of a demultiplexer is not used in practice. In the next sections, more efficient demultiplexing implementations are presented.
The half-band FIR filter is a special case of FIR design for sampling rate reduction by a factor of 2. The filter length is $4M - 1$, where $M$ is an integer greater than 0. The frequency response is symmetrical about the half-band frequency $\pi / 2$, justifying the name half-band filters.

For the complex input $x[n]$ and prototype filter coefficients $h[n]$, the decimated output of the low-pass filter $y_L[m]$, and the high-pass filter $y_H[m]$ are [Qi 92]:

$$y_L[m] = \sum_{n=0}^{N-1} h[n]. e^{j\frac{\pi}{2}n}. x[2m - n] \quad (2.17a)$$

$$y_H[m] = \sum_{n=0}^{N-1} h[n]. e^{j\frac{3\pi}{4}n}. x[2m - n] \quad (2.17b)$$

where the impulse response $h[n]$ satisfies the following conditions:

$$h[n] = \begin{cases} 
    h[N - 1 - n] & n = 0, 2, 4, \ldots, N - 1 \\
    0.5 & n = (N - 1) / 2 \\
    0 & \text{otherwise}
\end{cases} \quad (2.18)$$

Equations (2.17a) and (2.17b) state that by shifting the prototype filter by $f_s / 8$, the channels in the band [0 to $f_s / 8$] can be demultiplexed. After decimating by 2, the
attenuated channels in the band \([f_s/4\) to \(f_s/2]\) will vanish. Similarly, by shifting the prototype filter by \(3f_s/8\), the channels in the band \([f_s/4\) to \(f_s/2]\) are demultiplexed. After decimating by 2, the attenuated channels in the band \([0\) to \(f_s/4]\) will vanish. The low-pass and high-pass filters together with sampling rate compressors form a node of a demultiplexing architecture called the tree structure. The number of nodes at every stage of the tree structure is twice that at the previous stage. Therefore, this type of tree structure is known as the binary tree structure.

To estimate the length \(N\) of the prototype filter \(h[n]\) the following equation is used \([Bel80]\):

\[
N \approx \frac{2}{3} \log \left( \frac{1}{10\delta_1\delta_2} \right) \frac{f_s}{\Delta f} \quad (2.19)
\]

With the transition band \(\Delta f\) equal to \(f_s/4\) and the passband tolerance \(\delta_1\) equal to the stopband tolerance \(\delta_2\), \((\delta_1 = \delta_2 = \delta)\), the above equation simplifies to:

\[
N \approx \frac{8}{3} \log \left( \frac{1}{10\delta^2} \right) \quad (2.20)
\]

As an example, an estimate of the filter length for \(\delta = 0.01\) is 8 taps. For practical applications this estimate must be decreased to 7 taps or be increased to 11 taps. With the Parks and McClellan equiripple (FIR) algorithm \([McC73]\), stopband attenuation of about 39 dB and 55 dB can be achieved for 7- and 11-tap HBFS respectively\(^1\). Therefore, it comes with no surprise that in applications in which there is a constraint on the memory size, HBFS are attractive to use. Further saving in computations can be made by taking advantage of the property of Equation (2.18) which reduces the number of multipliers to \(M + 1\) for the low-pass or the high-pass filters.

### 2.5.1 Tradeoffs in using the FIR Tree Structure

Although the binary tree structure has a low hardware complexity and its implementation is modular, it has the following disadvantages:

1. Fast algorithms, such as discrete Fourier transform (DFT), cannot be used.
2. Rational sampling rate conversion cannot be achieved.
3. The number of channels must be an integer power of 2.
4. The channels must be oversampled by a factor of 2.

\(^1\)See chapter 6 for the designed filter with 11 taps.
5. No channels can be placed in the frequency band from \( f = f_s / 2 \) to \( f = f_s \), where \( f_s \) is the sampling frequency of the FDM signal.

6. A post-filter is required to perform rational sampling rate conversion of the recovered channels\(^2\).

### 2.6 Polyphase-Matrix Generalised DFT (GDFT) Structure

The disadvantages of using an FIR tree structure can be resolved by using polyphase matrix GDFT structure [Yim92b]. The polyphase matrix GDFT structure is an extension to the polyphase matrix structure (PMS) which performs rational sampling rate conversion by a factor \( L / M \). The relationship between the real input \( S[n] \) and the output \( r[i] \) of the PMS is [Yim91]:

\[
r_p[i] = \sum_{q=0}^{M-1} \sum_{i=q}^{M-1} S_q[i - i] h_{p,q}[i], \quad p = 0, 1, \ldots, L - 1
\]

where \( h_{p,q}[n] \) is the matrix of low-pass sub-filters, and is determined from the prototype filter coefficients \( h[i] \) according to

\[
h_{p,q}[i] = \begin{cases} 
  h[iML + pM + qL], & \text{if } 0 \leq iML + pM + qL \leq N - 1 \\
  0, & \text{otherwise}
\end{cases}
\]

With the above decomposition the sub-filters do not have the symmetry property of the FIR prototype filter. The upper bound, \( \beta_{p,q} \), and the lower bound, \( \alpha_{p,q} \), of each sub-filter are given by:

\[
\begin{align*}
\alpha_{p,q} &= -\left\lfloor \frac{p}{L} - \frac{q}{M} \right\rfloor \\
\beta_{p,q} &= \left\lceil \frac{N - 1}{ML} - \frac{p}{L} - \frac{q}{M} \right\rceil
\end{align*}
\]

where \([x]\) is the floor function which returns the largest integer less than or equal to \( x \), and \([x]\) is the ceiling function which returns the smallest integer greater than or equal to \( x \). Therefore rather than going through all the combinations of \( p \)'s and all \( q \)'s, one can use the following:

\[
\begin{align*}
\alpha &= -\left\lfloor \frac{L - 1}{L} - \frac{M - 1}{M} \right\rfloor \\
\beta &= \left\lceil \frac{N - 1}{ML} \right\rceil
\end{align*}
\]

\(^2\)This will be made clear with a design example in Chapter 6.
The filtering delay through the polyphase structure is \((\beta - \alpha)T_s\), where \(T_s\) is the sampling period. This is an improvement over the direct form implementation in which the filtering delay is \((N - 1)T_s / 2\).

An example of the PMS configuration is illustrated in Figure 2.8 in which the decimation and interpolation factors are 3 and 4, respectively. The out-of-range coefficients have been shaded and are replaced by zeros (see (2.22)). With a prototype filter of length 47 coefficients, the filtering delay through the PMS is \(4T_s\), which is considerably lower than the delay of \(23T_s\) if the filter were implemented in the direct form. An overview of the operation of the PMS is as follows:

Assume that there are \(P\) sample at the input port, where \(P > M\). Only \(N' = \lfloor P / M \rfloor M\) samples will be processed. The \(N'\) samples are divided into \(MQ\) blocks, where \(Q = N' / M\). The first block is read from the input port. By an anticlockwise commutation these samples are placed into the shift registers \(S_q\). By convolving these samples with the decomposed filter coefficients, \(L \times M\) samples
are generated. Samples $0, L, \cdots, (M - 1)L$ are added together to form an output sample. The commutator then advances one position anti-clockwise and the samples $1, L + 1, \cdots, (M - 1)L + 1$ are added together to form the second output sample. The commutation continues until the samples $L - 1, 2L - 1, \cdots, (M - 1)L - 1$ are added together to form the last output sample in the current block of $L$ samples. Therefore, from a block of $M$ samples, a block of $L$ samples has been generated, i.e. the sampling rate has been changed by a factor $L / M$. The second block of $M$ samples are read from the input port, and the PMS performs the same processing as above to generate another block of $L$ samples. Filtering continues until all the $N$ samples at the input are processed.

Figure 2.9: Polyphase-matrix generalised discrete Fourier transform (GDFT) structure.

To demultiplex the uniformly spaced channels, either the channel is shifted to baseband, or the filter $h_{pq}[r]$ must be centred on the desired channel. Either method gives the same result. Shifting the sub-filters can be achieved by single-sideband
modulation. Therefore (2.21) is changed to:

\[ r_{p}^{k}[l] = \sum_{q=0}^{M-1} \sum_{i=\alpha}^{\beta} S_{q}[l - i] h_{p,q}[i] e^{j\omega i} \]  

(2.25)

or

\[ r_{p}^{k}[l] = \sum_{q=0}^{M-1} \sum_{i=\alpha}^{\beta} S_{q}[l - i] h_{p,q}[i] e^{j\omega(iML+qL)} \]  

(2.26)

By splitting the frequency shift term, we get

\[ r_{p}^{k}[l] = e^{j\omega p M} \sum_{q=0}^{M-1} e^{j\omega q L} \sum_{i=\alpha}^{\beta} S_{q}[l - i] h_{p,q}[i] e^{j\omega iML} \]  

(2.27)

If the channels centre frequencies are shifted to zero, the multiplication term is

\[ e^{-j\omega M(l+q)} \]  

(2.28)

Finally, the general demultiplexer equation with zero centred channels is:

\[ r_{p}^{k}[l] = e^{-j2\pi(k+\frac{\alpha}{2})Zl} \sum_{q=0}^{M-1} e^{j2\pi(k+\frac{\alpha}{2})(q+q_{0})} \sum_{i=\alpha}^{\beta} S_{q}[l - i] h_{p,q}[i] e^{j2\pi \frac{q}{Z} Zl} \]  

(2.29)

where

- \( J \) is an integer that forces a permutation of channel number. The relation between \( J \) and the center frequency of the \( k \)-th channel is

\[ f_{k} = (kJ + k_{0})W \]  

(2.30)

where \( k_{0} \) is the centre frequency of the first channel and \( W \) is the channel spacing.

- The number of channels \( K \) and the decimation factor are related by \( Z \)

\[ \frac{K}{M} = \frac{1}{Z} \]  

(2.31)

The complex exponential terms involving \( k_{0} \) and \( Z \) can be eliminated by choosing \( k_{0} = 0.5 \) and \( Z \) to be an even number. Equation (2.29) simplifies to:

\[ r_{p}^{k}[l] = e^{-j2\pi kZl} \sum_{q=0}^{M-1} e^{j2\pi(k+\frac{\alpha}{2})(q+q_{0})} \sum_{i=\alpha}^{\beta} S_{q}[l - i] h_{p,q}[i] \]  

(2.32)
The inner summation is a polyphase matrix network independent of the channel number. The outer summation is a generalised rectangular \((M \times K)\) GDFT. The Generalised DFT is defined by [Cro81]:

\[
y_k^{GDFT} = \sum_{n=0}^{K-1} y[n] W_k^{-(k+k_0)(n+n_0)}, \quad k = 0, 1, \cdots, K - 1
\] (2.33)

In the GDFT the frequency and the time references are shifted by \(k_0\) and \(n_0\), respectively. With \(k_0 = 0.5\) and \(n_0 = 0\) an odd-frequency DFT results which can be used to transform channels with odd stacking [Ver71]. Similarly, if the time samples are taken as odd multiples of half the sampling period \(T / 2\) and frequency samples are taken as odd multiples of \(\frac{NT}{2}\), where \(N\) is the size of the transform, an odd-time odd-frequency DFT (O^2DFT) results. The simplified block diagram of a polyphase GDFT structure is shown in Figure 2.9.

### 2.6.1 Disadvantages of the Polyphase Matrix GDFT Structure

1. The structure is not flexible. If it is designed to demultiplex \(K\) channels, it cannot be used to demultiplex a different number of channels \(K'\), where \(K' > K\). The following must be re-implemented:
   
   (a) DFT block;
   
   (b) serial-to-parallel converter;
   
   (c) prototype filter.

2. A large memory size and, hence, a large number of arithmetic operations are required.

### 2.7 Weighted Overlap-Add GDFT Structure

Another basic multirate structure for a uniform GDFT filter bank is the weighted overlap-add structure (WOAS) [Cro80, Cro83]. This structure is based on block-by-block transform analysis of the signal. The number of channels \(K\) and the decimation factor \(M\) can be unrelated; that is, there is no restriction to the relationship between \(M\) and \(K\). An overview of the derivation is follows:

The block diagram of the filter bank is shown in Figure 2.10. Assume there is a shift register of size \(NK\), where \(N\) is an integer and \(K\) is the size of the Fourier
2.7 Weighted Overlap-Add GDFT Structure

Figure 2.10: Weighted overlap-add GDFT structure.

Transform. Blocks of samples of size $M$ are read into the shift register. These samples are weighted by a time-reversed analysis filter $h[n]$: 

$$x_w[n] = h[n]x[n], \quad n = 0, 1, \ldots, K - 1$$  \hspace{1cm} (2.34)

The weighted samples are then time-aliased. That is, the samples are stacked and added. The time-aliased samples are:

$$x_s[r] = \sum_{r'=0}^{K-1} x_w[r + r'K], \quad r = 0, 1, \ldots, K - 1$$  \hspace{1cm} (2.35)

As a final stage, a generalised DFT is performed on the time-aliased data:

$$X_s[k] = \sum_{r=0}^{K-1} x_s[r]W_K^{(k+r)\beta} = \sum_{r'=0}^{K-1} x[r + r'K]h[r + r'K]W_K^{(k+r)\beta}$$  \hspace{1cm} (2.36)

where $W_K = e^{-j2\pi\beta}$. Combining Equations (2.34), (2.35) and (2.36) results in:

$$X_s[k] = \sum_{r=0}^{K-1} \sum_{r'=0}^{K-1} x[r + r'K]h[r + r'K]W_K^{(k+r)\beta}$$  \hspace{1cm} (2.37)

The shift register is now updated by another block of $M$ samples by first performing:

$$x[NK - 1 - n] = x[NK - M - 1 - n], \quad n = 0, 1, \ldots, NK - M - 1$$  \hspace{1cm} (2.38)
2.8 IIR Approach to Demultiplexing

and then placing a new block of \( M \) samples at

\[
x[M - 1], x[m - 1], \cdots, x[1], x[0]
\]  

(2.39) 

The process of weighting, time aliasing and transforming will be repeated as above to obtain a new set of \( K \) samples for the channels.

2.7.1 Disadvantages of the Weighted Overlap-Add Structure

This structure is similar to the polyphase GDFT demultiplexer with respect to a lack of flexibility, and a large memory size. Furthermore, the structure cannot perform a rational sampling rate conversion. Only sampling rate reduction is possible. This means that, a post-filter is required to perform a rational sampling rate conversion on the recovered channels. The overall hardware complexity of the post-filter and the demultiplexer is a major disadvantage of using the weighted overlap-add demultiplexer.

2.8 IIR Approach to Demultiplexing

The demultiplexing structures we analysed so far were different forms of nonrecursive (FIR) filtering. In order to give our research a clear direction which had not been taken by others, it was decided to explore the feasibility of using recursive (IIR) demultiplexing for OBP satellites. The problem with the conventional IIR filter design algorithms [Atl86] is that they have a nonlinear phase response, which is a drawback in applying these algorithms for demultiplexing purposes. With a nonlinear phase response different frequency components of the input signal experience different timing delays. This may distort the signal. In the following sections the implementation of an IIR filter design package based on the Ansari and Liu algorithm is discussed. The attractive feature of this algorithm is that IIR filters with an approximately linear phase response can be designed. The application of the designed filters for demultiplexing will be presented in Chapter 6.

2.9 A Package for Designing IIR Demultiplexers

The block diagram of the main components of the developed package are shown in Figure 2.11. It consists of a user interface, an optimisation algorithm and the plotting routines. The user provides the package with an initial estimates of the pole
positions and the lower bound (minimum) of the error\textsuperscript{3} function to be minimised. Furthermore, the analytical expressions for the objective function and its gradients must be specified. Separate set of routines have been developed for the manipulation of vectors and matrices. After a set of optimised filter poles are found, they can be plotted. These poles can also be used as input parameters in the new COSSAP models developed for this package. In the following sections, the filter design and the optimisation algorithms will be described.

### 2.9.1 Ansari and Liu Algorithm

The Ansari and Liu (A&L) algorithm is for designing IIR polyphase sampling rate convertors. The structure consists of sub-filters; each one consisting of first- and second-order sections. The method discussed in this chapter is used to design prototype filters with an approximately linear phase response. The A&L algorithm provides general analytical expressions for the $q$-th weighted norm\textsuperscript{4} error function and its gradients. The solution to filter coefficients is found by an iterative optimisation algorithm [Fle63]. The implementation of these algorithms and the associated routines has resulted in a stand-alone filter design package.

\textsuperscript{3}The error between the desired and the actual filter responses.

\textsuperscript{4}See appendix A for more information on norms.
2.9.1.1 Design Method

The design of IIR prototype filters with an approximately linear phase is described in this section\(^5\). The error function is given by

\[
\varepsilon(q) = \left\{ \sum_{k=1}^{K} W_k |H_k - H_k^d|^{2q} \right\}^{1/2q}
\]  

(2.40)

where

- \(H_k\) is

\[
H(z) = 1 + \sum_{n=2}^{N} z^{n-1} (z^L)^{m_n} \left( \frac{P_{nj}z^L - 1}{(z^L - P_{nj})(z^L - \bar{P}_{nj})} \right) \Bigg|_{z=\epsilon^i n L}
\]

(2.41)

- \(H_p(z^L)\) is a cascade of first- and second-order\(^6\) sections

\[
H_p(z) = \prod_{i=1}^{N^p} \frac{-C_{ni}z^L}{z^L - C_{ni}} \times \prod_{j=1}^{N^s} \frac{(P_{nj}z^L - 1)(\bar{P}_{nj}z^L - 1)}{(z^L - P_{nj})(z^L - \bar{P}_{nj})} \Bigg|_{z=\epsilon^i n L}
\]

(2.42)

- \(H_k^d\) in (2.40) is the ideal filter response given by

\[
H_k^d = \begin{cases} 
L & \text{in passband for interpolator} \\
1 & \text{in passband for decimator} \\
0 & \text{stopband}
\end{cases}
\]  

(2.43)

- \(C_{ni}\) and \(P_{nj} = R_{nj}e^{i\theta_0}\) are the real and complex poles of the \(i\)-th first-order and \(j\)-th second-order sections of the \(n\)-th sub-filter.

- \(N\) is the number of sub-filters.

- \(N_p^n\) and \(N_s^n\) are the number of first- and second-order sections of the \(n\)-th sub-filter.

- For an interpolator polyphase filter \(L\) is the interpolation factor. For a decimator polyphase filter, \(L\) is the decimation factor.

- \(T'\) is the sampling period at the higher sampling rate.

- \((z^L)^{m_n}\) reduces the inherent delay in the \(n\)-th sub-filter so that it is not in excess of the required fractional delay. A suitable initial value of \(m_n\) is \(N_p^n + 2N_s^n - 1\). Generally a search in the neighbourhood of this value must be made to choose the best \(m_n\).

---

\(^5\)For the design of filters with a nonlinear phase response see Appendix B.

\(^6\)The order here designates the degree of the denominator polynomial of the transfer function of the recursive filter, with the numerator polynomial assumed to be of at most the same degree.
2.9 A Package for Designing IIR Demultiplexers

- $\omega_k$ is $k$-th angular frequency in the passband or stopband.

The sub-filters in the polyphase structure are arranged in parallel. Each of the
sub-filters is further arranged as a cascade of first- and second-order all-pass filters\(^7\).

To find the gradient of the error function with respect to any unknown parameter
$\beta_n$, it is more convenient to take the power of $2q$ of both sides of Equation (2.40) so that

$$e_1^{2q}(q) = \sum_{k=1}^{K} W_k |H_k - H_k^d|^{2q}$$

The above can be re-written as

$$e_1^{2q}(q) = \sum_{k=1}^{K} W_k \left\{ \sqrt{(H_k - H_k^d)(H_k - H_k^d)} \right\}^{2q}$$

The partial derivative of Equation (2.45) is

$$\frac{\partial e_1^{2q}(q)}{\partial \beta_n} = \sum_{k=1}^{K} W_k \cdot 2q |H_k - H_k^d|^{2q-2} \cdot \left[ (H_k - H_k^d) \frac{\partial H_k^d}{\partial \beta_n} + (H_k^d - H_k) \frac{\partial H_k}{\partial \beta_n} \right]$$

The terms in $\left[ \right]$ are complex conjugate pairs. Hence,

$$\frac{\partial e_1^{2q}(q)}{\partial \beta_n} = \sum_{k=1}^{K} W_k \cdot 2q |H_k - H_k^d|^{2q-2} \cdot \Re[(H_k^d - H_k) \frac{\partial H_k}{\partial \beta_n}]$$

In the above equation $H_k^d$ is a real filter, and hence $H_k^d = H_k^d$. From the definition of

$$\frac{\partial \ln y(x)}{\partial x} = \frac{1}{y(x)} \frac{\partial y(x)}{\partial x}$$

and assuming that $H_p = \frac{f_k(\beta_n)}{g_k}$, from (2.41)

$$\frac{\partial H_k}{\partial \beta_n} = z^p \cdot L^{m_a} \cdot 1 \cdot \frac{\partial f_k(\beta_n)}{\partial \beta_n}$$

Substituting Equation (2.49) in (2.47) results in

$$\frac{\partial e_1^{2q}(q)}{\partial \beta_n} = \sum_{k=1}^{K} 2q W_k |H_k - H_k^d|^{2q-2} \cdot \Re[G_{pk}(H_k^d - H_k^d) \frac{\partial \ln f_k(\beta_n)}{\partial \beta_n}]$$

where

$$G_{pk} = (z^L)^{m_e} \cdot z^n H_n(z^L) \Big|_{z=\exp(i\phi_n)}$$

For a first-order section the parameters $\beta_n$ are wholly real and have radii $R_n$. For
the second-order sections, the parameters $\beta_n$ are complex and have radii $R_n \exp(i\phi_n)$.

\(^7\)For a general discussion on the features of all-pass filters see Appendix C.
2.9.1.2 Phase Response and the Group Delay

The transfer function in (2.41) can be written as:

\[
H(\omega_k) = 1 + \sum_{n=2}^{N} e^{j(\omega_k T(n-1) + \omega_k T L_m + \sum_{i=1}^{N_r} \phi_{ni}^{FOS}(\omega_k) + \sum_{j=1}^{N_c} \phi_{nj}^{SOS}(\omega_k))}
\]  

(2.52)

where the phase responses of the first-order sections (FOS) and the second-order sections (SOS) are given by:

\[
\theta_{ni}^{FOS}(\omega_k) = \tan^{-1} \left( \frac{-c_{ni} \sin(\omega_k T)}{1 - c_{ni} \cos(\omega_k T)} \right)
\]

\[
\theta_{ni}^{SOS}(\omega_k) = \tan^{-1} \left( \frac{r_{ni} \sin(\omega_k T + \theta_{ni})}{r_{ni} \cos(\omega_k T + \theta_{ni}) - 1} \right) + \tan^{-1} \left( \frac{r_{ni} \sin(\omega_k T - \theta_{ni})}{r_{ni} \cos(\omega_k T - \theta_{ni}) - 1} \right) - \tan^{-1} \left( \frac{\sin(\omega_k T) - r_{ni} \cos(\theta_{ni})}{\cos(\omega_k T) - r_{ni} \cos(\theta_{ni})} \right) - \tan^{-1} \left( \frac{r_{ni} \sin(\theta_{ni}) + \sin(\omega_k T)}{\cos(\omega_k T) - r_{ni} \cos(\theta_{ni})} \right)
\]  

(2.53)

(2.54)

Therefore the overall phase response of the polyphase structure is:

\[
\theta(\omega_k) = \tan^{-1} \left( \frac{\sum_{m=1}^{L-1} \sin(\theta_m^T)}{1 + \sum_{n=1}^{L-1} \cos(\theta_n^T)} \right)
\]  

(2.55)

where

\[
\theta_m^T(\omega_k) = \omega_k T n + \omega_k T L_m + \sum_{i=1}^{N_r} \theta_{ni}^{FOS}(\omega_k) + \sum_{j=1}^{N_c} \theta_{nj}^{SOS}(\omega_k)
\]  

(2.56)

By taking the negative of the first derivative of (2.55) and performing some algebraic manipulations, the following expression for the group delay, \(\tau_g\), can be derived:

\[
\tau_g = - \left\{ \left[ \sum_{n=2}^{N} \cos(\phi_n) \frac{\partial \phi_n}{\partial \omega_k} \right] \left[ 1 + \sum_{n=2}^{N} \cos(\phi_n) \right] + \left[ \sum_{n=2}^{N} \sin(\phi_n) \frac{\partial \phi_n}{\partial \omega_k} \right] \left[ \sum_{n=2}^{N} \sin(\phi_n) \right] \right\} / \left[ \left( \sum_{n=2}^{N} \sin(\phi_n) \right)^2 + \left[ 1 + \sum_{n=2}^{N} \cos(\phi_n) \right]^2 \right]
\]  

(2.57)
where
\[
\frac{\partial \phi_n}{\partial \omega_k} = T'Lm_n + T'(n - 1) + \sum_{i=1}^{N_k} \frac{LT'(C_{ni}^2 - 1)}{1 + C_{ni}^2 - 2C_{ni}\cos(L\omega_kT')} + \sum_{j=1}^{N_k} \left[ \frac{LT'(R_{nj}^2 - 1)}{1 + R_{nj}^2 - 2R_{nj}\cos(L\omega_kT' + \theta_{nj})} \right]
\]

The plotting routines of the developed package can be used to plot the phase response, group delay, and the frequency response.

The filter design algorithm only specifies the analytical expressions for the error function and the gradients. To minimise the function, in order to find a set of optimum poles, an optimisation algorithm must be used. In the next section, the optimisation algorithm used in the package developed is presented.

### 2.9.2 Fletcher and Powell Algorithm

Optimisation methods have been described adequately in the literature, e.g. [Wil67, Coo70, Bev70]. The problem of optimisation is to minimise or to maximise a given function. The Fletcher and Powell (F&P) method is a general procedure for finding a local minimum. The authors have proved a number of theorems to show that it always converges and converges rapidly. In describing the F&P algorithm, the following notations are used:

- \( |x| \): A column vector with elements \( x_1, x_2, \ldots, x_n \)
- \( <x| \): The row vector with the same elements as above
- \( <x|y> \): The scalar product of \( <x| \) and \( |y> \)
- \( |x><y> \): The cross-product of \( <x| \) and \( |y> \)
- \( f(|x>) \): The objective function with \( |x> \) as its argument
- \( |g> \): Gradient of \( f(|x>) \).

Furthermore, the following must be known in the implementation of the F&P algorithm:

1. Estimates of the minimum points \( |x_i> \) and the minimum value, \( f_o \), of the function to be minimised (objective function).
2. Analytical expressions for the objective function and its first partial derivatives.
3. The desired degree of accuracy $\epsilon$ with which the minimum point is determined.

4. The maximum number of allowable iterations $N$.

### 2.9.2.1 Minimisation Procedure

The F&P algorithm is an optimisation technique to minimise a general multi-variable, unconstrained nonlinear objective function. The algorithm is iterative and at every iteration new values of the objective function and its gradients are found. The optimisation algorithm determines a search direction, conducts a one dimensional search to converge to the minimum point. If the convergence test is positive the local minimum point has been found. Otherwise, a new search direction is determined and the optimisation repeats. The details of the algorithm are as follows:

1. From the number of elements in the vector $|x_0>$ determine its size $n$.

2. Create a symmetric positive definite matrix $[H]$ of size $n \times n$.

3. Assign 1 to all the elements of the matrix, i.e. $[H]^1 = 1$.

4. Set the iteration counter $k$ to zero.

5. Establish a direction along which to search for a relative minimum by setting the search direction to

$$|s^k| = -[H]^k|g^k|$$

(2.59)

With $[H]^1$ set initially equal to unity, $|s^1>$ is in the steepest descent direction.

6. Find the next local minimum point in the interval between $|x^k>$ and $|y^k| >= |x^k| + \eta|s^k|$ where

$$\eta = \min \left\{ 1, \frac{-2(f(|x^k|) - f_o)}{D_x} \right\}$$

(2.60)

(a) Now interpolate cubically in the interval defined by the search above using Davidon's equation [Dav59]:

$$\alpha^k = \left( 1 - \frac{D_y + w - z}{D_y - D_x + 2w} \right) \eta$$

(2.61)

---

8A symmetric matrix $[H]$ is positive definite if and only if all the eigenvalues of $[H]$ are positive [Ant91]. Similarly, for such a matrix $<x|H|x>$ is greater than 0 where $|x| \neq 0$.
2.9 A Package for Designing IIR Demultiplexers

where

\[ w = \sqrt{z^2 - D_x D_y} \]  \hspace{1cm} (2.62a)

\[ z = \frac{3f(< x^k)|f(< y^k)\lambda}{4} + D_x + D_y \]  \hspace{1cm} (2.62b)

\[ D_x = < g^k(< x^k)|s^k > \]  \hspace{1cm} (2.62c)

\[ D_y = < g^k(< y^k)|s^k > \]  \hspace{1cm} (2.62d)

(b) If the value of the actual function at the new minimum is less than that at the interval ends, terminate the search for the current iteration. Otherwise reduce the interval and repeat the interpolation; choose the interval \((|x^*|, |x^*| + |\alpha^*|s^k)\) if the gradient at the minimum point is positive, i.e. we know that the minimum cannot exist outside the above range. Otherwise choose the interval \((|x^*| + |\alpha^*|s^k, |y^k|)\)

7. Update the Hessian matrix:

\[ [H]^{k+1} = [H]^k + [A]^k - [B]^k \]  \hspace{1cm} (2.63)

where

\[ [A]^k = \frac{|\Delta x^k > < \Delta x^k|}{< \Delta x^k|\Delta g^k >} \]  \hspace{1cm} (2.64a)

\[ |\Delta x^k >= |x^{k+1}| - |x^k| \]  \hspace{1cm} (2.64b)

\[ |\Delta g^k >= |g^{k+1}| - |g^k| \]  \hspace{1cm} (2.64c)

\[ [B]^k = \frac{|H|^k|\Delta g^k > < \Delta g^k|\Delta g^k >}{< \Delta g^k|\Delta g^k >} \]  \hspace{1cm} (2.64d)

The above update is a desirable modification because it causes \([H]\) to approach the inverse of the matrix of the second derivative \(G^{-1}\) without directly evaluating \(G^{-1}\) [Con70] which includes the second derivatives of the objective function.

8. If convergence has not been obtained and the iteration count is less than \(N\), increment the counter and repeat the search from Step 5. The new value of the gradient and the new vector of the point will be used for the next iteration.

Following the completion of the package, some tests were performed to verify that the package was working. The initial convergence functional tests and their results are presented in the following section.

Chapter 2 Demultiplexing for OBP Satellites
2.9.2.2 Convergence Test of The Rosenbrock Function

The Rosenbrock function is [Fle63]:

\[ y(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \]  \hspace{1cm} (2.65)

and is plotted in Figure 2.12. This function represents a highly nonlinear surface in the shape of narrow curved falling valleys, and is a classic test function to show that an optimisation algorithm can maneuver around its curves. Computer simulations were started from \( x_1 = -1.2, \ x_2 = 1 \), so that the current point had to descend into the valley, and then follow it around its curve to the minimum point. The numerical results in the Table 2.1 show that the developed routines converged to the minimum in 23 iterations. The Rosenbrock function has a minimum value \( y(x_1, x_2) = 5.56784 \times 10^{-13} \) at \( x_1 = 1 \) and \( x_2 = 0.999999 \), with a curved valley following the parabola \( x_2 = x_1^2 \).

2.9.2.3 Convergence Test of The Powell Function

The Powell function is given by [Pow61]

\[ f(x_1, x_2, x_3, x_4) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4 \]  \hspace{1cm} (2.66)
Table 2.1: Convergence test of the Rosenbrock function.

<table>
<thead>
<tr>
<th>Iteration</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$y(x_1, x_2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-1.2</td>
<td>1</td>
<td>24.2</td>
</tr>
<tr>
<td>2</td>
<td>-1.03023</td>
<td>1.0693</td>
<td>4.12811</td>
</tr>
<tr>
<td>3</td>
<td>-0.867547</td>
<td>0.679228</td>
<td>4.02662</td>
</tr>
<tr>
<td>4</td>
<td>-0.929892</td>
<td>0.828769</td>
<td>3.85358</td>
</tr>
<tr>
<td>22</td>
<td>.999939</td>
<td>0.999876</td>
<td>4.3868 x 10^{-9}</td>
</tr>
<tr>
<td>23</td>
<td>1</td>
<td>0.999999</td>
<td>5.56784 x 10^{-13}</td>
</tr>
</tbody>
</table>

Starting at (3,-1,0,1), in six iterations the optimisation method reduced $f$ from 215 to $2.3 \times 10^{-8}$.

### 2.9.2.4 Convergence Test of Quadratic Equations

Test 1: The following quadratic equation was used:

$$f(x_1, x_2) = x_1^2 - 2x_1x_2 + 2x_2^2$$  \hspace{1cm} (2.67)

The complete progress of the optimisation process is given in Table 2.2. The results presented agree with those given in [Fle63].

Table 2.2: Convergence test of a quadratic function.

<table>
<thead>
<tr>
<th>$x$</th>
<th>-4</th>
<th>2</th>
<th>-1.69</th>
<th>-1.08</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>1</td>
<td>0</td>
<td>0.781</td>
<td>0.361</td>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>0.361</td>
<td>0.411</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>$</td>
<td>\Delta x</td>
<td>&gt;$</td>
<td>2.31</td>
<td>-3.08</td>
<td>1.69</td>
<td>1.08</td>
</tr>
<tr>
<td>A</td>
<td>0.069</td>
<td>-0.092</td>
<td>0.931</td>
<td>0.592</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-0.092</td>
<td>0.123</td>
<td>0.592</td>
<td>0.377</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>$f$</td>
<td>40</td>
<td>1.54</td>
<td>$10^{-15}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k$</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Test 2: The following test problem was taken from [Kue73]

$$f(x_1, x_2) = -3803.84 - 138.08x_1 - 232.92x_2 + 123.08x_1^2 + 203.64x_2^2 + 182.25x_1x_2$$ \hspace{1cm} (2.68)
With the starting points $x_1 = 1$ and $x_2 = 0.5$, the package minimised the function in 3 iteration. The algorithm answers were

<table>
<thead>
<tr>
<th>$x$</th>
<th>1</th>
<th>0.5</th>
<th>0.586712</th>
<th>0.182635</th>
<th>0.205659</th>
<th>0.479863</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f$</td>
<td>-3793.27</td>
<td>-3858.7</td>
<td>-3873.92</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k$</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The answers shown in the above table agree with those given in [Kue73].

### 2.9.2.5 Comments on the Convergence Tests

The following comments about the above convergence tests can be made:

1. The behaviour of the optimisation algorithm is such that it moves steadily towards the neighbourhood of the local minimum point and then converges rapidly.

2. The number of iterations depends on how good the set of the estimated parameters is.

3. Excluding some round-off error, the gradient vanishes at the minimum point.

### 2.9.3 Realisation of the Designed Filters

After the optimised filter poles have been found, the system must be implemented. A systematic method has been outlined in [Mit74] to realise the first- and second-order digital all-pass networks of Equation (2.42). To derive the network, the transfer matrices are formed. The results are both canonic and noncanonic configurations. In a canonic configuration, the number of delay elements is equal to the order of the filter. The choice of the optimum configuration, which has the minimum multiplication round-off error, depends on the location of the real (first-order sections) and the complex poles (second-order sections). For each of the network configurations presented in the following sections there is a transpose implementation. The characteristics of each of the realisations must be individually studied in order to find the best network for a particular application.
2.9.3.1 Realisation of First-Order Sections

The transfer function of a first-order section is given by

\[ H(z) = \frac{z^{-L} - b_1}{1 - b_1 z^{-L}} \]  

(2.69)

If it is required the final realisation to contain a single multiplier \( b \), then it will be of the form of a constrained two-pair [Mit73] as shown in Figure 2.13, where the two-pair contains delays and adders.

![Figure 2.13: The generalised first-order all-pass section.](image)

The equation describing the network of Figure 2.13 are then

\[
\begin{bmatrix}
Y_1 \\
Y_2
\end{bmatrix} = 
\begin{bmatrix}
t_{11} & t_{12} \\
t_{21} & t_{22}
\end{bmatrix}
\begin{bmatrix}
X_1 \\
X_2
\end{bmatrix}
\]  

(2.70)

and

\[ X_2 = b_1 Y_2 \]  

(2.71)

where \( t_{ij} \) are the transfer parameters [Mit73] of the two-pair network. Eliminating the unwanted variables from (2.70) and (2.71), the following is obtained

\[
\frac{Y_1}{X_1} = \frac{t_{11} - b_1 (t_{11} t_{22} - t_{12} t_{21})}{1 - b_1 t_{22}}
\]  

(2.72)

comparing (2.72) with (2.69), the following are identified:

\[ t_{11} = t_{22} = z^{-L} \]  

(2.73)

and

\[ t_{11} t_{22} - t_{12} t_{21} = 1 \]  

(2.74)

---

\(^9\)First-order in \( z^{-L} \)
Substituting (2.73) in (2.74), it can be re-written as

\[ t_{12}t_{21} = z^{-2L} - 1 \] (2.75)

There are four possible realizable sets of values of \( t_{12} \) and \( t_{21} \) satisfying (2.75), which lead to four different realisations of the first-order section. The transfer matrices of the corresponding two-pairs are:

Network 1A:

\[
\begin{bmatrix}
Y_1 \\
Y_2 \\
\end{bmatrix} = 
\begin{bmatrix}
z^{-L} & z^{-2L} - 1 \\
1 & z^{-L} \\
\end{bmatrix}
\begin{bmatrix}
X_1 \\
X_2 \\
\end{bmatrix}
\] (2.76)

Network 1B:

\[
\begin{bmatrix}
Y_1 \\
Y_2 \\
\end{bmatrix} = 
\begin{bmatrix}
z^{-L} & z^{-L} + 1 \\
z^{-L} - 1 & z^{-L} \\
\end{bmatrix}
\begin{bmatrix}
X_1 \\
X_2 \\
\end{bmatrix}
\] (2.77)

Network 1A:

\[
\begin{bmatrix}
Y_1 \\
Y_2 \\
\end{bmatrix} = 
\begin{bmatrix}
z^{-L} & 1 \\
z^{-2L} - 1 & z^{-L} \\
\end{bmatrix}
\begin{bmatrix}
X_1 \\
X_2 \\
\end{bmatrix}
\] (2.78)

Network 1B:

\[
\begin{bmatrix}
Y_1 \\
Y_2 \\
\end{bmatrix} = 
\begin{bmatrix}
z^{-L} & z^{-L} - 1 \\
z^{-L} + 1 & z^{-L} \\
\end{bmatrix}
\begin{bmatrix}
X_1 \\
X_2 \\
\end{bmatrix}
\] (2.79)

From the above transfer matrices and using (2.71), the realisations of the first-order sections shown in Figure 2.14 can be obtained.

### 2.9.3.2 Realisation of Second-Order Sections

The transfer function of a second-order section is:

\[
G(z) = \frac{z^{-2L} - b_2z^{-L} + b_3}{1 - b_2z^{-L} + b_3z^{-2L}}
\] (2.80)

where \( b_2 \) and \( b_3 \) are real numbers. The final realisation can be considered as a constrained three-pair, as illustrated in Figure 2.15, where the network contains only delays and adders.

The equations characterising the network are:

\[
\begin{bmatrix}
Y_1 \\
Y_2 \\
Y_3 \\
\end{bmatrix} = 
\begin{bmatrix}
t_{11} & t_{12} & t_{13} \\
t_{21} & t_{22} & t_{23} \\
t_{31} & t_{32} & t_{33} \\
\end{bmatrix}
\begin{bmatrix}
X_1 \\
X_2 \\
X_3 \\
\end{bmatrix}
\] (2.81)
Figure 2.14: First-order all-pass networks.

Figure 2.15: The generalised second-order all-pass section.

and

\[ X_2 = b_2 Y_2, \quad X_3 = -b_3 Y_3 \]  

where \( t_i \) are the transfer matrices of the second-order section. By performing some algebraic manipulations, the following transfer matrices can be derived:

\[
\begin{bmatrix}
  Y_1 \\ Y_2 \\ Y_3
\end{bmatrix} =
\begin{bmatrix}
  z^{-2L} & z^{-2L} - 1 & z^{-2L} - 1 \\
  z^{-L} & z^{-L} & z^{-L} \\
  z^{-2L} + 1 & z^{-2L} & z^{-2L}
\end{bmatrix}
\begin{bmatrix}
  X_1 \\ X_2 \\ X_3
\end{bmatrix}
\]  

(2.83)
2.9 A Package for Designing IIR Demultiplexers

Network 3B: $
\begin{bmatrix}
Y_1 \\
Y_2 \\
Y_3
\end{bmatrix} =
\begin{bmatrix}
z^{-2L} & z^{-L+1} & z^{-L+1} \\
z^{-L}(z^{-L} - 1) & z^{-L} & z^{-L} \\
(z^{-L} - 1)(z^{-2L} + 1) & z^{-2L} & z^{-2L}
\end{bmatrix}
\begin{bmatrix}
X_1 \\
X_2 \\
X_3
\end{bmatrix}
$ (2.84)

Network 3C: $
\begin{bmatrix}
Y_1 \\
Y_2 \\
Y_3
\end{bmatrix} =
\begin{bmatrix}
z^{-2L} & z^{-L} & z^{-L} \\
z^{-L}(z^{-L} + 1) & z^{-L} & z^{-L} \\
(z^{-L} + 1)(z^{-2L} + 1) & z^{-2L} & z^{-2L}
\end{bmatrix}
\begin{bmatrix}
X_1 \\
X_2 \\
X_3
\end{bmatrix}
$ (2.85)

Network 3D: $
\begin{bmatrix}
Y_1 \\
Y_2 \\
Y_3
\end{bmatrix} =
\begin{bmatrix}
z^{-2L} & 1 & 1 \\
z^{-3L} - z^{-L} & z^{-L} & z^{-L} \\
z^{-4L} - 1 & z^{-2L} & z^{-2L}
\end{bmatrix}
\begin{bmatrix}
X_1 \\
X_2 \\
X_3
\end{bmatrix}
$ (2.86)

The final realisations are shown in Figure 2.16.

![Figure 2.16: Second-order all-pass networks.](image)

Chapter 2 Demultiplexing for OBP Satellites
2.9.4 Evaluation of Multiplication Roundoff Errors

The error due to the product quantisation produced by the multiplier $b_i$ can be considered equivalent to a set of noise samples $e_{b_i}(nT)$ superimposed on the signal at the output of the multiplier [Gol69]. If the noise samples are assumed to be uniformly distributed, the variance of $e_{b_i}(nT)$ is simply $q^2 / 12$ where $q$ is the quantisation step. The steady-state value of the variance of the noise samples at the output of the filter due to $e_{b_i}(nT)$ is:

$$\sigma_e^2 = \frac{q^2}{12(2\pi j)} \int H(z)H(z^{-1})z^{-1}dz$$

(2.87)

where $H(z)$ is the transfer function from the output of the multiplier $b_i$ to the filter output.

2.9.4.1 First-Order Sections

Figure 2.17: Noise model of first-order all-pass filters.

The noise model of the first-order all-pass section is shown in Figure 2.17. Simple analysis yields

$$\frac{Y_1}{E_b} \bigg|_{x_1=0} = H_1(z) = \frac{t_{12}}{1 - b_1t_{22}}$$

(2.88)

To compute the steady-state output noise $\sigma^2$ for each first-order section, the pertinent expressions for $t_{12}$ and $t_{22}$ are substituted and then the resultant expression is used to evaluate the contour integration in 2.87. In Table 2.3 the transfer function of each filter has been given.

The contour integration in (2.87) can be calculated by summing the residues of the poles which are inside the unit circle, i.e.

$$\sigma^2 = \frac{q^2}{12} \sum \text{residues}$$

(2.89)
Each of the realisations of the first-order sections, have residues at \( z = 0 \), \( z = b_1 \) and \( z = 1/b_1 \). Assuming \( b_1 \) is inside the unit circle, only residues at \( z = 0 \), \( z = b_1 \) are required. In the network of (2.76) there is a second-order term. For this realisation the residue at \( z = 0 \) is found by taking the first derivative of

\[
\frac{(1 - z^2)(z^2 - 1)}{(z - b_1)(1 - b_1z)}
\]

with respect to \( z \) and substitute \( z = 0 \) in the derivative [Opp75]. In Table 2.4 the variance of the noise samples for each first-order network has been given.

### Table 2.4: Noise variance of first-order filters.

<table>
<thead>
<tr>
<th>Network</th>
<th>1A</th>
<th>1B</th>
<th>1A_r</th>
<th>1B_r</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \sigma_o^2 )</td>
<td>( \frac{q^2}{6} )</td>
<td>( \frac{q^2}{6(1 - b_1)} )</td>
<td>( \frac{q^2}{12(1 - b_1^2)} )</td>
<td>( \frac{q^2}{6(1 + b_1)} )</td>
</tr>
</tbody>
</table>

Figure 2.18 depicts the normalised variance for each network as a function of the pole position. Knowing where the pole is located, the user can select the network which has the minimum noise.

#### 2.9.4.2 Second-Order Section

The noise model for this case is shown in Figure 2.19. There are two multipliers causing roundoff errors. If \( H_2(z) \) denotes the transfer function from the output of the multiplier \( b_2 \) and \( H_3(z) \) denotes the transfer function from the output of the multiplier \( b_3 \) to the filter output, then the steady-state output noise is given as

\[
\sigma_o^2 = \sigma_2^2 + \sigma_3^2
\]

where

\[
\sigma_i^2 = \frac{q^2}{12(2\pi J)} \int H_i(z)H_i(z^{-1})z^{-1}dz, \quad i = 2, 3.
\]
Hirano and Mitra [Mit74] have calculated the variances $\sigma_2^2$ and $\sigma_3^2$ for the second order sections. The results are shown in Table 2.5, where

\[
\begin{align*}
R_1 &= \frac{2}{1 - r^2} \\
R_2 &= \frac{2}{(1 - r^2)(1 - 2r \cos \theta + r^2)} \\
R_3 &= \frac{2}{(1 - r^2)(1 + 2r \cos \theta + r^2)} \\
R_4 &= \frac{2(1 - r^2 \cos 2\theta)}{(1 - r^2)(1 - 2r^2 \cos 2\theta + r^4)} \\
\end{align*}
\]

(2.92)

The performance of these networks are plotted in Figure 2.20 as a function of the pole position $\theta$ for four different values of the pole radius $r$. From the knowledge of $\theta$ and $r$, one can select a second-order filter configuration which has the lowest multiplication roundoff error.
2.9 A Package for Designing IIR Demultiplexers

Table 2.5: Noise variance of second-order sections.

<table>
<thead>
<tr>
<th>Network</th>
<th>3A</th>
<th>3B</th>
<th>3C</th>
<th>3D</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma^2$</td>
<td>$\frac{1}{6}R_1$</td>
<td>$\frac{1}{6}R_2$</td>
<td>$\frac{1}{6}R_3$</td>
<td>$\frac{1}{6}R_4$</td>
</tr>
</tbody>
</table>

Figure 2.20: Output noise of the second-order sections.

2.9.5 Software Implementation

The implemented filter design package, which includes the design and optimisation algorithms and the plotting routines, can be run on any computer with the Sun operating system (SunOs) Release 4.1.1 or higher. The package was written
in the C++ programming language. C++ is a powerful object-oriented program-
ing (OOP) language with some advanced features, such as operator overloading [Dan93]. Since the filter design algorithm makes extensive use of complex filters, with operator overloading one can redefine the existing algebraic operators to be used with complex numbers. Furthermore, the optimisation algorithm is mainly based on using vectors and matrices. C++ offers a neater approach to performing vector and matrix arithmetic.

2.10 Summary

After reviewing the basic principles of sampling rate conversion, different de-
multiplexing algorithms were analysed. The direct method demultiplexing has a prohibitively high hardware complexity and, hence, not suitable for practical applications. The FIR tree structure has a design modularity, a low hardware complexity, and a low computation rate. There are some disadvantages in using this structure. For example, a post-filter is required to perform a rational sampling rate conversion on the recovered FDM channels. To overcome this problem, the polyphase matrix GDFT can be used. Lack of flexibility and a high number of coefficients are the disadvantages in using the polyphase filter. The weighted overlap-add structure was presented next. This demultiplexer also has the same disadvantage of the polyphase demultiplexer; both have a high computation rate. However, the weighted overlap-add structure is restricted to decimation. This means that a separate post-filter will be required. The overall hardware complexity of using the combination of the weighted overlap-add structure and the post-filter, makes the implementation of this demultiplexer inefficient for practical applications.

Finally, the details of the developing a stand-alone package to design IIR prototype filters which can be used in a tree configuration were presented. Unlike the FIR tree demultiplexer in which a post-filter is required, no post-filtering is needed in the IIR tree demultiplexer. This makes the the complexity of the IIR demultiplexer less than that of the FIR demultiplexer. The detailed comparisons will be presented in Chapter 6.
Chapter 3
Frequency Synchronisation Algorithms

In this chapter, a survey of carrier frequency error detection algorithms will be presented. Under the light of this survey, a new heuristic rotational frequency error detection algorithm has been developed. With respect to algorithmic complexity and noise performance, the new algorithm is superior to other algorithms.

One of the important issues in communications via the proposed on-board processing Inmarsat-P or Iridium satellites is the presence of a large frequency error in the received signal. Several digital algorithms for detecting the frequency error in additive white Gaussian noise (AWGN) have been investigated [Mes79, Nat84, Alb89, Gar90, Hon90, Tak92, Cla93]. It is not always clear which algorithms are to be preferred for a given application. There are many criteria to consider, e.g. acquisition time, acquisition range and algorithmic complexity. Acquisition time is important in channels with short bursts of data, or in channels which are subject to deep fades. Depending on how much of the predicted Doppler shift is removed prior to frequency synchronisation, one has to consider the acquisition range of the synchronisation algorithms. Finally synchronisation algorithms with a low algorithmic complexity are desirable.

This chapter has been organised as follows:

An overview of Doppler shift which is a major source of the frequency error is presented first. Estimation of the Doppler shift by searching the frequency uncertainty range is then presented. To overcome the problem of high computation rate in the search technique, frequency acquisition by phase-locked loops (PLLs)
3.1 Features of Doppler Shift

In communications via nongeostationary satellites, since the distance between a satellite and a mobile changes, even if the mobile is stationary, the signals transmitted/received between them suffer the effects of Doppler shift. Doppler shift is a systematic change in frequency of a carrier wave and results when transmitter and receiver are moving relative to each other. For a typically nongeostationary satellite, the Doppler shift history is represented by a S-curve \([Kat92]\). The S-curve for the Inmarsat-P \([Inm94]\) and the Iridium \([Iri93]\) satellites are shown in Figure 3.1.

The S-curve becomes steeper as the satellite altitude decreases. For a satellite in circular orbits, the steepest part of the graph occurs at the point of the closest approach, and the observed frequency at this point is equal to the carrier frequency. Since the value of the Doppler shift is proportional to carrier frequency, the normalised Doppler shift, which is \(\frac{\text{Doppler shift}}{\text{carrier frequency}}\) has been used. The figure shows that the absolute values of the shift in the Iridium satellites are large compared with those of the Inmarsat-P satellites. From the figure, it is also seen that the values of the shift change rapidly in the Iridium satellites.

Figure 3.2 shows the change in Doppler shift as a function of time. This rapid change of frequency implies that not only static but also dynamic characteristics of
3.1 Features of Doppler Shift

Figure 3.1: Instantaneous Doppler shift.

Figure 3.2: Rate of change of Doppler shift with time.

Doppler shift in nongeostationary satellites have to be considered.

The contributors to Doppler shift are:

1. the angular rotational velocity of the Earth about its North-South axis;

2. the motion of the satellite in its orbital plane;

3. the velocity of the mobile.

The angular velocity of the Earth has small effect on frequency error, while the relative motion of the satellite and the mobile has a greater effect. For a circular orbit the maximum and minimum Doppler shift remain constant. To find these extreme cases, it is enough to find the contribution of each of the above factors and then add (subtract) them to find the maximum (minimum) Doppler shift.

Chapter 3 Frequency Synchronisation Algorithms
3.1 Features of Doppler Shift

As an example of the Doppler shift, a satellite in the Iridium constellation and a fixed station on the ground is considered. It is assumed that the carrier frequency is 1.5 GHz and the Iridium constellation is perfectly circular. The Doppler shift due to the rotational motion of the Earth [Dav90] in Guildford, with latitude $\phi = 51.241^\circ$ North, is

\[
\Delta f = \omega_e R_e \cos \phi \times \frac{f_o}{c} = 1.46 \text{ kHz} \tag{3.1}
\]

where

- $\omega_e$ is the angular velocity of the Earth and is approximately $2\pi / 24$ radian per day.
- $R_e$ is the average radius of the Earth and is assumed to be 6371 km.
- $c$ is the speed of light which has been taken as $3 \times 10^8$ m/s.

The worst case Doppler shift contribution due to the motion of the satellite is [Dav90]

\[
\Delta f = \frac{(GM_e)^{1/2}R_e f_o}{r^{3/2}c} = 33.258 \text{ kHz} \tag{3.2}
\]

where

- $G$ is the Universal Gravitational constant.
- $M_e$ is the mass of the earth ($GM_e = 3.986 \times 10^{14}$).
- $r$ is the distance from the geocentre to the satellite.

Therefore in theory, the worst case predicted maximum Doppler shift for the LEO satellite passing overhead in Guildford is about 34.72 kHz. In Iridium satellites the modulation scheme will be QPSK with a symbol rate of 2.4 kHz [Iri93]. The predicted Doppler shift is about 14.5 times the symbol rate. This Doppler shift is far beyond the acquisition range of the existing frequency synchronisation algorithms. Similar calculations for Inmarsat-P satellites show that the theoretical maximum Doppler shift experienced in Guildford is about 10.8 kHz, which is about 5 times the QPSK symbol rate of 2.4 kHz.
To overcome the problem of large Doppler shifts, a solution is to increase the data rate to such an extent that the Doppler shift becomes a fraction of the symbol rate. This option requires a higher transmission power, which may not be feasible for a hand-held terminal. Furthermore as the bit rate increases, the transmission bandwidth must also increase. The general trend in satellite communications is to decrease the transmission power and occupy less bandwidth. The practical solution is to remove the predicted Doppler shift and then use a synchronisation loop to remove the residual Doppler shift.

An approach to remove the predicted Doppler shift is that the OBP multi-beam satellite offsets all the signals received from (transmitted to) mobiles within the $n$-th beam by an average value $f_n$. The problem with this approach is that the above frequency offset is the same for a mobile which is at the centre of the beam and a mobile which is at the edge of the beam.

![Diagram of Doppler shift prediction](image)

Figure 3.3: Removing the predicted of the Doppler shift from the received signal.

A more efficient method to remove the predicted Doppler shift is shown in Chapter 3 Frequency Synchronisation Algorithms.
Figure 3.3. The assumption made is that the OBP satellite and the mobiles are equipped with global positioning system (GPS) receivers. Also, the satellite knows its own position from on-board Keplerian elements\(^1\). On the uplink, the satellite receives the coordinates of the mobile and, from on-board Keplerian elements, calculates its position. Therefore, an estimate of the Doppler shift is made. This estimate is used to control a frequency synthesiser which changes the frequency at which the received signal is down converted. With this approach, the predicted Doppler shift is removed and the demodulator will only synchronise to a residual frequency error. Similarly on the downlink, the satellite receives the coordinates of the mobile. The satellite again makes an estimate of the Doppler shift and, hence, adjusts the up conversion frequency such that when the signal arrives at the mobile, it has only residual frequency error.

In the following sections digital algorithms which can be used to detect the residual Doppler shift are discussed.

### 3.2 Frequency Search Techniques

Error detection by searching the frequency uncertainty range is based on the fact that if there is no frequency error, the measured average power at the output of the matched filter (MF) is maximum. Search can be performed in parallel or in series.

A block diagram of parallel search is shown in Figure 3.4. The received signal is fed to a bank of \(N\) single-sideband (SSB) demodulators. Each SSB demodulator shifts the received signal by a normalised frequency \(-2\pi n f_0 / f_s\), where \(f_s\) is the sampling frequency of the input signal and \(n f_0\) is the frequency step with which the spectrum of the received signal is shifted. The output of each SSB demodulator is then fed to a MF. Denoting the MF complex output by

\[
x[n] = x_r[n] + jx_i[n]
\]

(3.3)

where \(x_r[n]\) and \(x_i[n]\) are the real and the imaginary components of \(x[n]\), the average power at the output of each MF will be

\[
p_{nf_0} = \frac{1}{K} \sum_{k=0}^{K-1} x_{n k f_0}^2[n] + x_{q n f_0}^2[n], \quad n = 0, 1, \ldots
\]

(3.4)

where \(K\) is the averaging length of the integrate-and-dump (I&D) filters. The frequency \(n f_0\) at which the power is maximum, corresponds to the frequency error

\(^1\)Inclination angle, right ascension of ascending node (RAAN), eccentricity, argument of perigee, mean anomaly, mean motion and decay rate [Dav90].
v. Assume the parallel search must be performed in the frequency range of $f = 0$ to $f = R/r$, where $r$ is a positive integer number. It is further assumed that the accuracy of the frequency error estimation must be to $d$ decimal places. To determine the required hardware complexity a parameter $N$ is defined

$$N = \frac{10 \times R \times d}{r} + 1$$

which is the number of parallel stages in the search, i.e. $N$ SSB demodulators, $N$ MFs, and $N$ units to take the average power are required. As an example, with $R = 2.4$ kHz, $d = 3$, $r = 2$, $N$ will be 36000. Therefore, the implementation of parallel search on-board a satellite is not physically possible. To overcome this problem, a serial search can be performed. In serial search, the average power, $p_0$, at the output of the MF is determined. The spectrum of the received signal is then shifted by $f_0$ and the new average power, $p_1$, at the output of the MF is measured. The spectrum is further shifted by $2f_0$, and the new value of the average power, $p_2$, is measured. This is continued until the whole range of the frequency uncertainty is covered. The value of $nf_0$ at which the average power is maximum corresponds to the estimate of the frequency error.

Although the hardware complexity in the serial search is $N$ times lower, the estimation time is $N$ times slower than that with the parallel search. Therefore, serial search is not suitable for LEO/ICO satellites either.
In the following sections, more efficient frequency error detection algorithms are analysed.

### 3.3 Frequency Synchronisation by PLLs

A PLL can be used to acquire frequency errors. An example of frequency synchronisation by a second-order PLL in a BPSK modem is shown in Figure 3.5. For simplicity, the channel does not include AWGN, clock timing error or carrier phase error. The initial acquisition of this PLL is a problem, as the loop bandwidth \( B_L T \) is generally small to reduce carrier phase jitter due to noise. Consequently, this small loop bandwidth severely restricts the lock-in range. It is seen from the simulation results that fast acquisition of the PLL is only possible if there is a very small frequency error.

![Figure 3.5: Use of the a second-order PLL for frequency acquisition.](image)

### 3.4 Acquisition Aiding by AFC Loops

The major drawback of using PLLs for frequency acquisition is their very small pull-in (acquisition) range. An option to increase this range is to use an automatic frequency control (AFC) loop whose components are a frequency difference detector (FDD), a loop filter and a voltage controlled oscillator (VCO) which are arranged in a
feedback configuration. In the following sections, different algorithms implemented in AFC loops are presented.

### 3.4.1 Gardner Frequency Error Detector

![Gardner frequency discriminator and tracking loop.](image)

The model used to describe the frequency error correction is shown in Figure 3.6 in which the received signal plus noise $r(t)$ is applied to a frequency rotator. The rotated signal is applied to a signal MF and a frequency MF$^2$. The filtered complex output is then fed to a ML frequency error detector (FED) which generates an estimate of the frequency error $u_v(n)$. This error frequency is then smoothed by a loop filter to generate a trial frequency $\tilde{v}$, which will drive the frequency rotator and will, hence, close the feedback loop. The scaling factor $\beta$ controls the acquisition time of the loop. The higher $\beta$ is, the faster the loop acquires the error. However, the tracking jitter will be larger.

The impulse response of the frequency MF is found by taking the partial derivative of the transfer function of signal MF with respect to frequency $f$ [Gar90]

$$\frac{\partial H^*(f)}{\partial f} \Delta H^*(f) = \int_{-\infty}^{\infty} h(-t)(-j2\pi f)e^{-j2\pi ft} dt$$

$$= -j2\pi \int_{-\infty}^{\infty} t h(-t)e^{-j2\pi ft} dt \quad (3.6)$$

where $h(-t)$ is the baseband signal MF impulse response with its transfer function given by

$$H^*(f) = \int_{-\infty}^{\infty} h(-t)e^{-j2\pi ft} dt \quad (3.7)$$

$^2$As will be shown in Figure 3.7, a frequency MF has a frequency response which is the frequency derivative of the frequency response of the signal MF. By duality, a timing MF has an impulse response which is the time derivative of the impulse response of the signal MF.

*Chapter 3 Frequency Synchronisation Algorithms*
Hence, the impulse response of the frequency MF is $-j2\pi t h(-t)$

$$h_{FMF}(-t) = -j2\pi t h_{SMF}(-t)$$ (3.8)

whose time discrete representation is

$$h_{FMF}[nT] = -j2\pi nT h_{SMF}[nT]$$ (3.9)

The interpretation of (3.8) and (3.9) is that the same prototype filter can be used for both frequency MF and signal MF. To implement the frequency MF, each coefficient $h[n]$ is scaled by $2\pi n$. "$-j"$ simply means that the real and imaginary components of the rotated symbol are interchanged with the new imaginary component negated.

![Figure 3.7: PSD of the signal and frequency MFs.](image)

For the case of QPSK, the frequency error detection algorithm used in the Gardner frequency discriminator and tracking loop is [Gar90]

$$u_e[n] = \Re(p[n]p^*[n])$$ (3.10)

where

$u_e[n]$ is the estimate of the frequency error.

$n$ is the symbol index.
3.4 Acquisition Aiding by AFC Loops

$p[n]$ and $p_v[n]$ are the complex outputs of the signal and frequency MFs, respectively.

$p_v^*[n]$ is the complex conjugate of $p_v[n]$.

$\Re(x)$ is the real part of $x$.

Unlike the OQPSK FED which requires 2 samples per symbol for correct operation [Ahm92], the BPSK/QPSK ML FEDs need only 1 sample per symbol.

3.4.1.1 Frequency Error Detector S-Curve

The S-curve of any error detector is a graph of the variations of the averaged estimates $\hat{\Psi}$ at the output of the error detector as a function of the actual error $\Psi$. Gardner [Gar90] has derived the theoretical expressions in Table 3.1 for the S-curve of the detector in (3.10):

<table>
<thead>
<tr>
<th>Frequency range</th>
<th>$u_v$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0 \leq \nu T &lt; \alpha$</td>
<td>$\sin^2 \left[ \frac{\pi \nu T}{2\alpha} \right] + \frac{\pi}{4} \left[ 1 - \frac{\nu T}{\alpha} \right] \sin \left[ \frac{\pi \nu T}{\alpha} \right] \left( 1 + \cos \frac{2\pi \nu T}{\alpha} \right)$</td>
</tr>
<tr>
<td>$\alpha \leq \nu T &lt; 1 - \alpha$</td>
<td>1</td>
</tr>
<tr>
<td>$1 - \alpha \leq \nu T &lt; 1$</td>
<td>$\frac{1}{2} \left{ 1 + \sin^2 \left[ \frac{\pi}{2\alpha} (\nu T - 1) \right] - \frac{\pi}{4} \left[ 1 + \frac{C}{C} \right] \sin \left[ \frac{\pi}{\alpha} C \right] \right}$</td>
</tr>
<tr>
<td>$1 \leq \nu T &lt; 1 + \alpha$</td>
<td>$\frac{1}{2} \left{ \cos^2 \left[ \frac{\pi}{2\alpha} C \right] - \frac{\pi}{4} \left[ 1 - \frac{C}{C} \right] \sin \left[ \frac{\pi}{\alpha} C \right] \right}$</td>
</tr>
</tbody>
</table>

where $C = \nu T - 1$, and $T$ is the symbol period.

The theoretical curve and the simulation results are shown in Figures 3.8 and 3.9. The following can be deduced about the characteristics of the Gardner algorithm:

1. The S-curve has a piecewise structure and is periodic\(^3\) with a period of $1/(4\pi)$.
   The curve is odd-symmetry about zero frequency. If the average error $u_v$ is positive, the control loop infers $\nu > 0$ and attempts to increase the trial

\(^3\)The portion from $|3\pi/2|$ to $|2\pi|$ has not been shown.

---

Chapter 3 Frequency Synchronisation Algorithms
3.4 Acquisition Aiding by AFC Loops

Figure 3.8: Variations in the Gardner ML FED S-curve with rolloff.

Figure 3.9: Variations in the Gardner ML FED S-curve with timing error.

frequency $\tilde{v}$, thereby intending to reduce $v$. Similarly, if $u_r$ is negative, the loop infers $v < 0$ and attempts to increase $v$ by increasing $\tilde{v}$.

2. For QPSK, the S-curve is a function of the timing error $\tau$ and the rolloff factor
3.4 Acquisition Aiding by AFC Loops

\(\alpha\). As \(\alpha\) decreases, the slope of the curve increases. At zero rolloff, the curve becomes a step function. The S-curve is a function of timing error in the range of

\[-\alpha \leq \nu T < \alpha\]

For the same \(\alpha\), the slope decreases as the timing error increases. This means that the performance of the algorithm deteriorates in the presence of timing error, and fails to operate when the timing error is half the symbol period. Therefore, the timing error must be corrected before frequency synchronisation is performed. This feature makes the Gardner algorithm unsuitable for applications in which there is a timing error; there is no QPSK timing error synchronisers which can operate in the presence of frequency error. In [Ahm92] the simulation results for OQPSK has been shown, in which the Gardner algorithm is independent of timing error.

3. The shape of the S-curve is independent of phase error \(\theta\). Hence, frequency lock can be achieved without having a previous knowledge of the carrier phase.

4. The S-curves have a common zero crossing at zero frequency error. Therefore, the loop frequency error has a zero dc component at the equilibrium.

Gardner [Gar90] has shown that under the following conditions the frequency error detector is self-noise free:

1. zero frequency (\(\nu = 0\)) and zero timing errors (\(\tau = 0\));

2. overall pulse shape is Nyquist, so that the condition \(g[iT] = 0, i \neq 0\).

An example of the performance of the Gardner FED algorithm is shown in Figure 3.10.

3.4.2 Dual Filter Discriminator (DFD)

Alberty and Hespelt [Alb89] proposed the DFD algorithm. The block diagram is shown in Figure 3.11. It consists of two band-pass filters, \(H_p(f)\) and \(H_n(f)\), two square law devices and a summing junction. The band-pass filters are complex and, hence, consist of real and imaginary impulse responses

\[h_p = h_{pr} + jh_{pi}\]  \hspace{1cm} (3.11)

\(4\)The time index \([n]\) has been omitted from equations 3.11–3.21 for brevity.

Chapter 3 Frequency Synchronisation Algorithms
3.4 Acquisition Aiding by AFC Loops

Figure 3.10: Acquisition performance of the Gardner FED algorithm.

Figure 3.11: Dual Filter Discriminator.

and

$$h_n = h_{nr} + jh_{ni}$$  \hfill (3.12)

If the input is

$$s = x + jy$$  \hfill (3.13)

the output of the band-pass filter $H_p(f)$ is

$$s_p = h_p \otimes s$$
$$= (h_{pr} + jh_{pi}) \otimes (x + jy)$$
$$= (h_{pr} \otimes x - h_{pi} \otimes y) + j(h_{pr} \otimes y + h_{pi} \otimes x)$$  \hfill (3.14)
where ⊗ is the convolution operator. Similarly the output of the band-pass filter $H_n(f)$ is

$$s_n = (h_{nr} \otimes x - h_{ni} \otimes y) + j(h_{nr} \otimes y + h_{ni} \otimes x)$$  \hspace{1cm} (3.15)

The squared magnitude of $s_p$ is

$$|s_p|^2 = (h_{pr} \otimes x)^2 + (h_{pi} \otimes y)^2 - 2(h_{pr} \otimes x)(h_{pi} \otimes y)$$
$$+ (h_{pr} \otimes y)^2 + (h_{pi} \otimes x)^2 + 2(h_{pr} \otimes y)(h_{pi} \otimes x)$$  \hspace{1cm} (3.16)

and similarly the squared magnitude of $s_n$ is

$$|s_n|^2 = (h_{nr} \otimes x)^2 + (h_{ni} \otimes y)^2 - 2(h_{nr} \otimes x)(h_{ni} \otimes y)$$
$$+ (h_{nr} \otimes y)^2 + (h_{ni} \otimes x)^2 + 2(h_{nr} \otimes y)(h_{ni} \otimes x)$$  \hspace{1cm} (3.17)

Imposing the mirror image symmetry on $H_p(f)$ and $H_n(f)$

$$h_{pr} = h_{nr}$$  \hspace{1cm} (3.18)

and

$$h_{pi} = -h_{ni}$$  \hspace{1cm} (3.19)

Equation (3.17) can be written as

$$|s_n|^2 = (h_{pr} \otimes x)^2 + (h_{pi} \otimes y)^2 + 2(h_{pr} \otimes x)(h_{pi} \otimes y)$$
$$+ (h_{pr} \otimes y)^2 + (h_{pi} \otimes x)^2 - 2(h_{pr} \otimes y)(h_{pi} \otimes x)$$  \hspace{1cm} (3.20)

Subtracting (3.20) from (3.16) results in the error sample

$$u_f = 4(h_{pr} \otimes y)(h_{pi} \otimes x) - 4(h_{pr} \otimes x)(h_{pi} \otimes y)$$  \hspace{1cm} (3.21)

which may be implemented as shown in Figure 3.13. To calculate each output sample, four real convolutions, one real addition and three real multiplications are required. In practice, the multiplication by 4 is combined with the loop scaling factor $\beta$. Therefore, the number of real multipliers in the implementation of the DFD reduces to two.

An indication of the shape of the DFD S-curve can be obtained by considering the spectra shown in Figure 3.12. The pulse shape of the received signal is designated by $G(f)$ and is band-limited between $\left|\frac{(1 + \alpha)R}{2}\right|$. The spectrum of $H_p(f)$ is limited between $\left(\frac{1 - \alpha}{2}\right)R$ and $\left(\frac{3 + \alpha}{2}\right)R$. Since $H_n(f)$ is the mirror image of $H_p(f)$, its spectrum extends from $\left(-\frac{3 - \alpha}{2}\right)R$ to $\left(\frac{1 - \alpha}{2}\right)R$. The question which has to be answered now is, "what should the minimum sampling rate be for the correct operation of the DFD?" If the sampling rate is $2R$, the filters $H_p(f)$ and $H_n(f)$ overlap.
3.4 Acquisition Aiding by AFC Loops

The DFD output will be zero for all frequency errors. This problem can be solved by increasing the sampling rate to $4R$. With this constraint, the change in the S-curve is intuitively investigated next as a function of the frequency error.

Case I $v = 0$

The values of $u_p$ and $u_n$ are theoretically equal. The DFD output, $u_v$, is very close to zero in practice.

Case II $v = R$

As the error increases, the difference between $u_p$ and $u_n$ increases linearly. This
3.4 Acquisition Aiding by AFC Loops

difference will be maximum at \( v = R \). At this error, the S-curve is at its peak.

Case III \[ v = \frac{3}{2}R \]

Further increase in error will result in a decrease in \( u_f \). At \( v = \frac{3}{2}R \), some of \( G(f) \) passes the \( 2R \) frequency point. However, there is no aliasing.

Case IV \[ v = 2R \]

The decrease in \( u_f \) continues as the error increases. At \( v = 2R \), \( u_p \) and \( u_n \) are equal and, hence, \( u_f \) is zero. This means that at \( v = 2R \), the S-curve passes through zero.

The above discussions are also applicable if the error is in the negative direction. As mentioned earlier, the DFD is based on difference power measurements. Therefore, the phase error does not have any influence on the shape of the S-curve.

Referring to Figure 3.14, the following features can be deduced about the DFD S-curve:

1. There is zero frequency error at the output of the error detector in the absence of input frequency error; the algorithm is unbiased.

2. Tracking range is \( \pm R \).

3. S-curve is periodic with a period of \( 1 / (4R) \).

4. The maximum error between the theoretically triangular S-curve and the S-curve obtained in practice occurs at \( \pm R \) and is about 10.1%.

5. Timing and phase errors do not affect the S-curve in the linear region. Therefore, the DFD is used before bit timing and carrier phase synchronisation algorithms in all-digital demodulators.

6. Medium to high SNRs \( (E_s / N_0) \) do not affect the S-curve in the linear region\(^5\).

\(^5\)Simulation results for high SNRs have not been shown.

Chapter 3 Frequency Synchronisation Algorithms
3.4 Acquisition Aiding by AFC Loops

Figure 3.14: DFD characteristics.

(a) Effect of timing error

(b) Effect of phase error

(c) Effect of channel AWGN
3.4.3 Balanced Quadricorrelator (BQC)

The balanced quadricorrelator (BQC) algorithm has been studied in [Mes79, Nat84, Gar85a, Men90]. A simple derivation of the algorithm is as follows:

Assuming the received carrier phase is known, the complex envelope of the received signal is given by

\[ x_L[t] = \left( \sum_m a_m h_i[t - mT - \tau] + j \sum_m b_m h_q[t - mT - \tau] \right) e^{2\pi vt} \]  \hspace{1cm} (3.22)

where \( v \) is the frequency error. For conciseness in notation, (3.22) is written as

\[ x_L[t] = (a[t] + jb[t]) e^{2\pi vt} \]

\[ = a[t] \cos 2\pi vt - b[t] \sin 2\pi vt + j(b[t] \cos 2\pi vt + a[t] \sin 2\pi vt) \]  \hspace{1cm} (3.23)

The instantaneous phase of the received signal is

\[ \theta[t] = \tan^{-1} \left( \frac{y_c}{y_i} \right) \]  \hspace{1cm} (3.24)

The instantaneous frequency error is the time derivative of (3.24)

\[ \Delta \omega = \frac{d\theta[t]}{dt} \]  \hspace{1cm} (3.25)

or

\[ \Delta \omega = \frac{\dot{y}_c y_i - y_c \dot{y}_i}{y_i^2 + y_c^2} \]  \hspace{1cm} (3.26)

The discriminator, i.e. the part which generates the error signal, is known as the balanced quadricorrelator and is shown in Figure 3.15. The differentiators in both in-phase and quadrature channels are approximated by delay elements.

By simple analysis it can be shown that the output of the detector is

\[ u_v[t] = (a[t]a[t - T_d] + b[t]b[t - T_d]) \sin 2\pi v T_d \]

\[ + (b[t]a[t - T_d] - a[t]b[t - T_d]) \cos 2\pi v T_d \]  \hspace{1cm} (3.27)

where \( T_d \) is the timing delay of one sample. The first line is the desired error signal (zero-frequency component) and the terms on the second line contribute to the self-noise (ripple). The ripple vanishes only if the BPSK modulation scheme is used\(^6\). The simulation result for a QPSK modem is shown in Figure 3.16. There is a large self-noise even in the absence of AWGN. Similar result for OQPSK has been shown

\(^6\text{In BPSK, } b[t] = b[t - T_d] = 0\)
3.4 Acquisition Aiding by AFC Loops

The self-noise can be reduced by additional filtering of the output of the detector. In [Nat84] the use of an integrate-and-dump (I&D) filter before the loop filter to limit the input noise has been suggested. With this approach, as the I&D filter length increases, the delay increases. Increasing the delay may prevent the loop acquiring the signal.

![Figure 3.15: Discrete time equivalent of BQC.](image)

![Figure 3.16: Frequency acquisition with balanced quadricorrelator.](image)
3.4.4 Cross-Dot Product Frequency Error Detector

A variation of the BQC is shown in Figure 3.17 and is known as the cross-dot product (CDP) frequency error detector (FED). In this scheme, both the cross and dot products of adjacent signal samples are computed and multiplied together to generate the output error $u[k]$ which is represented by

$$u[k] = (x[k - 1]x[k] + y[k - 1]y[k]) \times (x[k - 1]y[k] - x[k]y[k - 1]) \quad (3.28)$$

By performing some algebraic manipulations, it can be shown that the output of the detector is

$$u_v[t] = (a[t]a[t - T_d] + b[t]b[t - T_d])(\sin 2\pi v T_d + \cos 2\pi v T_d) + (b[t]a[t - T_d] - a[t]b[t - T_d])(\cos 2\pi v T_d - \sin 2\pi v T_d) \quad (3.29)$$

The discriminator curve is shown in Figure 3.18. The normalised tracking range is approximately 0.12. Similar to the performance of the BQC, the cross-dot product algorithm has a large self-noise for (O)QPSK modulation schemes.

3.5 Feedforward Carrier Synchronisation

Common to all the work mentioned so far, is the fact that they operate in feedback mode. In applications where data are transmitted in short bursts, or in applications where a fast acquisition after a deep fade is required, the acquisition time of an AFC loop may be too long. In such cases a technique which employs feedforward
(FF) frequency estimation, as shown in Figure 3.19, may be advantageous. In the following sections, FF frequency error estimation in the time and frequency domains are presented.

3.5.1 Rotational Frequency Error Detectors

The derivation of rotational frequency error detectors is based on a differential phase measurement between succeeding complex samples. Assuming the Doppler shift is constant, the complex signal is rotated by a constantly incrementing angle $2\pi v$, where $v$ is the frequency error normalised to the sampling rate. For the simple case of transmitting an unmodulated signal, the phase difference between two adjacent received samples divided by $2\pi$ is the normalised angular frequency error. If the random data are modulated, the phase of the received complex signal is affected by the actual frequency error and the random nature of the transmitted data. For phase
estimation, it is necessary that the phase variations due to modulation to be removed first.

An ad hoc feedforward frequency error estimation algorithm is [Cla93]

\[ 2\pi F_e T = \frac{1}{M} \arg \left[ \sum_{k=-K+1}^{K} \beta_k v_k \right] \]  \hspace{1cm} (3.30)

where

- BPSK: M=2, (O)QPSK: M=4.
- \( v_k = F(|z_k|)F(|z_{k-1}|)\exp(jM(\phi_k - \phi_{k-1})). \) \( |z_k| \) is the magnitude of the \( k \)-th complex sample of the received signal.
- \( \beta_k \) is a set of optimised coefficients given by

\[ \beta_k = \frac{6}{K(K^2 - 1)} \left[ \frac{K^2 - 1}{4} - k(k - 1) \right], \quad K = 2K' + 1. \]  \hspace{1cm} (3.31)

- \( K \) is the length of the I&D filter.
- \( F(u) \) is a function to raise \( u \) to an arbitrary power.
- \( \phi_k \) is the argument of \( z_k \).

The idea behind the above algorithm is that \( M(\phi_k - \phi_{k-1}) = [2\pi MFT] \mod 2\pi \) in the absence of noise. The block diagram is shown in Figure 3.20. This algorithm has similarities with the Viterbi and Viterbi7 carrier synchronisation algorithm [Vit83]. The acquisition range is limited to \( \pm \frac{R}{2M} \).

![Block diagram of the feedforward frequency estimation structure.](image)

Figure 3.20: Block diagram of the feedforward frequency estimation structure.

Figure 3.21 shows simulation results for QPSK. The solid line is the Cramer-Rao bound [Bel90]

\[ 4\pi^2 \text{var}(\hat{\nu}T) = \frac{12}{2L(L^2 - 1)10^{8\text{NR/10}}} \]  \hspace{1cm} (3.32)

See chapter 5 for the analysis of the Viterbi and Viterbi algorithm.
where $L$ is the averaging length. Simulation results show that the Cramer-Rao bound can be met for high SNR. The function $F(|z|)$ can be optimised. If this nonlinearity is selected to be 1, the variance is closer to the bound at any given SNR.

### 3.5.2 A New Rotational FED

A new rotational frequency error detector is proposed which works in burst mode. The operation is as follows:

At the head of the burst (frame), a preamble consisting of samples of an unmodulated carrier are placed. This guarantees that the rotation of the carrier is $2\pi v$ from one sample to the other, where $v$ is the normalised frequency error. At the receiver, the detector makes an estimate of the error by operating on the unmodulated carrier only. Following the estimation of the frequency error, the preamble is removed and the data is rotated by $2\pi v_e$, where $v_e$ is the estimated normalised frequency error. The details of the operation of the FED are shown in the Table 3.2 in which $\alpha_1$ is the previous angle and $\alpha_2$ is the present angle. With the conditions specified, it is possible to detect frequency errors less than half the sampling frequency.

An implementation of the proposed FED is shown in Figure 3.22. The output of the MF is rotated by $2\pi v_e$. In noisy channels it is essential to perform averaging to estimate the error. Before demodulating the data, the preamble is removed from the frame. A design example of the acquisition response is shown in Figure 3.23.
### Table 3.2: Operation of the proposed heuristic FED.

<table>
<thead>
<tr>
<th>case</th>
<th>$\alpha_1$ and $\alpha_2$</th>
<th>estimated frequency $\nu_e$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$-\pi &lt; \alpha_1 &lt; -\pi/2$&lt;br&gt;$\pi/2 &lt; \alpha_2 &lt; \pi$</td>
<td>$\nu_e = (-2\pi + \alpha_2 - \alpha_1)/(2\pi T)$</td>
</tr>
<tr>
<td>2</td>
<td>$\pi/2 &lt; \alpha_1 \leq \pi$&lt;br&gt;$-\pi &lt; \alpha_2 &lt; -\pi/2$</td>
<td>$\nu_e = (2\pi - \alpha_1 + \alpha_2)/(2\pi T)$</td>
</tr>
<tr>
<td>3</td>
<td>$-\pi/2 &lt; \alpha_1 &lt; 0$&lt;br&gt;$0 &lt; \alpha_2 &lt; \pi/2$</td>
<td>$\nu_e = (\alpha_2 - \alpha_1)/(2\pi T)$</td>
</tr>
<tr>
<td>4</td>
<td>$0 &lt; \alpha_1 &lt; \pi/2$&lt;br&gt;$-\pi/2 &lt; \alpha_2 &lt; 0$</td>
<td>$\nu_e = (\alpha_2 - \alpha_1)/(2\pi T)$</td>
</tr>
</tbody>
</table>
| 5    | $0 < \alpha_1 < \pi/2$<br>$-\pi < \alpha_2 < -\pi/2$ | $f_1 = (2\pi - \alpha_1 + \alpha_2)/(2\pi T)$<br>$f_2 = (\alpha_2 - \alpha_1)/(2\pi T)$<br>$\nu_e = f_1$ if $|f_1| < |f_2|$
|      |                  | $\nu_e = f_2$ if $|f_1| > |f_2|$ |
| 6    | $-\pi < \alpha_1 < -\pi/2$<br>$0 < \alpha_2 < \pi/2$ | $f_1 = -(2\pi - \alpha_2 + \alpha_1)/(2\pi T)$<br>$f_2 = (\alpha_2 - \alpha_1)/(2\pi T)$<br>$\nu_e = f_1$ if $|f_1| < |f_2|$
|      |                  | $\nu_e = f_2$ if $|f_1| > |f_2|$ |
| 7    | $-\pi/2 < \alpha_1 < 0$<br>$\pi/2 < \alpha_2 < \pi$ | $f_1 = -(2\pi - \alpha_2 + \alpha_1)/(2\pi T)$<br>$f_2 = (\alpha_2 - \alpha_1)/(2\pi T)$<br>$\nu_e = f_1$ if $|f_1| < |f_2|$
|      |                  | $\nu_e = f_2$ if $|f_1| > |f_2|$ |
| 8    | $\pi/2 < \alpha_1 < \pi$<br>$-\pi/2 < \alpha_2 < 0$ | $f_1 = (\alpha_2 - \alpha_1)/(2\pi T)$<br>$f_2 = (2\pi - \alpha_1 + \alpha_2)/(2\pi T)$<br>$\nu_e = f_1$ if $|f_1| < |f_2|$
|      |                  | $\nu_e = f_2$ if $|f_1| > |f_2|$ |
| 9    | otherwise        | $\nu_e = (\alpha_2 - \alpha_1)/(2\pi T)$ |

![Figure 3.22: Implementation of the proposed rotational FED.](image)

---

**Chapter 3 Frequency Synchronisation Algorithms**
3.5 Feedforward Carrier Synchronisation

The initial delay of $55T$ is due to the averaging loop filter. The good tracking performance is due to using a cascade of two smoothing loop filters.

In terms of hardware and algorithmic complexity, the advantages of the above detector are:

1. Unlike the Gardner ML algorithm\(^8\) or the Alberty and Hespelt DFD algorithm\(^9\), no FIR filtering is required. Therefore, the implementation is simpler.

2. Unlike the Classen et. al algorithm\(^{10}\), optimum coefficients $\beta_k$ or nonlinearities are not required. Therefore, the computational burden is lower.

![Figure 3.23: Estimation performance of the proposed rotational FED.](image)

Assuming the channel introduces a phase error $\theta$, the measured phases are

$$\alpha_1 = \theta + 2\pi v_1 T$$  \hspace{1cm} (3.33)

and

$$\alpha_2 = \theta + 2\pi v_2 T$$  \hspace{1cm} (3.34)

Referring to the Table 3.2, from $|\alpha_2 - \alpha_1|$, the phase difference $\theta$ disappears. Therefore, the phase error has no influence on the proposed frequency error detector.

\(^8\)See section 3.4.1.
\(^9\)See section 3.4.2.
\(^{10}\)See section 3.5.1.
and, hence, the algorithm can work without the previous knowledge of carrier phase error. In section 3.6, the noise performance of the developed FED will be compared against other algorithms surveyed in this chapter.

3.5.3 Frequency Error Detection by DFT Techniques

Takeuchi and Kobayashi [Tak92] proposed frequency synchronisation techniques for burst mode modems to be used in digital mobile satellite communications. Their feedforward estimation techniques are based on Discrete Fourier transform (DFT) and the sliding DFT (S-DFT) operations. The estimation of unmodulated carrier frequency is accomplished by finding a frequency component having a peak power. The power level estimation is performed by a DFT operation. In the estimation of carrier frequency based on DFT operation, there is a maximum frequency estimation error of \( F_d / 2 \) when the frequency resolution is \( F_d \). In order to improve the accuracy of the estimated frequency an interpolation method, in which both the peak power level and its neighbouring power levels are obtained, is used. The operation is as follows:

For a block of \( N \) samples, the DFT is defined as

\[
Y_k = \sum_{i=0}^{N-1} y[i] W^{ik} \quad (-N/2 < k < N/2)
\]  

(3.35)

where \( W = e^{-j2\pi k/N} \), and the frequency resolution is \( F_d = 1 / (NT_s) \). If \( Y_k \) has the maximum power level at \( k = k_{max} \), the unknown frequency error \( f \) is interpolated according to whether the power at \( k_{max} + 1 \) is higher than that at point \( k_{max} - 1 \)

\[
f = \begin{cases} 
    k_{max} + |Y_{k_{max}+1}| / (|Y_{k_{max}}| + |Y_{k_{max}+1}|), & F_d \cdot |Y_{k_{max}+1}| > |Y_{k_{max}+1}| \\
    k_{max} - |Y_{k_{max}-1}| / (|Y_{k_{max}}| + |Y_{k_{max}-1}|), & F_d \cdot |Y_{k_{max}-1}| < |Y_{k_{max}-1}|
\end{cases}
\]  

(3.36)

or

\[
f = \begin{cases} 
    k_{max} + |Y_{k_{max}+1}| / (|Y_{k_{max}}| + |Y_{k_{max}+1}|), & F_d \cdot |Y_{k_{max}+1}| > |Y_{k_{max}+1}| \\
    k_{max} - |Y_{k_{max}-1}| / (|Y_{k_{max}}| + |Y_{k_{max}-1}|), & F_d \cdot |Y_{k_{max}-1}| < |Y_{k_{max}-1}|
\end{cases}
\]  

(3.37)

The above technique is very useful when the timing error is known in advance. However, in practice the modem may not know the precise receiving time of the burst signal. In this case it is required to perform the DFT processing at every sample time. The hardware complexity prohibits the implementation of the above approach even when an FFT is used instead of DFT.

In the same paper by Takeuchi and Kobayashi, the adoption of the sliding DFT (S-DFT) has been proposed as an alternative calculation technique for the DFT to
be used for the fast frequency acquisition and the burst detection. The derivation of
the S-DFT is as follows:

By denoting $i$-th sample of input signal as $y[i]$, the frequency component at $k$-th
frequency cell obtained by the DFT operation performed to $N$ samples $y[m - N + 1], y[m - N + 2], \cdots, y[m]$ is expressed by

$$ Y_k[m] = \sum_{i=0}^{N-1} y[m - N + 1 + i]W^{ik} \quad (3.38) $$

Then the frequency component at $k$-th frequency cell one sample before is given as

$$ Y_k[m - 1] = \sum_{i=0}^{N-1} y[m - N + i]W^{ik} \quad (3.39) $$

Hence the following relation is derived from (3.38) and (3.39)

$$ Y_k[m] = W^{-k} \{ Y_k[m - 1] + y[m] - y[m - N] \} \quad (3.40) $$

By using the above relation, the frequency component can be updated at every
sample with less calculation than that required in the DFT or the FFT.

![Figure 3.24: Acquisition performance of the sliding DFT FED.](image)

The simulation results of a modem with the specifications given in Table 3.3 are
shown in Figure 3.24. It can be seen that the sliding DFT FED model has acquired
the frequency error after a delay of 8 samples period when the timing error is zero.
Table 3.3: Summary of the parameters used in simulating the (S)DFT FEDs.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmit filter</td>
<td>Nyquist with 50% rolloff</td>
</tr>
<tr>
<td>Modulation</td>
<td>QPSK</td>
</tr>
<tr>
<td>DFT length</td>
<td>8 points</td>
</tr>
<tr>
<td>Preamble length</td>
<td>96 samples</td>
</tr>
<tr>
<td>Frequency error</td>
<td>0.1R</td>
</tr>
<tr>
<td>Timing error</td>
<td>0.5T</td>
</tr>
<tr>
<td>SNR</td>
<td>∞</td>
</tr>
<tr>
<td>Number of samples per symbol</td>
<td>4</td>
</tr>
</tbody>
</table>

The delay increased to 10 samples period when the timing error was changed to 0.5T, where T is the symbol period. With the sliding DFT, the timing error only delays the signal and the signal is not distorted. Following the detection of the error \( f_0 \), the preamble is removed from the data frame and the frequency of the modulated signal is rotated by \(-f_0\).

![Figure 3.25: Acquisition performance of the DFT FED.](image)

Under the same specifications, the performance of the DFT FED was simulated (see Figure 3.25). This FED maintains the frequency error for only 96/8=12 samples. This is a disadvantage over sliding DFT FED in which the error is maintained over the full length of the preamble. This means that the sliding DFT requires \( N \) times
less preamble than the DFT where $N$ is the DFT length. Furthermore since with the DFT FED the timing error distorts the data, the FED must be switched off as soon as the preamble has finished. This is another disadvantage over the sliding DFT FED which does not require switching. For these reasons, the DFT FED has been excluded from further considerations.

### 3.6 Noise Performance of the FB and FF FEDs

It is interesting to compare the algorithms studied so far for their noise performance. The variance of the frequency error (actual error - estimated error) in the presence of AWGN is measured when the loop has acquired the error. The simulation results are shown in Figure 3.26. The balanced quadricorrelator and the cross-dot product algorithms suffer from a large self-noise. The difference between their performance is substantial at low SNR. Due to the excessive self-noise, these two algorithms are excluded from further considerations. The remaining two candidates are the DFD and the ML FEDs. It is seen from the results that the DFD outperforms the Gardner ML FED at moderate to high SNR. This is due to the fact the DFD is self-noise free, but the ML algorithm is only self-noise free when the frequency error is zero.

![Graph showing noise performance of FB and FF FEDs](image)

**Figure 3.26**: Noise performance of FB and FF FEDs.

With FF algorithms the output must be averaged. The longer the averaging length, the more accurate the results will be. On the other hand, long averaging
time means a long acquisition time. For comparison purposes, the averaging time of the FF algorithms was selected to be 125T. The sliding DFT method has a better noise performance than DFD at all values of SNR. It is only at low SNR that the performance of the Gardner ML algorithm is better than that of the sliding DFT algorithm. The most important result in the above figure is the performance of the proposed FED. At all SNRs this algorithm has a better noise performance than all other techniques. This performance is due to the very low error in detecting the frequency error. Furthermore, the other factor which contributes to this superior performance is the computation rate which is considerably lower than that of other algorithms.

3.7 Summary

In this chapter several algorithms for estimating the carrier frequency of signal were reviewed. The choice of the algorithm depends on the application. In general the following recommendations are made:

- Parallel search requires a large hardware complexity. It is not physically possible to implement for satellite communications in general.

- Serial search is very slow to detect the frequency error. It is not suitable for implementation in LEO/ICO satellites.

- The small acquisition range of PLLs prohibits their use as a means to detect frequency errors in LEO/ICO satellites.

- Balanced quadricorrelator and cross-dot product frequency error detectors can acquire small to medium range frequency errors. Their implementation is very simple. Large self-noise is a major drawback in using these algorithms.

- If the frequency error must be estimated from the received signal, without preambles, the Alberty and Hespelt DFD and the Gardner ML FED can be used. The advantage of using the DFD is that it has the maximum acquisition range. The ML algorithm has a useful acquisition range less than the rolloff factor times the symbol rate. Unlike the DFD which has a constant gain in its acquisition range, the variations in the gain of the ML algorithm in the above range is nonlinear. The DFD is always self-noise free at high SNR, while the Gardner FED is self-noise free under certain conditions.
3.7 Summary

- The DFT and the sliding DFT FEDs operate on preambles. The latter requires a shorter preamble than the former. In the presence of timing error, the DFT FED fails to operate.

- The feedforward algorithm by Classen et al. operates on modulated data. It is simple to implement. Like other rotational frequency error detectors, this algorithm has an acquisition range less than \( R / (2M) \). This small acquisition range is a drawback in using the above algorithm.

- The proposed algorithm works on preambles. With respect to acquisition time, algorithmic and hardware complexity, and noise performance the algorithm is superior to all the above FEDs. The acquisition time is ideally (at \( SNR = \infty \)) 2 samples period. Only current and previous phases of the complex signal are compared. In digital implementations a lookup table is required to determine the angle of the complex signal. For arithmetic operations only adders and subtractors are required. The acquisition range is maximum. The algorithm works without the prior knowledge of the phase error.

In brief, from the algorithms surveyed, the Alberty and Hespelt DFD and the proposed FED will be considered for further investigation in Chapter 6. Although the sliding DFT FED, similar to the proposed FED, operates on preamble, the number of computations required to detect the error in the sliding DFT algorithm is higher. Therefore, the sliding DFT algorithm will not be considered further.

The results presented in this chapter were based on using a single-channel modem. In Chapter 6 the selected algorithms will be compared for detecting the frequency error in the received FDM signal which consists of 8 channels.
Chapter 4

Symbol Timing Synchronisation Algorithms

In this chapter, different timing error detection and correction algorithms will be surveyed. The optimisation of the modified Mueller and Müller (mM&M) algorithm, by self-noise cancellation, will be presented. The optimised mM&M algorithm has a very fast acquisition time which is a desirable feature in the proposed Iridium and Inmarsat-P satellites.

Clock timing synchronisation is concerned with determining the optimum sampling time at the output of the matched filter. Various algorithms have been proposed in the literature to detect the timing error [Gar88a, Kob71, Jes91, Gar85b, Lin72, Moe88, Oer90]. One way to accomplish timing synchronisation is by synchronising the receiver and transmitter clocks to a master clock. This may be the case for radio communication systems that operate in the very low frequency (below 30 kHz) band [Pro94]. Another approach to achieve timing synchronisation is to multiplex a timing signal along with the data. To extract the timing signal, a narrow band filter centred on the transmitted clock frequency is used. This technique is simple to implement. However, there are several disadvantages. One is that the transmitter must allocate some of its available power to transmit the timing signal. Another is that a fraction of the bandwidth is allocated to transmitting the timing signal.

The most important technique for timing recovery is by self-synchronisation in which the timing signal is extracted from the received data signal. The timing error detection algorithms surveyed in this chapter are:
- the decision-directed (DD) and the non data-aided search techniques;
- the maximum-likelihood (ML) tracking algorithm;
- the two-point difference tracker;
- the Gardner algorithm;
- the DD minimum mean-square error (MMSE) algorithm;
- the data transition tracking loop (DTTL);
- the modified Mueller and Müller (mM&M) algorithm.

Unlike the carrier phase and frequency correction\(^1\), which involves the use of a voltage controlled oscillator followed by a complex multiplication, timing error correction has received more attention due to its complex operation [Gar88a, Ver93, Yim92a]. The timing error correction techniques presented in this chapter are:

* the Verdin & Tozer polyphase interpolator;
* the Yim polyphase interpolator.

The comparison criteria for the timing synchronisation algorithms studied are the algorithmic complexity, the acquisition time, the error variance, and the self-noise. The comparison criterion for timing error correction methods is their hardware complexity. For the proposed Iridium and Inmarsat-P satellites, a synchroniser with a rapid acquisition time must be used. To avoid unnecessary computations, algorithms which have the minimum of algorithmic complexity must be chosen. One of the factors which affects the performance of a modem, is the self-noise of the timing error detector. In this respect, an error detection algorithm which is free from self-noise is preferred. A novel self-noise cancellation technique will be presented for the modified Mueller and Müller timing error detector (TED). With the presented technique, the above algorithm can acquire the error in as low as 10-15 symbols without spurious locks.

Before analysing different timing error detection algorithms, it is useful to review the signal representation in an all-digital modem and the principles of the ML signal estimation.

\(^1\)For carrier phase correction see Chapter 5.
4.1 Signal Representation

Assuming the satellite channel adds white Gaussian noise to the transmitted signal and does not introduce distortion and interference, the complex envelope of the received signal format is [Jes91]

\[
\begin{align*}
    r(t, \Psi) &= \left[ \sum_{n=1}^{N} a_n h_r(t - nT - \tau) + j \sum_{n=1}^{N} b_n h_i(t - nT - \varepsilon T - \tau) \right] e^{-j(\theta + 2\pi v t)} + w(t) \\
    &= s(t, \Psi) + w(t) \quad (4.1)
\end{align*}
\]

where

- \( a_n \) and \( b_n \) are the real and imaginary components of the transmitted complex data symbols.
- \( h_r(t) \) and \( h_i(t) \) are the real and the imaginary baseband pulses normalised to unit energy.
- \( 1/T \) is the symbol rate.
- \( \varepsilon \) is the stagger coefficient.
- \( \theta, \nu, \tau \) are the actual unknown phase, frequency and the symbol timing.
- \( w(t) \) is white, stationary, Gaussian noise of two-sided spectral density \( N_0 / 2 \) which is added to the signal at the receiver front end.
- \( \Psi \) is the vector of the unknowns \( \nu, \theta, \tau, a_n, b_n \).

The above signal model for BPSK/(O)QPSK modulation schemes is interpreted as:

**BPSK:** \( h_r(t) = h(t), h_i(t) = 0, \varepsilon = 0 \)

\[
r(t) = \sum_{n=1}^{N} a_n h(t - nT - \tau) e^{-j(\theta + 2\pi v t)} + w(t)
\]

**QPSK:** \( h_r(t) = h_i(t) = h(t), \varepsilon = 0 \)

\[
r(t) = \sum_{n=1}^{N} (a_n + j b_n) h(t - nT - \tau) e^{-j(\theta + 2\pi v t)} + w(t)
\]
4.1 Signal Representation

QPSK: \( h_r(t) = h(t), h_i(t) = h(t - T/2), \varepsilon = 1/2 \)

\[
r(t) = \left[ \sum_{n=1}^{N} a_n h(t - nT - \tau) + j \sum_{n=1}^{N} b_n h(t - nT - T/2 - \tau) \right] e^{-j(\theta+2\pi n)} + w(t)
\]

Assuming the frequency response of the transmit and the receive filters are root raised-cosine, the overall pulse shape is

\[
g(t) = \frac{\sin(\pi t / T)}{\pi t / T} \cos(\alpha \pi t / T)
\]

and its Fourier transform is

\[
G(f) = \begin{cases} 
T, & 0 \leq |f| \leq \frac{1-\alpha}{2T} \\
\frac{T}{2} \left[ 1 - \sin \left( \frac{\pi T}{\alpha} (f - \frac{1}{2T}) \right) \right], & \frac{1-\alpha}{2T} < |f| \leq \frac{1+\alpha}{2T} \\
0, & \frac{1+\alpha}{2T} < |f| 
\end{cases}
\]

where \( \alpha (0 \leq \alpha \leq 1) \) is the rolloff factor of the pulse shape.

![Impulse response](image)

Figure 4.1: ISI free transmission of band-limited impulses.

The pulse in (4.2) has the important characteristic that it is zero for nonzero sampling instances, i.e. \( g(kT) = 0, k \neq 0 \). This feature makes the transmission of band-limited impulses inter-symbol interference (ISI) free (see Figure 4.1).
4.2 Maximum-Likelihood (ML) Estimation

On page 93, it was mentioned that the important technique of timing error recovery is by self-synchronisation. This is also true for carrier phase and frequency synchronisation. With self-synchronisation, the receiver extracts the desired information from the received signal. A powerful analytical tool for self-synchronisation is the maximum-likelihood parameter estimation, which maximises the probability density function (PDF) of the unknown parameter $\Psi$ over an observation period $T_0$.

The PDF is conditioned on $\Psi$, where $\Psi$ is trial value of $\Psi$. The purpose of ML parameter estimation is to find that parameter $\Psi$ which maximises the likelihood of $\Psi$ being the same as $\Psi$. The PDF of the $\Psi$ conditioned on $\Psi$ is expressed by the likelihood function $L(\Psi)$, which is defined by [Gar88a, page 88]

$$L(\Psi) = C_1 \exp \left[ -\frac{C_2}{N_0} \int_{t=0}^{T_0} |r(t) - s(t, \Psi)|^2 dt \right]$$

where $C_1$ and $C_2$ are constants. Substituting (4.1) in the ML expression results in

$$L(\Psi) = C_1 \exp \left[ -\frac{C_2}{N_0} \left[ \int_{t=0}^{T_0} |r(t)|^2 dt + \int_{t=0}^{T_0} |s(t, \Psi)|^2 dt - 2 \int_{t=0}^{T_0} \Re[r(t)s^*(t, \Psi)] dt \right] \right]$$

where $\Re[x]$ is the real part of $x$, and $s^*(x)$ is the complex conjugate of $s(x)$. The first integral is the energy of the received signal plus noise and is not influenced by $\Psi$. That integral can be considered as a constant $C_e$. The second term can also be treated as a constant if $\Psi$ is the carrier phase or frequency (the signal has the same energy for any phase or frequency). If $\Psi$ is the timing error $\tau$, the second term will no be a constant. However over an observation time much greater than the symbol period, the second term will be a weak function of $\tau$. It is also treated as a constant $C_t$, which is merged with $C_e$ to form $C_3$. Therefore, Equation (4.5) simplifies to

$$L(\Psi) = C_3 \exp \left[ \frac{2C_2}{N_0} \int_{t=0}^{T_0} \Re[r(t)s^*(t, \Psi)] dt \right]$$

The integral in (4.6) is the real part of the correlation between $r(t)$ and $s(t, \Psi)$. In deriving the ML algorithms it is more convenient to work with the log-likelihood function, $\Lambda(\Psi)$, rather than the likelihood function, $L(\Psi)$, which has an exponential term. Any value of $\Psi$ which maximises $L(\Psi)$ also maximises $\ln[L(\Psi)]$. Therefore,

$$\Lambda(\Psi) = \frac{2C_2}{N_0} \int_{t=0}^{T_0} \Re[r(t)s^*(t, \Psi)] dt$$
4.2 Maximum-Likelihood (ML) Estimation

Since the constant term ln(C3) vanishes in the process of maximisation, it has been dropped from (4.7). By substituting (4.1) in the real part of the correlation integral, the following is obtained

\[ R(N, \tilde{\phi}) = \Re \left[ \exp(-j\tilde{\phi}) \sum_{n=1}^{N} \tilde{a}_n p(n) \right] + \Im \left[ \exp(-j\tilde{\phi}) \sum_{n=1}^{N} \tilde{b}_n q(n) \right] \quad (4.8) \]

where

\[ p(n, \tilde{\tau}) = \int_{-\infty}^{\infty} r(t) h(t - nT - \tilde{\tau}) dt \]
\[ q(n, \tilde{\tau}, c) = \int_{-\infty}^{\infty} r(t) h(t - nT - cT - \tilde{\tau}) dt \quad (4.9) \]

and \( \Im[x] \) is the imaginary part of \( x \). By defining \( c_n = a_n + jb_n \), (4.8) becomes

\[ R(N, \tilde{\phi}) = \Re \left[ \exp(-j\tilde{\phi}) \sum_{n=1}^{N} c_n^* p(n, \tilde{\tau}) \right] \quad (4.10) \]

The result in (4.10) can be used to determine the timing error by a search technique. The decision-directed ML search algorithm to determine an estimate of the timing error is given by [Gar88a]

\[ \tilde{\tau} = \max_{n=1}^{N} \left\{ \Re \{ c_n^* p(n) \} \right\} \quad (4.11) \]

A search is performed over all trial values of \( \tilde{\tau} \) in the symbol interval. A parallel search is shown in Figure 4.2.

If there are \( N \) samples per symbol (SPS) period, the maximum timing error that can occur is when \( N/2 \) of the received samples are in error. To determine the error, the output of the matched filter is sampled with delays of 0, 1, 2, \( \cdots \), \( NT/2 \), where \( T \) is the symbol period. The maximum value of \( \Lambda(\tau) \) is obtained by integrating \( p(n)\tilde{c}_n^* \) in (4.11) for each branch (The integrate-and-dump (I&D) operation in Figure 4.2). The data corresponding to the branch whose \( \tilde{\tau} \) is maximum, is used for further processing. The simulation results for the case of transmitting 16 SPS are shown in Figure 4.3. The parameter \( \epsilon \) is the number of samples deleted at the beginning of transmission. The time at which the curve is maximum, corresponds to the actual timing error. The averaging has been performed over a segment of 64 symbols.

The above search technique is straightforward. The major drawback of the parallel search algorithm is its hardware complexity. With \( N \) SPS, \( N \) parallel stages.
4.2 Maximum-Likelihood (ML) Estimation

Figure 4.2: Parallel search.

Figure 4.3: Simulation results of the parallel search for QPSK with phase error of 40°.
4.2 Maximum-Likelihood (ML) Estimation

are required. To reduce the hardware complexity, the search can be performed in serial. In serial search, a segment of the input signal is passed through the matched filter to obtain \( p(n) \). Following a decision on \( p(n) \), the log-likelihood function is found by summing \( \hat{c}_n^p(n) \) over the the segment. Rather than storing \( \hat{r}_0, \hat{r}_1, \cdots \hat{r}_{N/2} \), and the corresponding data values, the following simple technique can be used to find the \( \tau_{\text{max}} \):

1. Set \( n = 0 \).
2. Find the first \( \tau_0 \).
3. Set \( \tau_{\text{max}} = \tau \).
4. Increment \( n \).
5. Find \( \tau_n \).
6. If \( \tau_n > \tau_{\text{max}} \), set \( \tau_{\text{max}} = \tau_n \).
7. If \( n > N/2 \), output \( \tau_{\text{max}} \) and exit. Otherwise, increment \( n \).
8. Go to step 5.

The advantage of the above approach is that only 1 memory location is required to find \( \tau_{\text{max}} \). Compared to the parallel search, the serial search is \( N \) times less complex to implement at the expense of being \( N \) times slower to determine the timing error.

The algorithm in (4.11) uses the decisions of the data. Replacing the decision values with the matched filter output results in the non data-aided ML feedforward search algorithm

\[
\hat{\tau} = \max \sum_{n=1}^{N} p_n^2(n) + q_n^2(k) \tag{4.12}
\]

The non data-aided and the decision-directed algorithms are independent of phase error \( \theta \). The verification is as follows:

\[
\hat{\tau} = \max \sum_{n=1}^{N} (\Re[p(n)e^{-j\theta}]^2 + (\Im[q(n)e^{-j\theta}]^2)
\]

\[
= \max \sum_{n=1}^{N} |p(n)e^{-j\theta}|^2 \tag{4.13}
\]

Chapter 4 Symbol Timing Synchronisation Algorithms
In taking the magnitude, the phase term $e^{-j\theta}$ is discarded. Therefore,

$$\hat{\tau} = \max_{n=1}^N |p(n)|^2$$

(4.14)

showing that the search technique is independent of phase error. To justify the above analysis, simulations in Figure 4.3 were performed by introducing different values of phase errors. The results obtained were the same as those shown in the above figure. Therefore, the search technique works without a previous knowledge of the phase error.

4.3 Digital Timing Tracking Algorithms

![Diagram](image)

Figure 4.4: The generic timing synchronisation loop.

The generic form of a tracking clock synchroniser for the case of QPSK is shown in Figure 4.4. The received signal plus noise is coherently demodulated, yielding the real and imaginary components of the baseband waveform. The digital clock synchroniser operates on samples of the real and imaginary waveforms, taken at a rate $F_s$ determined by a fixed clock running at a multiple of the symbol rate $R$. These samples are fed to a combined digital matched filter and interpolator. The output sequence is down-sampled to only one or two sample(s) per symbol. From these samples, the detector generates an output sample which is an indication of the instantaneous timing error. Smoothing the instantaneous error is performed by a filter whose output is the trial value of the timing error. The scaled trial error is applied to the combined matched filter and interpolator. The scaling factor $\beta$ determines the acquisition time of the synchronisation loop. The higher this scaling factor is, the faster the acquisition time will be. However, excessively high values of $\beta$ may either drive the loop to instability or cause symbol slipping. The purpose
of using an up-sampler in Figure 4.4 is to ensure that the sampling rates at both input ports of the interpolator are the same.

4.3.1 Tracking Performance

For the considered clock synchronisation algorithms, the timing error variance (without self-noise) is evaluated as [Moe88]

\[ \sigma^2 = \frac{S(fT = 0)}{(K^2_r)} (2B_l T) \]  \hspace{1cm} (4.15)

where

- \( S(fT = 0) \) is the power spectral density (PSD), evaluated at zero frequency.
- \( K_r \) is the gradient of the synchronisation surface.
- \( 2B_l T \) is the normalised two-sided loop noise bandwidth.

To evaluate \( S(fT = 0) \), the error samples are fed to a narrowband ideal low-pass filter. For instance, with a cutoff frequency of 0.05% of the symbol rate, the variance measured is

\[ \sigma^2 = 2 \int_0^{0.05} S(fT) d(fT) \approx 0.1S(0) \]  \hspace{1cm} (4.16)

or \( S(fT = 0) = 10\sigma^2 \). The gradient \( K_r \) of the synchronisation surface must be determined as a function of the signal-to-noise ratio. For this purpose, a second TED is used in parallel whose timing error is offset by \( \epsilon \) with respect to the other detector. The ratio of the mean of the difference in timing detectors output to \( \epsilon \) equals the gradient. The offset must be selected appropriately. For very small offsets, the simulations must be run for a large number of iterations to get a sufficiently precise mean. When the offset is too large, the approximation of the gradient tends to become smaller than the actual value due to the rounding of the detector S-curve.

4.4 Maximum-Likelihood Tracking Algorithm

The algorithm was developed by Kobayashi [Kob71] as part of a joint carrier phase and clock synchronisation, and is expressed by

\[ \frac{\partial \Lambda(\tau)}{\partial \tau} = \sum_{n=1}^{N} \frac{\partial \Lambda p(n, \tau)}{\partial \tau} \]  \hspace{1cm} (4.17)

or
4.4 Maximum-Likelihood Tracking Algorithm

\[
\frac{\partial \Lambda(\tilde{\tau})}{\partial \tilde{\tau}} = \sum_{n} \hat{a}_{n} \frac{\partial}{\partial \tilde{\tau}} \int_{-\infty}^{\infty} r(t)h(t - nT - \tilde{\tau})dt
\]  
(4.18)

\[\frac{\partial p(n, \tilde{\tau})}{\partial \tilde{\tau}}\] is the output of a filter whose impulse response is the derivative of the impulse response of the matched filter, \(h'(t)\). The summation in (4.18) is performed by the loop filter in the feedback loop. The block diagram of the algorithm is shown in Figure 4.5.

\[\begin{array}{c}
\text{Baseband signal in} \\
\text{Interpolator} \\
\end{array} \quad \begin{array}{c}
\downarrow r(t - \tilde{\tau}) \\
N \text{sps} \\
\end{array} \quad \begin{array}{c}
MF \\
\downarrow h(-t) \\
1 \text{sps} \\
\end{array} \quad \begin{array}{c}
p(n) \\
1 \text{sps} \\
\end{array} \quad \begin{array}{c}
\text{Decision} \\
\end{array} \quad \begin{array}{c}
\hat{\nu}(n) \\
\text{Data out} \\
\end{array} \]

Figure 4.5: The Kobayashi clock synchronisation algorithm.

In the implementation of the algorithm, rather than using a separate filter whose impulse response is the derivative of the impulse response of the matched filter, the output of the matched filter is cascaded with a differentiator filter which simply differentiates its input samples.

The realisation in Figure 4.5 is inefficient if a discrete Fourier transform (DFT) operation is used in the design of the filter. With a DFT operation of length \(N\), there will be a delay of \(N\) symbols before the filter starts processing the input samples. To overcome this problem, the loop is re-arranged as shown in Figure 4.6. The matched filter and the differentiator are now in the feedforward path and therefore, the acquisition delay entirely depends on the loop components. For QPSK modulation scheme \(2N_i + 2\) real multiplications and \(4N_i\) real additions are performed per timing error trial \(\tilde{\tau}\), where \(N_i\) is the number of coefficients in the interpolator sub-filter\(^2\).

To analyse the tracking performance, \(r(t)\) in (4.18) is replaced with

\[
r(t) = A_{o} \sum_{i=-\infty}^{\infty} a_{i}h(t - iT - \tau)
\]  
(4.19)

Therefore,

\[
\frac{\partial \Lambda(\tilde{\tau})}{\partial \tilde{\tau}} = \sum_{n=1}^{N} \hat{a}_{n} \int_{-\infty}^{\infty} A_{o} \sum_{i=-\infty}^{\infty} a_{i}h(t - iT - \tau) \frac{\partial h(t - nT - \tilde{\tau})}{\partial \tilde{\tau}} dt
\]  
(4.20)

\(^2\)See Section 4.10 for the implementation of the interpolator.
4.4 Maximum-Likelihood Tracking Algorithm

The integral can be identified as the response of the derivative matched filter to the signal waveform, which is

\[ g'[n - i)T - \tau + \tilde{\tau}] = \frac{\partial g[(n - i)T - \tau + \tilde{\tau}]}{\partial \tilde{\tau}} \quad (4.22) \]

In this algorithm there is self-noise due to the nonzero product of \( \hat{a}_i a_i \) when \( i \neq n \). To find the other source of self-noise, the first derivative of the matched filter output is taken

\[
g'(t) = \cos \left( \frac{\pi t}{T} \right) \cos \left( \frac{\alpha \pi t}{T} \right) \tau^{-1} \left( 1.0 - \frac{4 \alpha^2 \tau^2}{T^2} \right)^{-1}
- \sin \left( \frac{\pi t}{T} \right) T \cos \left( \frac{\alpha \pi t}{T} \right) \pi^{-1} \tau^{-2} \left( 1.0 - \frac{4 \alpha^2 \tau^2}{T^2} \right)^{-1}
- \sin \left( \frac{\pi t}{T} \right) \sin \left( \frac{\alpha \pi t}{T} \right) \alpha^{-1} \left( 1.0 - \frac{4 \alpha^2 \tau^2}{T^2} \right)^{-1}
+ 8 \sin \left( \frac{\pi t}{T} \right) \cos \left( \frac{\alpha \pi t}{T} \right) \alpha^2 \pi^{-1} T^{-1} \left( 1.0 - \frac{4 \alpha^2 \tau^2}{T^2} \right)^{-2} \quad (4.23) \]

At sampling times \( t = mT \), the last three terms in (4.23) are zero and, therefore, \( g'(mT) \) simplifies to

\[ g'(mT) = \frac{\cos(m\pi) \cos(\alpha m\pi)}{mT(1 - 4\alpha^2 m^2)} \quad (4.24) \]

The variations of \( g'(t) \) as a function of time have been shown in Figure 4.7. It can be clearly seen that at sampling times \( mT \), the derivative is not zero. The Nyquist raised cosine pulse shape is zero at all sampling times except time \( t = 0 \)

Figure 4.6: Improved realisation of the Kobayashi clock synchroniser.

\[ \frac{\partial \Delta(\tilde{\tau})}{\partial \tilde{\tau}} = A_0 \hat{a}_n \sum_{i=-\infty}^{\infty} a_i \int h(t - iT - \tau) \frac{\partial h(t - nT - \tilde{\tau})}{\partial \tilde{\tau}} \, dt \quad (4.21) \]
Figure 4.7: Variations in $g'(t)$ with time.

(See Figure 4.1). Therefore, the integration in (4.21) is nonzero even in the absence of AWGN. The timing performance is shown in Figure 4.8. The SNR is assumed to be infinite and the channel introduces normalised timing errors $\tau$ of $0, 0.25T$ and $0.5T$. The simulation results justify the analysis that the algorithm has self-noise. The acquisition time of this synchroniser is slow. If the SNR is not infinite, the acquisition time will be even longer.

The noise performance is shown in Figure 4.9. The performance degradation at low $E_s/N_0$ is caused by a reduction of the gradient of the TED characteristics. For moderate to high SNR ($E_s/N_0$), the effect of decision errors can be ignored and the curve approaches the Cramer-Rao bound [Jes91]

$$\frac{S(fT=0)}{K^2_t} = \frac{N_0}{2E_s(-g''(0)T^2)}$$  \hspace{1cm} (4.25)

where $g''(0)$ denotes the second time derivative of the baseband pulse $g(t)$ at the matched filter output at time $t = 0$, and is given by

$$-g''(t)T^2 = -T \sin \left( \frac{\pi f}{T} \right) \pi \cos \left( \frac{\alpha \pi t}{T} \right) t^{-1} \left( 1.0 - \frac{4 \alpha^2 T^2}{T^2} \right)^{-1}$$
4.4 Maximum-Likelihood Tracking Algorithm

\[ -2 T^2 \cos \left( \frac{\pi t}{T} \right) \cos \left( \frac{\alpha \pi t}{T} \right) t^{-2} \left( 1.0 - \frac{4 \alpha^2 t^2}{T^2} \right)^{-1} \]

\[ -2 T \cos \left( \frac{\pi t}{T} \right) \sin \left( \frac{\alpha \pi t}{T} \right) \alpha \pi t^{-1} \left( 1.0 - \frac{4 \alpha^2 t^2}{T^2} \right)^{-1} \]

\[ +16 \cos \left( \frac{\pi t}{T} \right) \cos \left( \frac{\alpha \pi t}{T} \right) \alpha^2 \left( 1.0 - \frac{4 \alpha^2 t^2}{T^2} \right)^{-2} \]

\[ +2 T^3 \sin \left( \frac{\pi t}{T} \right) \cos \left( \frac{\alpha \pi t}{T} \right) \pi^{-1} t^{-3} \left( 1.0 - \frac{4 \alpha^2 t^2}{T^2} \right)^{-1} \]

\[ +2 T^2 \sin \left( \frac{\pi t}{T} \right) \sin \left( \frac{\alpha \pi t}{T} \right) \alpha^{-2} \left( 1.0 - \frac{4 \alpha^2 t^2}{T^2} \right)^{-1} \]

\[ -8 T \sin \left( \frac{\pi t}{T} \right) \cos \left( \frac{\alpha \pi t}{T} \right) \alpha^2 \pi^{-1} t^{-1} \left( 1.0 - \frac{4 \alpha^2 t^2}{T^2} \right)^{-2} \]

\[ -T \sin \left( \frac{\pi t}{T} \right) \cos \left( \frac{\alpha \pi t}{T} \right) \alpha^2 \pi t^{-1} \left( 1.0 - \frac{4 \alpha^2 t^2}{T^2} \right)^{-1} \]

\[ -16 \sin \left( \frac{\pi t}{T} \right) \sin \left( \frac{\alpha \pi t}{T} \right) \alpha^3 \left( 1.0 - \frac{4 \alpha^2 t^2}{T^2} \right)^{-2} \]

\[ +128 \sin \left( \frac{\pi t}{T} \right) \cos \left( \frac{\alpha \pi t}{T} \right) \alpha^4 t^{-1} \pi^{-1} \left( 1.0 - \frac{4 \alpha^2 t^2}{T^2} \right)^{-3} \] (4.26)
The above expression at time $t = 0$ simplifies to
\[ -g'(0)T^2 = \frac{\pi^2 + 3\pi^2\alpha^2}{3} - 8\alpha^2 \] (4.27)

By substituting the result of (4.27) in (4.25), the CRB has been plotted as a function of the SNR in Figure 4.9.

![Figure 4.9: Normalised PSD at $f_T = 0$ for DD ML tracking algorithm.](image)

**4.5 Two-Point Difference Tracker**

The algorithm is given by [Gar88a]
\[
\begin{align*}
u(n) &= \Re\{(p(n + 1/2) - p(n - 1/2))\hat{\epsilon}_T^m\} \\
&= [p_r(n + 1/2) - p_r(n - 1/2)]\hat{a}_n + [q_i(n + 1/2) - q_i(n - 1/2)]\hat{b}_n
\end{align*}
\] (4.28)

The implementation of this algorithm requires two samples per symbol. One sample occurs at the symbol strobe time, and the other one occurs midway between the symbol strobe times. To analyse the performance of the tracker, the output of the MF is considered
\[
p(n) = \int_{-\infty}^{\infty} r(t)h(t - nT - \tilde{\tau})dt
\] (4.29)
where
\[ r(t) = A_0 \sum_{i=-\infty}^{\infty} a_i h(t - iT - \tau) \quad (4.30) \]

Substituting (4.30) in (4.29) results in
\[ p(n) = A_0 \sum_{i=-\infty}^{\infty} a_i \int_{-\infty}^{\infty} h(t - iT - \tau) + h(t - nT - \tilde{\tau}) dt \quad (4.31) \]

Denoting the overall pulse shape by
\[ g[(n - i)T - \tau] = \int_{-\infty}^{\infty} h(t - iT - \tau)h(t - nT - \tilde{\tau}) dt \quad (4.32) \]

where \( \tau \) is the timing error, Equation (4.31) simplifies to
\[ p(n) = A_0 \sum_{i=-\infty}^{\infty} a_i g[(n - i)T - \tau] \quad (4.33) \]

Assuming that the BPSK modulation scheme has been used, the second term in (4.28) is zero. Therefore
\[ u(n) = [p_r(n + 1 / 2) - p_r(n - 1 / 2)] \hat{a}_n \quad (4.34) \]

Substituting (4.33) in (4.34) results in
\[ u(n) = A_0 \hat{a}_n \sum_{i=-\infty}^{\infty} g([n + 1 / 2 - i]T - \tau) - g([n - 1 / 2 - i]T - \tau) \quad (4.35) \]

or
\[ u(n) = A_0 \hat{a}_n g(T/2 - \tau) - g(-T/2 - \tau) \]
\[ + A_0 \hat{a}_n \sum_{i \neq n} g([n + 1 / 2 - i]T - \tau) - g([n - 1 / 2 - i]T - \tau) \quad (4.36) \]

The first line contributes to the detection of timing error, while the second line is self-noise contributed by adjacent pulses. Simulation results are shown in Figure 4.10. It is clearly seen that the output of the detector is zero only when there is no change of sign in the mid-point samples. This is desirable. However, when there is a change of sign, the output is \( 2 / \sqrt{2} \), which is a major source of large tracking noise.
4.6 Gardner Algorithm

The algorithm is based on delay differencing between the current sample and another sample delayed by half the symbol period, i.e.

\[ x_d(t) = P_r(t) - P_r(t - T/2) \quad (4.37) \]

By passing \( x_d(t) \) through a square-law rectifier, the following is obtained

\[ u(n) = x_d^2(t) = p_r^2(t) + p_r^2(t - T/2) - 2p_r(t)p_r(t - T/2) \quad (4.38) \]

It is due to the above squaring that the operation of the Gardner algorithm becomes independent of the carrier phase. Substituting the early sampling time, \( t_e = nT + \tau \), and the late sampling time, \( t_l = nT + \tau + T/2 \) in (4.38) gives

\[ u(n) = p_r^2(\tau + nT) - p_r^2(\tau + [n - 1]T) \]
\[ - 2p_r(\tau + [n - 1/2]T) \{p_r(\tau + nT) - p_r(\tau + [n - 1]T)\} \quad (4.39) \]

By simulation, it has been shown that the first two terms have a major contribu-
tion to the self-noise (see Figure 4.11\(^3\)). Therefore, the algorithm is approximated by dropping the \(p^2\) terms

\[
u(n) = -p_r(\tau + [n - 1/2]T) \{p_r(\tau + nT) - p_r(\tau + [n - 1]T)\}
\]  \(4.40\)

At equilibrium, \(\tau\) is zero. Therefore (4.40) becomes

\[
u(n) = p_r(n - 1/2)\{p_r(n) - p_r(n - 1)\}
\]  \(4.41\)

The sign reversal in (4.41) is compensated by changing the sign of the loop gain factor \(\beta\).

![Figure 4.11: Performance of the Gardner algorithm.](image)

Equation (4.40) is for BPSK. For (O)QPSK modulation scheme, the Gardner algorithm becomes

\[
u(n) = p_r(n + 1/2)\{p_r(n) - p_r(n + 1)\} + q_i(n + 1/2)\{q_i(n) - q_i(n + 1)\}
\]

\[
= \Re\{p(n + 1/2)[p^*(n) - p^*(n + 1)]\}
\]  \(4.42\)

The block diagram of the Gardner algorithm is shown in Figure 4.12. At the output of the matched filter, the sampling rate is reduced by a factor of \(N/2\), where

\(3\) The polarity of the performance without approximations has been reversed for clarity.

*Chapter 4 Symbol Timing Synchronisation Algorithms*
$N$ is the number of samples per transmitted symbol. The remaining 2 samples per symbol are used in (4.42) to generate an error sample. The demultiplexers separate the samples which occur at the symbol strobe times from the samples which occur half-way between the symbol strobe times. For QPSK modulation scheme, $2N_t + 3$ real multiplications and $2N_t + 2$ real additions are performed per timing error estimate $\hat{\tau}$.

The S-curve is shown in Figure 4.13(a). It can be clearly seen that as the timing error increases, the gradient of the timing error detector decreases and, hence, the acquisition time increases. The longest acquisition time occurs when the normalised timing error is $\pm 0.25$. On the same figure, it has been shown that the operation of the algorithm is independent of the phase error $\theta$. An interesting feature of the Gardner algorithm is that its gradient remains unaffected by additive white Gaussian noise. The gradient of the timing error detector remains at 1.48 (See Figure 4.13(b)). The gradient of other synchronisation algorithms is a function of the SNR; as SNR decreases, the gradient decreases. A low gradient means a slow acquisition time.

The noise performance of the Gardner algorithm is shown in Figure 4.13(c). For increasing $E_s / N_o$, the simulation results converge to Cramer-Rao bound [Jes91]

\[
\frac{S(fT = 0)}{k^2} = \frac{N_o}{E_s \alpha \pi^2}
\] (4.43)
4.6 Gardner Algorithm

Normalised timing error, $\tau/T$

(a) S-curve

(b) Change in gradient of the error detector as function of SNR

(c) Normalised PSD at $fT = 0$

Figure 4.13: Performance of the Gardner algorithm.
4.7 Decision-Directed Minimum Mean-Square Error Algorithm

From Figure 4.13(c) it can be deduced that, even at medium to high SNR, there is self-noise in this algorithm. When $\alpha$ is less than 100%, the zero crossings of data transitions do not lie midway between the desired symbol strobe points. The average location is centred on the midway point, but any individual point can depart from the average, causing self-noise. The self-noise and the acquisition time are related to the loop gain factor $\beta$. By simulations it has been found that the root mean square (RMS) of the self-noise changes as

\[ \sigma_r = 0.1\beta \]  

(4.44)

Low values of $\beta$ result in smaller jitter at the expense of longer acquisition time. Even replacing the symbol strobe values in (4.42) with the hard decision on the symbol, as suggested in [Gar85b], does not eliminate the effect of noise in the timing loop. It is only with the rolloff factor of 100% that the self-noise in the Gardner algorithm vanishes. Further simulations have shown that the gradient of the Gardner algorithm characteristics, $K_r$, is a function of the rolloff factor $\alpha$ by the following relationship

\[ K_r(\alpha) \approx -2.75\alpha - 0.06 \]  

(4.45)

Therefore the algorithm is not suitable for very small rolloff factors.

4.7 Decision-Directed Minimum Mean-Square Error Algorithm

The minimum mean-square error (MMSE) symbol synchroniser uses a feedback algorithm to minimise the mean square error between the input and the output of the receiver's decision. The algorithm was investigated in [Men77]. The analysis is as follows:

The error $\epsilon$ between the complex conjugate of the matched filter output

\[ p(n) = p_r(n) + jq_i(n) \]  

(4.46)

and the complex conjugate of the decision outputs

\[ \hat{c}_n = \hat{a}_n + j\hat{b}_n \]  

(4.47)

is given by [Gar88a]

\[ \epsilon = p^*(n) - \hat{c}_n^* \]

\[ = (p_r - \hat{a}_n) - j(q_i - \hat{b}_n) \]  

(4.48)
Squaring the error results in

\[ e^2 = p_r^2(n) + \hat{a}_n^2 - 2p_r\hat{a}_n + q_i^2(n) + b_n^2 - 2q_i\hat{b}_n \]

\[-j\{2(p_r(n) - \hat{a}_n)(q_i(n) - \hat{b}_n)\}\] (4.49)

To minimise the mean of \( e^2 \), \( E(e^2) \), it is differentiated with respect to \( \hat{\tau} \)

\[ u(n) = \Re \frac{\partial E(e^2)}{\partial \hat{\tau}} \]

\[ = \Re \{2p'_r[p_r(n) - \hat{a}_n] + 2q'_i(n)[q_i(n) - \hat{b}_n] \]

\[-j[2q'_i(n)[p_r(n) - \hat{a}_n] + 2p'_r(n)[q_i(n) - \hat{b}_n]]\} \]

\[ = [p_r(n) - \hat{a}_n)p'_r(n) + [q_i(n) - \hat{b}_n]q'_i(n) \]

\[ = \Re \{p'(n)[p^*(n) - \hat{c}_r^*]\} \] (4.50)

where \( p'(n) \) is the derivative matched filter output sampled at times \( kT + \hat{\tau} \). Assuming the receivers decisions are correct, \( p^*(n) \) and \( \hat{c}_r^* \) in (4.50) cancel each other out and, therefore, the algorithm is self-noise free. The block diagram is shown in Figures 4.14. The decisions are subtracted from the sampled matched filter output before multiplication with the sampled derivative matched filter output. For QPSK modulation scheme, \( 2N_t + 6 \) real multiplications and \( 2N_t + 5 \) real additions are performed per timing error estimate \( \hat{\tau} \).

![Figure 4.14: Decision-directed MMSE algorithm.](image)

The implementation of the MMSE algorithm requires an automatic gain control (AGC), which controls the signal level at the input of the decision device [Jes91]. The need for an AGC can be shown from the following analysis:
Neglecting noise, and using Equations (4.19)–(4.22), the MMSE detector yields

\[ u(n) = A_o \sum_{i=-\infty}^{\infty} a_i g \{ (n - i)T - \Delta \tau \} \left[ A_o \sum_{k=-\infty}^{\infty} a_k g \{ (n - k)T - \Delta \tau \} - \hat{a}_n \right] \quad (4.51) \]

where \( \Delta \tau = \tau - \hat{\tau} \). Assuming \( g(t) \) has Nyquist pulse shape, the samples \( g(mT) = 0 \) for all \( m \neq 0 \) and the bracketed factor will vanish for \( \Delta \tau = 0 \) and \( A_o = 1 \), suppressing self-noise under these conditions. Therefore, the MMSE requires adjustment of amplitude \( A_o \).

**4.8 Data Transition and Tracking Loop (DTTL)**

This algorithm is given by [Lin72]

\[ u(n) = \Re \left[ \hat{c}_{n-1} - \hat{c}_n \right] p^*(n - 1/2) \]
\[ = [\hat{a}_{n-1} - \hat{a}_n]p_r(n - 1/2) + [\hat{b}_{n-1} - \hat{b}_n]p_i(n - 1/2) \quad (4.52) \]

The block diagram of the synchroniser is shown in Figure 4.15. Two samples per symbol are required; one sample occurs at the symbol strobe time, and the other occurs half way between the strobe times. For QPSK modulation scheme \( 2N_i + 5 \) real multiplications and \( 2N_i + 1 \) real additions are required per timing error estimate \( \hat{\tau} \). The timing error is zero when there is no transition in the decided data at adjacent sampling points. The noise performance is shown in Figure 4.16. For moderate and high \( E_s / N_o \), the effect of decision errors can be ignored and the simulation results converge to

\[ \frac{S(fT = 0)}{k_T^2} = \frac{N_o}{2E_s} \frac{1}{2g(-T/2)T^2} \quad (4.53) \]
where by substituting $t = -T/2$ in (4.7), the following is obtained

$$g'(-T/2)T = \frac{4 \cos\left(\frac{\alpha \pi}{2}\right)}{\pi (1 - \alpha^2)} + \frac{2 \sin\left(\frac{\alpha \pi}{2}\right)\alpha}{1 - \alpha^2} - \frac{8 \cos\left(\frac{\alpha \pi}{2}\right)\alpha^2}{\pi (1 - \alpha^2)^2} \quad (4.54)$$

![Figure 4.16: Normalised PSD at $fT = 0$ for DTTL TED.](image)

The results in Figure 4.16 show that when close to optimum noise performance is required, the DTTL algorithm is suitable.

The acquisition performance of the algorithm is shown in Figure 4.17 with closed-loop gain factors $\beta = 0.1$ and $\beta = 1$. With the former the acquisition time is slow, while with the latter the acquisition time is faster. A slow acquisition time is not desirable for fast application, such as the Iridium satellites. It is also seen from the same figure that by increasing $\beta$, the tracking jitter has also increased. This tracking jitter, in turn, may cause decision errors and hence a poor BER performance. From several design examples it has been verified that the operation of the DTTL is influenced by the channel phase error. The algorithm had to be used in a joint\textsuperscript{4} configuration with a phase synchroniser. With the joint operation, the acquisition time further increased. Since in this thesis, the emphasis is on fast applications, the DTTL is not be considered further.

\textsuperscript{4}For a discussion on joint synchronisation see Chapter 6.
4.9 Self-Noise Cancellation of the mM&M Algorithm

The mM&M algorithm is given by [Moe88]

\[
\begin{align*}
\hat{u}(n) &= [\hat{a}_{n-1} - \hat{a}_{n+1}]p_r(n) + [\hat{b}_{n-1} - \hat{b}_{n+1}]q_t(n) \\
&= \Re\{\hat{p}^*(n)[\hat{c}_{n-1} - \hat{c}_{n+1}]\} \tag{4.55}
\end{align*}
\]

To analyse the performance, it is assumed that the channel SNR is infinite and the received signal is

\[
r(t) = A_o \sum_{i=-\infty}^{\infty} a_i h(t - iT - \tau) \tag{4.56}
\]

where \(a_i\) is the data values, \(h(t)\) is the shape of the transmit pulse, \(\tau\) is the timing shift of the incoming signal, \(1/T\) is the symbol rate, \(A_o\) is the amplitude of the received signal. By passing \(r(t)\) through a MF, the following output samples are obtained

\[
p(n) = \int_{-\infty}^{\infty} r(t) h(t - nT - \bar{\tau}) dt \tag{4.57}
\]

where \(\bar{\tau}\) is the trial value of the timing error. For the time being, \(p(n)\) is assumed to be real. In the absence of noise, the matched filter output is

\[
p(n) = A_o \sum_{i=-\infty}^{\infty} a_i \int_{-\infty}^{\infty} h(t - iT - \tau) h(t - nT - \bar{\tau}) dt \tag{4.58}
\]
4.9 Self-Noise Cancellation of the mM&M Algorithm

The integral in (4.58) can be shown to be equal to \( g([n - i]T - \tau) \), where \( g(t) \) is the time continuous response of the receiver filter to the input pulse shape \( h(t) \), and \( \tau \) is the timing error. Therefore,

\[
p(n) = A_o \sum_{i=-\infty}^{\infty} a_i g([n - i]T - \tau)
\]

The mM&M algorithm for the BPSK modulation scheme is [Moe88]

\[
u(n) = [\hat{a}_{n-1} - \hat{a}_{n+1}]p(n)
\]

where \( \hat{a}_n \) is the receiver’s decision about the data symbol \( a_n \). By substituting (4.59) in (4.60), the mM&M algorithm can be expanded as

\[
u(n) = [\hat{a}_{n-1} - \hat{a}_{n+1}]a_{n+1}g(-T - \tau) \\
+ [\hat{a}_{n-1} - \hat{a}_{n+1}]a_{n-1}g(T - \tau) \\
+ [\hat{a}_{n-1} - \hat{a}_{n+1}]a_n g(-\tau) \\
+ [\hat{a}_{n-1} - \hat{a}_{n+1}] \sum_{i\neq 0} a_i g([n - i]T - \tau)
\]

The terms on the first two lines contribute to the DC output of the synchroniser. Assuming the overall pulse shape is raised cosine, \( g(MT) = 0, m \neq 0 \). Therefore, the terms on the last line vanish. The terms on the third line cause self-noise. The self-noise will not be zero even when the timing error \( \tau \) is zero. The tracking performance under the following conditions has been shown in Figure 4.18:

- infinite SNR;
- two samples per transmitted symbol, \( T_s = 2T \);
- timing error of half a symbol, \( \tau = 0.5T \);
- zero frequency and phase errors;
- loop gain factor \( \beta = 0.1 \) and \( \beta = 0.18 \).

Although the loop has acquired the error, there is a large self-noise with \( \beta = 0.1 \). By increasing \( \beta \) to 0.18, the loop was forced to acquire the error even faster. Initially, the synchroniser detected and tracked the timing error. But, at time \( t = 2540T \), the loop slipped one symbol. The symbol slip occurred again at \( t = 6810T \). Although, the loop returns to the correct tracking point after each symbol slip, this has a
4.9 Self-Noise Cancellation of the mM&M Algorithm

Figure 4.18: The mM&M clock synchroniser.

disabling effect on the performance of the modem. Symbol slips give rise to many errors.

The performance of the detector algorithm in the presence of AWGN is now considered. The tracking error variance is shown in Figure 4.19. For increasing
4.9 Self-Noise Cancellation of the mM&M Algorithm

$E_s / N_0$, the simulation results converge to [Jes91]

$$
S(fT = 0) = \frac{N_0}{K_t} \frac{1}{E_s} \left. 4g(t)T^2 \right|_{t=-T}
$$

(4.62)

where by substituting $t = -T$ in (4.7) the following is obtained

$$
g'(-T)T = \frac{\cos(\alpha T)}{1 - 4\alpha^2}
$$

$$
= \frac{\pi}{4} |_{\alpha = 0.5}
$$

(4.63)

![Figure 4.19: Normalised PSD at fT = 0 for the mM&M TED.](image)

The existence of self-noise with the mM&M algorithm can be deduced from the results in Figure 4.19.

With the above background on the mM&M algorithm, a novel technique is described to cancel the self-noise. The details are as follows [Dan95]:

To cancel the self-noise, the following is added to the self noise

$$
a_n a_{n+1} - \hat{a}_n \hat{a}_{n-1}
$$

(4.64)

If $a_n \hat{a}_{n+1} - a_n \hat{a}_{n-1}$ is used instead of (4.64), the detector output will be zero no matter what the timing error is. Assuming the receiver's decisions are correct, the
self-noise entirely vanishes. From (4.59), it can be deduced that adding (4.64) to (4.61) means adding the following to the original algorithm

\[ \hat{a}_n[p(n + 1) - p(n - 1)] \]  

Therefore, the optimised mM&M algorithm emerges as

\[ u(n) = [\hat{a}_{n-1} - \hat{a}_{n+1}]p(n) + \hat{a}_n[p(n + 1) - p(n - 1)] \]

\[ = u_1(n) + u_2(n) \]

Examining the terms in (4.66) in more detail reveals that \( u_1(n) \) and \( u_2(n) \) each generates error samples which are symmetrical about the timing error \( \tau \). That is

\[ \tau = [u_1(n) + u_2(n)]0.5 \]

The multiplication by 0.5 can be included in the loop gain \( \beta \) to avoid an extra multiplier. To generalise (4.66) to include QPSK and OQPSK modulation schemes, the mM&M algorithm is

\[ u_\tau(n) = \Re\{[\hat{c}_{n-1} - \hat{c}_{n+1}]p^*(n)\} \]

By going through Equations (4.60) - (4.66), the generalisation of the optimised algorithm becomes

\[ u(n) = \Re\{[\hat{c}_{n+1} - \hat{c}_{n-1}]p^*(n) + \hat{c}_n^*[p(n + 1) - p(n - 1)]\} \]

The block diagram of the optimised mM&M algorithm is shown in Figure 4.20. The output of the combined matched filter and interpolator is sampled at symbol rate, \( R \). Therefore, the operation of the optimised algorithm requires one sample per symbol. This is an advantage over algorithms, such as the Gardner algorithm, which require 2 samples per symbol to operate; the proposed algorithm works at twice the speed of the Gardner algorithm.

The algorithmic complexity of the optimised algorithm is reasonably low. For QPSK modulation scheme, \( 2N_i + 11 \) real multiplications and \( 2N_i + 9 \) real additions are performed per timing error estimate \( \hat{\tau} \). Furthermore, no differentiators or de-multiplexer have been used. This makes the proposed algorithm more attractive for ASIC implementation.

The simulation results are shown in Figure 4.21. The effect of Gaussian noise on the performance of the detector is shown in Figure 4.21(a). The noise performance...
approaches the Cramer-Rao bound (CRB) as the SNR increases. At $E_s / N_0$ greater than about 10 dB, the performance is optimum. To appreciate the significance of this breakthrough, the simulation result of the original algorithm in Figure 4.19 is considered. The performance never becomes optimum. The difference between the simulation results and the CRB is the indication of the self-noise of the original algorithm. This difference in the optimised algorithm is zero. Therefore the result shown in Figure 4.21(a) is sufficient to justify that at medium to high SNR, the proposed optimised symbol synchroniser is self-noise free.

To compare the tracking performance of the optimised and the original algorithms, the results in Figures 4.21(b) and 4.19 are considered. The simulation in Figure 4.21(b) prove the analysis that the self-noise has been completely cancelled. There are no symbol slips over the observation period shown. To emphasise the fast acquisition of the optimised algorithm, the result in Figure 4.21(b), magnified in the region $t = 0$ to $t = 100T$, has been shown in Figure 4.21(c). It is seen that the acquisition of the worst case symbol error occurs in $10T - 15T$. The interesting thing is that the loop gain factor $\beta$ was set to 1 which is about 5.6 times the gain at which the original mM&M algorithm showed symbol slips. Further simulations were performed with BPSK, OQPSK and $\pi / 4$-QPSK modulation schemes. The longest acquisition time occurred in the case of the OQPSK which was $15T - 20T$ (not shown). Fast acquisition, fast processing, low algorithmic complexity, zero self-noise and optimum noise performance are the promising features of the optimised mM&M algorithm for application in systems such as the Iridium or Inmarsat-P satellites.

Figure 4.20: The block diagram of the optimised mM&M clock synchroniser.
4.9 Self-Noise Cancellation of the mM&M Algorithm

(a) Noise performance

(b) Acquisition performance

(c) Acquisition performance over the first 100 symbols

Figure 4.21: Performance of the optimised mM&M clock synchroniser.

Chapter 4 Symbol Timing Synchronisation Algorithms
4.10 Timing Error Correction

One of the features of an all-digital implementation of a modem is that the sampling is controlled by a fixed clock whose sample timing is not locked to the symbol timing nor is the sample rate exactly equal to the symbol rate, or a rational multiple thereof [Gar88b]. Timing adjustment must be performed to sample the signal at the optimum time. It is well known that one can achieve fractional delays by using multirate techniques [Lim88]. The specific multirate function required for timing error correction is the interpolation of the incoming samples, which generates $L$ samples from every input sample. By interpolation the underlying continuous waveform is reconstructed, which is then re-sampled at the optimum point.

In the following section the polyphase interpolator for timing error correction is discussed.

4.10.1 Polyphase Filters

The interpolation filters are obtained by decomposing an FIR prototype filter into $L$ sub-filters, $S_0, S_1, \ldots, S_{L-1}$. If the prototype filter has a length $Q$, then the individual filters will each have length $K = Q/L$. Although by interpolation the sampling rate increases by a factor of $L$, only one of $L$ samples is selected. Therefore, the sampling rate is not changed in overall (see Figure 4.22).

![Figure 4.22: Timing error correction by polyphase filters.](image)

The process of obtaining the interpolated samples $y_n$ by convolving the input samples $x_n$ with the decomposed filters $S_l$ can be expressed by the following simple relationship [Ver93]
4.10 Timing Error Correction

\[
\begin{bmatrix}
y_n \\
y_{n-1} \\
\vdots \\
y_{n-L}
\end{bmatrix} =
\begin{bmatrix}
h_0 & h_L & \cdots & h_{(K-1)L} \\
h_1 & h_{L+1} & \cdots & \vdots \\
\vdots & \vdots & \ddots & \vdots \\
h_{L-1} & h_{2L-1} & \cdots & h_{KL-1}
\end{bmatrix}
\begin{bmatrix}
x_n \\
x_{n-1} \\
\vdots \\
x_{n-K}
\end{bmatrix}
\]

\[
= \begin{bmatrix}
S_0 \\
S_1 \\
\vdots \\
S_{L-1}
\end{bmatrix}
\begin{bmatrix}
x_n \\
x_{n-1} \\
\vdots \\
x_{n-K}
\end{bmatrix}
\]

\( (4.70) \)

The number of filters \( L \) specifies the resolution by which the timing error is found. Higher number of sub-filters results in a finer resolution, while a lower number of sub-filters results in a coarser resolution. The choice of the delay resolution is a compromise between the accuracy in results and the restriction on the memory size to store the filter coefficients.

It is interesting to consider the polyphase lattice structure (PLS) [Yim91] and make a comparison with the above interpolator. The operation of the PLS is expressed by

\[
r_{p,d}(l) = \sum_{q=0}^{M-1} \sum_{i=\alpha_{p,q,d}}^{\beta_{p,q,d}} s_q(l-i)h_{p,q,d}(i) \quad 0 \leq p \leq L - 1
\]

where \( s_q \) is the \( q \)-th shift register, and the lattice of sub-filters \( h_{p,q,d}(n) \) is a decomposition of the prototype filter \( h(n) \) which is given by

\[
h_{p,q,d}(i) = h(iMLD + pMD + qLD + dM)
\]

The parameter \( d \) determines the amount of delay and can be specified to have an arbitrary range, e.g. the interval \( \left[-\frac{p}{2}, \frac{p}{2} - 1\right] \).

The summation limits are given by:

\[
\alpha_{p,q,d} = \left\lfloor -\frac{p}{L} - \frac{q}{M} - \frac{d}{DL} \right\rfloor
\]

and

\[
\beta_{p,q,d} = \left\lceil \frac{N-1}{ML} - \frac{p}{L} - \frac{q}{M} - \frac{d}{DL} \right\rceil
\]

where \( \lfloor u \rfloor \) denotes the largest integer less than or equal to \( u \) and \( \lceil u \rceil \) denotes the smallest integer greater than or equal to \( u \).
4.10 Timing Error Correction

The length of each sub-filter in the PLS is $\alpha + \beta + 1$. To calculate $\alpha$ and $\beta$, $M \times L \times D$ iterations are required. There are 3 real additions and 4 real multiplications in every iteration of determining $\alpha_{p,q,d}$. Similarly, there are 4 real additions and 6 real multiplications in every iteration of calculating $\beta_{p,q,d}$.

The length of each sub-filter in (4.70) is $N/L$, where $N$ is the number of coefficients in the prototype filter. Furthermore, the PLS performs decimation of the signal by $M$. The impact of performing decimation within the PLS is that the hardware complexity of the structure is at least $M$ times higher than that of (4.70). Furthermore, in the PLS, one input and $M$ output commutators are required. Finally, to decompose the prototype filter coefficients with the PLS, 8 real multiplications and 3 real additions are required. However, the decomposition in (4.70) is according to

$$h_p(i) = h(p + iL)$$

(4.75)

which requires only 1 real multiplication and 1 real addition.

4.10.1.1 Comments

At the beginning of this section, it was mentioned that the specific multirate operation required for timing error correction is the interpolation of the incoming samples. However in PLS, the signal is not only interpolated but also decimated. Although the sampling rate at the output of the interpolator does not change\footnote{For every one input samples, one sample is output.}, the implementation complexity will increase. This added complexity will not be in terms of a longer prototype filter length; a serial to parallel converter at the input, $M$ parallel-to-serial converters at the output and a summing junction are required. This can be avoided by setting the decimation factor $M$ to 1. Furthermore, setting $D$ and $d$ to 1 and 0, respectively, will simplify Yim's work to the practical interpolators which have been available in the published literature such as [Cro83] and [Ver93].

---

*Chapter 4 Symbol Timing Synchronisation Algorithms*
4.11 Summary

In this chapter a survey of different timing error synchronisers was presented. The following recommendations can be made with regards to choosing a suitable timing error detector:

- Timing error detection by performing a search operation is physically realizable. However, the speed of the operation makes this technique unsuitable in applications in which a rapid acquisition of error is important.

- The two-point tracker has excessive self-noise. This algorithm has a poor noise performance.

- The Kobayashi algorithm has a slow acquisition time and a poor tracking performance.

- When the speed of the circuit is a serious limitation (as can be the case of very high data rates) the mM&M and the optimised mM&M algorithms are suitable, since only one sample per symbol is required.

- When close to optimum tracking performance is required, the DTTL, the MMSE, and the optimised mM&M algorithm are suitable, since their tracking performance is close to the CRB.

- The Gardner algorithm is the only algorithm that can work without a prior knowledge of carrier phase error. Therefore phase synchronisation is performed after timing recovery. Other timing recovery algorithms must be implemented in a joint configuration with a phase synchroniser. The penalty for the joint configuration is a slightly longer acquisition time.

- When fast acquisition of timing error is required, the mM&M, the optimised mM&M are suitable. For QPSK modulation scheme, which will be used in the Iridium satellites, the acquisition time is about 15 symbols at high SNR in the absence of phase error. The acquisition time of MMSE is fast\(^6\), but the analysis shows that an automatic gain control (AGC) is required to adjust the amplitude of the received signal. Without AGC, the MMSE algorithm will not have an optimum noise performance.

\(^6\)Simulation result is not shown in this chapter.
In Table 4.1, the computation requirement of each algorithm for every timing error trial \( t \) has been tabulated. The parameter \( N_i \) is the size of the interpolator sub-filters. The modulation scheme is assumed to be QPSK. Furthermore, the tabulated results do not include the operations required to make a decision on the transmitted data.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>real multiplications</th>
<th>real additions</th>
<th>SPS</th>
<th>Self-noise free</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimised mM&amp;M</td>
<td>( 2N_i + 11 )</td>
<td>( 2N_i + 9 )</td>
<td>1</td>
<td>Yes</td>
</tr>
<tr>
<td>DTTL</td>
<td>( 2N_i + 5 )</td>
<td>( 2N_i + 1 )</td>
<td>2</td>
<td>Yes</td>
</tr>
<tr>
<td>MMSE</td>
<td>( 2N_i + 6 )</td>
<td>( 2N_i + 5 )</td>
<td>1</td>
<td>Yes</td>
</tr>
<tr>
<td>Gardner</td>
<td>( 2N_i + 3 )</td>
<td>( 2N_i + 2 )</td>
<td>2</td>
<td>No</td>
</tr>
<tr>
<td>Kobayashi</td>
<td>( 4N_i + 3 )</td>
<td>( 4N_i )</td>
<td>1</td>
<td>No</td>
</tr>
<tr>
<td>mM&amp;M</td>
<td>( 2N_i + 6 )</td>
<td>( 2N_i + 3 )</td>
<td>1</td>
<td>No</td>
</tr>
</tbody>
</table>

From the above recommendations and the results in Table 4.1, the following algorithms have been selected for further investigation in Chapter 6:

* The optimised mM&M algorithm for its low algorithmic complexity, fast speed, and optimum performance.

* The Gardner algorithm for its operation independent of phase error.

Two polyphase interpolators for timing error correction were considered. It was shown that the Verdin and Tozer interpolator has less hardware complexity than the PLS and is, therefore, selected for simulations in Chapter 6.

The results in this chapter were based on using a single-channel modem under the assumption of perfect carrier error synchronisation. In Chapter 6, the selected algorithms will be simulated in an 8-channel MCDD whose received FDM signal is subject to carrier frequency and phase errors.
Chapter 5
Phase Synchronisation Algorithms

In this chapter, different phase error detection (PED) algorithms will be surveyed. The details of the proposed optimisation of the decision-directed maximum-likelihood (ML) feedforward PED will be presented. Compared to the surveyed algorithms, the optimised algorithm has a superior noise performance at medium to high signal-to-noise ratio (SNR).

For coherent demodulation, the carrier phase error must be detected and corrected in the receiver by extracting the required phase information from the received signal. The most straightforward technique in extracting an estimate of the phase error is by performing a search in the expected uncertainty range. From every search, an estimate of the error is found. The estimate which maximises the log-likelihood function\(^1\) is of interest. The major problem with this approach is the hardware complexity that makes it unrealisable for practical applications.

Another technique in maximising the log-likelihood function is by taking its derivative with respect to the trial value of the phase error, and setting it to zero. In this chapter a survey of the following phase error detection algorithms that maximise the log-likelihood function is presented:

- the decision-directed Maximum-Likelihood (ML) algorithm;

- the digitised version of the Costas loop;

\(^1\)See Chapter 4.
- the decision-directed arctangent algorithm;
- the non data-aided algorithm;
- the Viterbi and Viterbi (V&V) algorithm;
- the proposed optimised decision-directed ML feedforward (FF) algorithm.

The algorithms are compared with respect to their algorithmic complexity and noise performance. Simulation results show that compared to the original decision-directed ML FF algorithm, the optimised algorithm has an improved phase error variance at low to medium signal-to-noise ratio (SNR). By further simulations, it has been shown that the optimised algorithm has a lower phase error variance at medium to high SNR compared to other algorithms.

5.1 FF and FB Configurations

Figure 5.1 shows two configurations that are used to synchronise the phase error. The main elements are a phase error detector (PED), a phase rotator and a smoothing filter. The matched filter output is sampled at the rate $1/T$, where $T$ is the symbol period. In a feedback configuration, the PED finds the difference between the phase of the received signal $Q$ and the phase of the locally generated carrier phase $\hat{\theta}$. The output voltage of the detector is proportional to the difference between the above phases. In a feedforward configuration, the phase error is directly measured from the received signal.

The smoothing filter is set between the PED output and the phase rotator. The filter removes any noise and high frequency components from the output voltage of the PED, thus giving an average (DC) voltage. Assuming the frequency error has been corrected, a first-order loop is sufficient. As there is already an integrator filter in the phase rotator of Figure 5.1, there is no need for the smoothing filter in a first-order loop. This means that the smoothing filter has unity transfer function, i.e. the input and output of the filter are the same. Other choices of the loop filter are discussed in Section 5.8.

In feedback loops, the smoothed voltage is scaled by a loop gain factor $\beta$. The choice of $\beta$ influences the acquisition behaviour of the synchroniser. Small values of $\beta$ result in a longer acquisition time and less tracking jitter, while large values of $\beta$ result in a faster acquisition time and higher tracking jitter. A suitable value of the loop gain factor can be found by simulation.
5.2 Decision-Directed ML Algorithm

Assuming the timing error has been synchronised, the correlation integral between the decision data \( \hat{c}_n \) and the rotated matched filter output is [Gar88a]

\[
R(N, \tilde{\theta}) = \Re \left[ \sum_{n=1}^{N} \hat{c}_n^* p(n) \exp(-j\tilde{\theta}) \right]
\]  

(5.1)

Figure 5.1: Simplified models of phase synchronisation loops.

The last main component of the FF and FB configurations is a phase rotator. A phase rotator consists of a voltage-controlled oscillator (VCO) and a complex multiplier. The baseband model of a VCO consists of an integrator followed by the sin/cos of the integrator output. To rotate the phase of the incoming complex signal by \(-\tilde{\theta}\), the output of the sin(.) term is multiplied by -1 before performing a complex multiplication.
5.2 Decision-Directed ML Algorithm

where $\hat{c}_n$ is given by

$$\hat{c}_n = c_n \exp[j2\pi k / M] \quad k = 0, 1, \ldots, M - 1$$  \hfill (5.2)

The parameter $k$ is the integer that minimises $|\theta - \hat{\theta} - 2\pi k / M|$ (BPSK: $M=2$, (O)QPSK: $M=4$). The sampled matched filter output in the absence of Gaussian noise is

$$p(n) = c_n \exp(j\theta)$$  \hfill (5.3)

Following the acquisition of the error, (5.2) is maximum and therefore, its gradient $\hat{\theta} = \partial R / \partial \theta$ is

$$\hat{\theta} = -3 \left[ \sum_{n=1}^{N} \hat{c}_n^* p(n) \exp(-j\hat{\theta}) \right]$$  \hfill (5.4)

which is ideally zero during tracking. In (5.1), $c_n^* p(n)$ are constant and the real part of the derivative of $\exp(-j\hat{\theta})$ is the same as its imaginary part, as shown in (5.4).

The block diagram is shown in Figure 5.2. The algorithm was developed by Kobayashi [Kob71] as part of a joint carrier phase and symbol timing recovery. For QPSK modulation scheme, $9N$ real multiplications and $5N$ real additions are performed per phase error trial $\hat{\theta}$ in an observation period of $N$ symbols.

![Figure 5.2: Decision-directed ML FB PED.](image)

In the FB loop, the summation in (5.4) is performed by the loop filter and no separate summation is required. The loop filter sums the terms in the brackets and applies the sum as the trial value of the phase error $\hat{\theta}$ to the phase rotator. Eventually, the FB loop drives the estimate to zero.
5.2 Decision-Directed ML Algorithm

To derive an expression for the detector characteristics, (5.2) and (5.3) are substituted in (5.4)

\[
U_e(\Delta \theta) = E \left[ \Re \left[ c_n^* \exp(-j2\pi k / M)c_n \exp(j\theta) \exp(-j\hat{\theta}) \right] \right] \\
= E \left[ \Re \left[ |c_n|^2 \exp(j(\theta - \hat{\theta} - 2\pi k / M)) \right] \right] \\
= E|c_n|^2 \sin(\Delta \theta - 2\pi k / M)
\] (5.5)

where \( E[u_n] \) represents the statistical average of the error. The plot of \( E[u_n] \) over a range of actual errors is known as the S-curve. Ideally, the S-curve of a carrier phase synchroniser must maintain sawtooth shaped with unity slope in the range \( |\pi/4| \). With a unity slope, the synchroniser will have a maximum gain and, hence, the least acquisition time.

![Figure 5.3: S-curve of the decision-directed ML PED.](image_url)

The S-curve of the decision-directed ML FB phase error detector is shown in Figure 5.3. The following comments can be made:

1. The curve repeats 4 times over \( 2\pi \) radian, exhibiting the familiar 4-fold phase ambiguity of QPSK reception. For OQPSK the curve repeats twice in one cycle, which shows that phase ambiguity is only two-fold in the case of
5.2 Decision-Directed ML Algorithm

OQPSK [Ahm92]. Also since there is a two-fold phase ambiguity in BPSK, the curve repeats twice in one cycle of the phase error [Wil93].

2. Although the segments of the noise-free S-curve are really sinusoidal, the segments are in the region of $\sin x = x$, and so the S-curve looks almost like a sawtooth. With this feature, the operating range is approximately $\pi / 4$ radian on either side of the zero phase error.

3. The slope of the synchronisation slope is a function of the SNR. At low SNR the S-curve departs from the fractionally sinusoidal shape to a truly sinusoidal shape. When the decision error probability is small, i.e. for high SNR, the slope approaches unity.

4. The equilibrium points are at $\pm m \frac{\pi}{4}$, where $m = 0, 1, 2, \ldots$. When the phase error is to the right of the null point, the PED output is positive. This will increase the frequency of VCO. Thereby, forcing the phase error to decrease. Similarly, if the phase error is to the left of the null point, the PED output is negative. The frequency of the VCO decreases and, hence, the phase error decreases. The gradient at the equilibrium point represents the PED gain

$$K_\theta = \frac{du_\theta}{d\theta}$$

In Figure 5.4(a) the loop noise power spectral density of the decision-directed ML algorithm is shown for QPSK. The power spectral density is flat. It is observed that the curve for $E_s/N_0 = 6$ dB has the same shape as the curve for $E_s/N_0 = 16$ dB (except for a shift towards larger variance). The influence of the rolloff factor $\alpha$ of the matched filter in Figure 5.1 on the phase error detector characteristics at different SNRs is shown in Figure 5.4(b). At low to medium SNR ($0 \leq E_s/N_0 \leq 7$ dB), the acquisition time is faster as $\alpha$ increases. From medium to high SNR ($7 \leq E_s/N_0 \leq 13$ dB), the slope of the curves becomes independent of $\alpha$. The power spectral density at DC ($fT = 0$) normalised to the square of the slope, is shown in Figure 5.4(c). At $E_s/N_0 \geq 11$ dB, the curve approaches the Cramer-Rao bound [Jes91]

$$\sigma_\phi^2 = \frac{N_0}{2E_s}(2B_L T) \quad \text{(neglecting self-noise)}$$  \hspace{1cm} (5.6)

If in (5.4), the decision values $\hat{c}_n = a_n + j b_n$ are replaced by

$$\hat{c}_n = \text{sgn}(x_n) + j \text{sgn}(y_n)$$  \hspace{1cm} (5.7)
5.2 Decision-Directed ML Algorithm

![Graph](attachment:graph.png)

(a) Noise power spectral density

![Graph](attachment:graph.png)

(b) Slope as a function of $E_s/N_0$

![Graph](attachment:graph.png)

(c) Normalised power spectral density

Figure 5.4: Performance of the decision-directed ML PED.

Chapter 5  Phase Synchronisation Algorithms
where $x_n$ and $y_n$ are the real and imaginary components of the sampled matched filter output, and

$$\text{sgn}(x) = \begin{cases} 
1, & x > 0 \\
0, & x = 0 \\
-1, & x < 0 
\end{cases}$$

then the phase error quantity is

$$u_\theta(n) = \Re \left[ (x_n + jy_n)(\text{sgn}(x_n) - j\text{sgn}(y_n)) \right]$$

$$= y_n\text{sgn}(x_n) - x_n\text{sgn}(y_n)$$

The above algorithm is the digitised version of the Costas loop, and has a simple implementation. For QPSK implementation, only two real multipliers and one real adder are required.

### 5.3 Decision-Directed Arctangent Algorithm

By rewriting the exponential term of (5.1) in rectangular form, a new configuration of the decision-directed ML algorithm is obtained. The derivation is as follows:

$$R(N, \tilde{\theta}) = \Re \left[ \sum_{n=1}^{N} c_n^* p(n) \{ \cos(\theta) - j \sin(\theta) \} \right]$$

(5.10)

By differentiating (5.10) with respect to $\tilde{\theta}$, the following is obtained

$$\frac{\partial R(N, \tilde{\theta})}{\partial \tilde{\theta}} = \sum_{n=1}^{N} \sin(\tilde{\theta}) \Re [c_n^* p(n)] - \cos(\tilde{\theta}) \Im [c_n^* p(n)]$$

(5.11)

At the maximum point the derivative is zero. Therefore

$$\sum_{n=1}^{N} \sin(\tilde{\theta}) \Re [c_n^* p(n)] = \sum_{n=1}^{N} \cos(\tilde{\theta}) \Im [c_n^* p(n)]$$

(5.12)

or

$$\tilde{\theta} = \arg \left( \sum_{n=1}^{N} c_n^* p(n) \right)$$

(5.13)

It is seen from (5.13) that the phase estimate is entirely independent of signal amplitude because of the division operation within the arctangent function. The block diagram of a FB loop employing the algorithm is shown in Figure 5.5. For QPSK modulation scheme, $10N$ real multiplications and $6N$ real additions are performed per phase error trial $\tilde{\theta}$ in an observation period of $N$ symbols. The general
5.4 Non Data-Aided Algorithm

The advantage of using decision-directed arctangent algorithm over the decision-directed ML algorithm of (5.4) is that the acquisition time is about 1 symbol in the decision-directed arctangent algorithm as compared to 15-20 symbols in the decision-directed ML algorithm. The 1 symbol delay is due to the self-scheduling delay\(^2\) in the FB loop. The above acquisition times are only applicable if the channel SNR is infinity and the channel disturbances caused by nonlinearities effects, Doppler shift and timing error have been perfectly synchronised. Further assumption is that the signal and the coefficients in the modem have infinite precision.

5.4 Non Data-Aided Algorithm

With decision-directed carrier phase recovery, decision errors are a problem at low SNR. Furthermore, the acquisition time of the decision-directed algorithms may be a problem in applications in which fast acquisition is required. For these reasons, the use of non-data aided algorithms may be advantageous. The non data-aided algorithm has a feedforward configuration and, as will be shown in the next subsection, the acquisition time is fast.

To derive an expression for the FF algorithm, the log-likelihood function is used

\[ \Lambda(\hat{\theta}) = \sum_{n=1}^{N} \ln \cosh \left( \frac{2E_b}{N_0} \Re \left[ e^{-j\hat{\theta}} p(n) \right] \right) + \ln \cosh \left( \frac{2E_b}{N_0} \Im \left[ e^{-j\hat{\theta}} p(n) \right] \right) \]  

(5.14)

\(^2\)The delay required in a feedback loop in simulation packages such as COSSAP.
5.4 Non Data-Aided Algorithm

By using \( \frac{d}{dx} \ln \cosh(x) = \tanh(x) \), the derivative of (5.14) is found to be

\[
\frac{d\Delta(\theta)}{d\theta} = \sum_{N} \frac{2E_b}{N_o} \mathfrak{E} \left[ e^{-j\theta} p(n) \right] \tanh \left[ \frac{2E_b}{N_o} \mathfrak{R} \left[ e^{-j\theta} p(n) \right] \right] - \sum_{N} \frac{2E_b}{N_o} \mathfrak{R} \left[ e^{-j\theta} p(n) \right] \tanh \left[ \frac{2E_b}{N_o} \mathfrak{I} \left[ e^{-j\theta} p(n) \right] \right]
\]

(5.15)

By substituting

\[
e^{-j\theta} = \cos \theta - j \sin \theta
\]

\[
p(n) = p_r(n) + j p_i(n)
\]

\[
\tanh(x) = x - \frac{x^3}{3} \quad \text{for} \quad \tanh(x) \ll 1
\]

(5.16)

in Equation (5.15) and setting the derivative to zero, the following expression is found

\[
\hat{\theta} = \frac{1}{4} \arctan \left( \frac{\sum_{n=1}^{N} 4p_r(n)p_i(n) [p_r^2(n) - p_i^2(n)]}{\sum_{n=1}^{N} [p_r^2(n) - p_i^2(n)]^2 - 4p_r^2(n)p_i^2(n)} \right)
\]

(5.17)

To simplify the above expression, the output of the matched filter is raised to the fourth power

\[
p^4(n) = \left( p_r^2(n) - p_i^2(n) \right)^2 - 4p_r^2(n)p_i^2(n) + 4p_r(n)p_i(n) \left( p_r^2(n) - p_i^2(n) \right)
\]

(5.18)

Therefore,

\[
\hat{\theta} = \frac{1}{4} \arctan \left( \frac{\sum_{n=1}^{N} \mathfrak{R} \left[ p^4(n) \right]}{\sum_{n=1}^{N} \mathfrak{I} \left[ p^4(n) \right]} \right)
\]

(5.19)

or

\[
\hat{\theta} = \frac{1}{4} \text{arg} \left( \sum_{n=1}^{N} p^4(n) \right)
\]

(5.20)

The block diagram is shown in Figure 5.6. For QPSK modulation scheme, 16\( N \) real multiplications and 8\( N \) real additions are performed in an observation period of \( N \) symbols.
5.4 Non Data-Aided Algorithm

If the matched filter output is squared, rather than quadrupled as above, the phase recovery loop for BPSK results:\(^3\)

\[ \hat{\theta} = \frac{1}{2} \arg \left( - \sum_{n=1}^{N} p^2(n) \right) \]  

(5.21)

5.4.1 Acquisition Performance

Assuming the delays through filters in the transmitter and in the receiver have been removed, the acquisition time in FF algorithms is considerably lower than that of the FB algorithm. The acquisition time in FF algorithms is dominated by the averaging filter. With an averager of length \(N + 1\) symbols, the first fully filtered sample appears at the output of the filter after a delay of \(N\) symbols.

Feedback algorithms have longer acquisition time. The acquisition time is determined by the loop gain factor \(\beta\). The higher the gain factor is, the faster the acquisition time will be. There is an upper limit in the gain factor before the loop becomes unstable. This upper limit may be determined by computer simulations.

To justify the fast acquisition of FF algorithms, as an example, the algorithm in (5.17) is considered. The sampled output of the matched filter is denoted by:

\[
\begin{align*}
P_r(n) &= a_n \cos \theta(n) \\
P_q(n) &= b_n \sin \theta(n)
\end{align*}
\]

(5.22)

where

\(^3\)In deriving the BPSK algorithm, in (5.16) \(\tanh(x) \approx x\) is used.

Chapter 5 Phase Synchronisation Algorithms
\( a_n \) is the \( n \)-th real component of the transmitted data.

\( b_n \) is the \( n \)-th imaginary component of the transmitted data.

\( \theta \) is the phase error.

Assuming the transmission is ISI free, then

\[
a_n = b_n = \pm \frac{1}{\sqrt{2}}
\]

(5.23)

Substituting (5.22) in (5.17) results in

\[
\hat{\theta} = \frac{1}{4} \arctan \left( \frac{\sum_{n=1}^{N} 4a_n b_n \sin \theta(n) \cos \theta(n) \left( a_n^2 \cos^2 \theta(n) - b_n^2 \sin^2 \theta(n) \right)}{\sum_{n=1}^{N} \left( a_n^2 \cos^2 \theta(n) - b_n^2 \sin^2 \theta(n) \right) - 4a_n^2 \cos^2 \theta(n) b_n^2 \sin^2 \theta(n)} \right)
\]

(5.24)

But

\[
a_n^2 = b_n^2 = \frac{1}{2}
\]

(5.25)

and

\[
a_n b_n = \pm \frac{1}{2}
\]

(5.26)

Therefore (5.24) is simplified to

\[
\hat{\theta} = \frac{1}{4} \arctan \left( \frac{\sum_{n=1}^{N} \pm 2 \sin 2\theta(n) \cos 2\theta(n)}{\sum_{n=1}^{N} \cos^2 2\theta(n) - 4 \cos^2 \theta(n) \sin^2 \theta(n)} \right) \pm \sum_{n=1}^{N} \sin 4\theta(n)
\]

\[
= \frac{1}{4} \arctan \left( \frac{\sum_{n=1}^{N} \pm \sin 4\theta(n)}{\sum_{n=1}^{N} \cos 4\theta(n)} \right)
\]

\[
= \frac{1}{4} \arctan (\pm \sum_{n=1}^{N} \tan 4\theta(n))
\]

(5.27)

or

\[
\hat{\theta} = \pm \frac{1}{N} \sum_{n=1}^{N} \theta(n)
\]

(5.28)

Hence, ideally the output of the error detector must be the same as the actual phase error. However, due to roundoff errors the first output sample is slightly different from the actual error. As an example, in a simulation of the non data-aided algorithm the first trial value of the phase error was 0.3999 radian for an actual phase error.
error of 0.4 radian. However with the decision-directed ML FB algorithm, the first trial error was close to zero and it took about 20 symbols for the loop to acquire the error. The acquisition time of 20 symbols was obtained with a high loop gain factor $\beta$. In the presence of AWGN, the loop gain has to be decreased to avoid the loop becoming unstable. By decreasing the value of $\beta$, the acquisition time increases.

![Signal-to-noise ratio, Eb/No vs. BER](image)

**Figure 5.7: BER performance of the non data-aided algorithm.**

Although the non data-aided algorithm has a fast acquisition and requires a low number of arithmetic operations to detect the phase error, its BER performance can be poor. In Figure 5.7 the noise performance with two different values of loop filter has been shown. With an averaging length of 10 symbols, the BER performance is noticeably poor. An improvement in the performance has been achieved by increasing the averaging length. However, a high averaging length at high SNR is not desirable.

### 5.5 Viterbi and Viterbi (V&V) Algorithm

The V&V algorithm is the generalisation of the non-data aided ML feedforward phase error synchroniser. The algorithm was proposed in [Vit83] and was further analysed in [Pad86]. The block diagram is shown in Figure 5.8
5.5 Viterbi and Viterbi (V&V) Algorithm

The operation of the V&V algorithm is as follows:

By a rectangular-to-polar conversion (R/P), the phase \( \varphi \) and the magnitude \( u \) of the sampled matched filter output are obtained. The function \( V(u) \) raises \( u \) to an arbitrary integer power \( z \), where

\[
z = 0, 1, \ldots
\]

With \( z = 4 \) and \( z = 2 \), the V&V algorithm becomes the same as the non-data aided ML algorithm for QPSK and BPSK modulation schemes respectively\(^4\). By multiplying the phase by \( M \), the data modulation is removed. Next, a polar-to-rectangular conversion (P/R) is performed to obtain

\[
P_r(n) = V(u) \cos(M \varphi)
\]

\[
P_i(n) = V(u) \sin(M \varphi)
\] (5.29)

The argument of the averaged components in (5.29), scaled by \( 1 / M \), is the estimate of the phase error

\[
\hat{\vartheta} = \frac{1}{M} \arg \left( -\frac{1}{N} \sum_{n=1}^{N} p'(n) \right)
\] (5.30)

or

\[
\hat{\vartheta} = \frac{1}{M} \arg \left( -\frac{1}{N} \sum_{n=1}^{N} V(u) \exp(jM \varphi) \right)
\] (5.31)

\(^4\)See Equations 5.20 and 5.21.
Depending on the type of the nonlinearity \( V(u) \) in Equation (5.31), the number of arithmetic computations performed in a QPSK synchroniser are shown in Table 5.1.

Table 5.1: Computational complexity of the V&V phase synchroniser.

<table>
<thead>
<tr>
<th>( V(u) )</th>
<th>real multiplications</th>
<th>real additions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8N</td>
<td>4N</td>
</tr>
<tr>
<td>( u )</td>
<td>10N</td>
<td>4N</td>
</tr>
<tr>
<td>( u^2 )</td>
<td>12N</td>
<td>4N</td>
</tr>
<tr>
<td>( u^4 )</td>
<td>16N</td>
<td>4N</td>
</tr>
</tbody>
</table>

The extra complexity of the algorithm is due to the square root of \( p_i^2(n) + p_j^2(n) \) in calculating the magnitude \( u \). Reference [Tel89] gives some DSP routines to perform the above calculation. It is faster to use a look-up table in a ROM than performing a number of computations.

Figure 5.9: Normalised PSD of the V & V algorithm.

The variations of the noise power spectral density at \( fT = 0 \) are shown in Figure 5.9. The best performance is obtained by using \( V(u) = u^2 \) and \( u \). Since
5.6 Optimisation of the Decision-Directed ML FF Algorithm

$V(u) = u$ does not include nonlinearity, it is preferred over $V(u) = u^2$. The worst response at all values of $E_s/N_0$ is when $V(u) = u^4$ is used. At $E_s/N_0 \geq 6$ dB, the noise performance of $V(u) = 1$ approaches that of $V(u) = u^2$ and $u$.

Although the V&V algorithm was originally derived as a FF carrier synchroniser, a FB algorithm can be implemented by analogy with the non-data aided ML algorithm. For BPSK ($M=2$) and QPSK ($M=4$) the FB synchronisation algorithm is given by [Jes91]

$$u_\phi(n) = \frac{1}{M} \sum \left[ V(u) \exp \left( jM\phi - jM\bar{\theta} \right) \right]$$

(5.32)

The tracking performance of the FB algorithm is the same as that of the FF algorithm. Faster acquisition time of the FF configuration, makes the original algorithm more attractive to use than the FB configuration.

**5.6 Optimisation of the Decision-Directed ML FF Algorithm**

![Diagram](image)

Figure 5.10: Original decision-directed ML FF PED.

In [Jes91] the feedforward configuration of the decision-directed phase error synchroniser, shown in Figure 5.10, was proposed. By careful investigation one can see that the loop has both feedforward and feedback paths. The feedback path has been marked from 1 to 5. A feedforward path is marked from A to G. The conventional FB loop works to drive the phase error to zero. In this algorithm
there is no distinct error measurement and the loop does not drive anything to null at equilibrium. Assuming the SNR is infinity, the minimum number of delay elements is one symbol. In satellite channels, the SNR is small (e.g. $E_b/N_0 < 10 \text{ dB}$). Therefore an averaging must be performed on the estimate of the error. In the original algorithm an integrate-and-dump (I&D) averaging filter has been used. With an I&D filter of length $N$, $N$ samples of the estimated phase error are read and the average is written to the output port. This means that the I&D filter reduces the sampling rate by a factor $N$. Therefore, the rate at the output of the filter must be increased by repeating the averaged sample $N$ times. Furthermore, with an averaging length of $N$ symbols, there is a delay of $N - 1$ symbols. With regards to eliminating the delay, an improvement to the original proposal has been shown in Figure 5.11.

![Figure 5.11: Proposed optimised decision-directed ML FF PED.](image)

The proposed optimised algorithm consists of only feedforward paths. No delay elements are required. The expression describing the behaviour of the loop is:

$$\hat{\theta} = \arg\left(\frac{1}{N} \sum_{n=1}^{N} p(n)\hat{e}_n^*\right)$$  \hfill (5.33)

or

$$\hat{\theta}_{QPSK} = \arg\left(\frac{1}{N} \sum_{n=1}^{N} [p_r(n)\hat{\alpha}_n + p_i(n)\hat{\beta}_n] + j[p_r(n)\dot{\alpha}_n + p_i(n)\dot{\beta}_n]\right)$$  \hfill (5.34)

$$\hat{\theta}_{BPSK} = \arg\left(\frac{1}{N} \sum_{n=1}^{N} p_r(n)\hat{\alpha}_n\right)$$  \hfill (5.35)

$$\hat{\theta}_{OQPSK} = \arg\left(\frac{1}{N} \sum_{n=1}^{N} [\hat{\alpha}_n p_r(n) + \hat{\beta}_n q_i(n)] + j[\dot{\alpha}_n p_r(n) - \dot{\beta}_n q_i(n)]\right)$$  \hfill (5.36)

where

Chapter 5 Phase Synchronisation Algorithms
\[ a_n \text{ is the } n\text{-th real component of the transmitted data.} \]

\[ b_n \text{ is the } n\text{-th imaginary component of the transmitted data.} \]

\[ \hat{a}_n \text{ and } \hat{b}_n \text{ are the receiver’s decision on } a_n \text{ and } b_n. \]

The simulation results of the variance of the above algorithms for BPSK and (O)QPSK modulation schemes are shown in Figure 5.12. The normalised loop noise bandwidth \( B_L T \) is 0.05. This bandwidth corresponds to having an integrate-and-dump filter of length 10 symbols. In all cases, the optimised algorithm has a better performance at low SNR.

The slightly poor performance at very low to moderate SNR with the decision-directed ML FF algorithm is due to feeding back the phase rotated symbols. However, in the optimised algorithm the phase rotated signal does not influence the noise performance; the phase rotator is not in the signal path in the optimised algorithm.

At medium to high SNR, OQPSK has the highest phase error variance compared to BPSK and QPSK. This is because unlike BPSK and QPSK, OQPSK suffers from self noise even at high SNR. BPSK and QPSK are self-noise free and have approximately the same noise variance. One way to improve the noise performance of OQPSK is by increasing the averaging length of the loop filter. This in turn increases the synchronisation delay.

Further simulations were done in which the decision values were replaced by the sign of the sampled matched filter output. The expression in (5.33) is modified as follows

\[ \hat{\theta} = \arg \left( \frac{1}{N} \sum_{n=1}^{N} p(n)\text{sgn}(p^*(n)) \right) \quad (5.37) \]

where \( \text{sgn}(x) \) has been defined in (5.8). This approach simplifies the implementation. There is no need to determine to which constellation point the received signal is closer to, in order to make a decision on transmitted data. The computation requirement of the original and the optimised algorithm are the same. For QPSK modulation scheme, \( 10N \) real multiplications and \( 6N \) real additions are performed in an observation period of \( N \) symbols.
5.6 Optimisation of the Decision-Directed ML FF Algorithm

Figure 5.12: Performance of the decision-directed ML FF PED algorithm before and after optimisation.
5.7 Noise Performance of Phase Synchronisers

It is interesting to compare the noise performance of the optimised decision-directed ML FF, Costas, decision-directed arctangent, non-data aided ML, and V&V algorithms. The normalised power spectral density for each algorithm is shown in Figure 5.13. For satellite communications a suitable algorithm must be capable to operate satisfactorily with \( 7 \leq \frac{E_s}{N_0} \leq 13 \, \text{dB} \). It is seen that the optimised decision-directed ML FF algorithm has a better performance in the above range. The difference in performance marginally increases as the SNR increases. Although all the curves approach the Cramer-Rao bound (CRB) at high SNR, the optimised algorithm has a noticeably lower self-noise. This is because as the SNR increases, the decision errors decrease. The poor performance of the non data-aided algorithm is the same as that of the Viterbi&Viterbi algorithm with \( V(u) = u^4 \).

![Figure 5.13: Normalised PSD of the optimised decision-directed ML FF, Costas, decision-directed arctangent, non-data aided ML, and V&V algorithms.](image)

The results in Figure 5.13 are an indication that the proposed optimised algorithm and the non data-aided algorithms have the lowest and the highest tracking jitter. The tracking jitter has a direct impact on the BER performance of the algorithm. The
tracking jitter of the non data-aided algorithm can be improved by increasing the loop filter averaging length. The effect of the averaging length has been demonstrated in Figure 5.7. A suitable averaging length can be found by simulation.

So far in this chapter, the details of phase error detection algorithms were presented. In the next section, the design of loop filters for the synchronisers of Figure 5.1 is discussed.

5.8 Design of the Loop Filter

5.8.1 FF Loops

The general form of the smoothing filter for FF algorithms is a symmetrical averager. For the input $x(n)$, the output $y(k)$ is determined from:

$$y(k) = \frac{\sum_{i=1}^{N_1} x(k - i) + \sum_{j=1}^{N_2} x(k + j)}{N_1 + N_2 + 1} \quad (5.38)$$

where $N_1$ and $N_2$ are the number of samples before and after the sample whose phase is to be estimated. From the above expression, one has the option of performing the averaging over

(i) past and present samples, $N_2 = 0$,

(ii) past, present and future samples, or

(iii) present and future samples, $N_1 = 0$.

The filtering delays are $N_1$ symbols, $N_1 + N_2$ symbols, and $N_2$ symbols for cases (i) to (iii), respectively. Viterbi and Viterbi [Vit83] have pointed out that a symmetric smoother provides an estimate that is not biased in the presence of an uncorrected frequency offset. Gardner [Gar88a] has proved the above property for BPSK and can be extended to QPSK. If $N_2$ in (5.38) is set to zero, the integrate-and-dump filter results.

5.8.2 FB Loops

Assuming the Doppler shift in the received signal has already been removed by some technique, a first order PLL is sufficient to synchronise the phase error. Since there is already an integrator in the phase rotator\(^5\), there is no need for a loop filter.

\(^5\)see Figure 5.1.
5.8 Design of the Loop Filter

In applications in which small frequency offsets must be corrected by a PLL, a second order loop filter with proportional-plus-integral control transfer function can be used. The block diagram of this filter is shown in Figure 5.14.

Figure 5.14: Second-order loop filter.

The transfer function is [Gar79]

\[ F(s) = K_p + \frac{K_i}{s} \]  

(5.39)

where

\[ K_i = \left( \frac{2B_lT}{\xi + \frac{1}{4\zeta^2}} \right)^2 \]  

(5.40)

and

\[ K_p = \frac{4B_lT}{1 + \frac{1}{4\zeta^2}} \]  

(5.41)

The performance of the loop is controlled through changing the damping factor \( \xi \) and the normalised loop noise bandwidth \( B_lT \). The choice of \( \xi \) is a compromise between the overshoot and the acquisition time. Lower values of \( \xi \) result in higher overshoots and a longer acquisition time. With \( \xi = \frac{1}{\sqrt{2}} \) the acquisition time is approximately \( 4/B_lT \) symbols (see Figure 5.15). It is also possible to split the filter into two branches [Ahm92]. The branch with \( k_p \) is connected to an automatic frequency control (AFC) VCO. The other branch is connected to the PLL VCO. Since \( K_p \) is normally small, the PLL output dominates the AFC output after the frequency error has been acquired.

Recently Riera and Mercader [Rie94] studied several aspects relevant to the design of phase synchronisation for land mobile satellite systems (LMSS). Their work follows the classical results that if closed-loop phase lock synchronisers are chosen, a third-order loop is optimum in the sense that it minimises the root mean squared (rms) error between the input phase when the received signal consists of...
5.9 Summary

In this chapter a survey of different carrier phase synchronisation algorithms was presented. A comparison between the decision-directed ML FF algorithm and its newly optimised version was presented. It was shown that for QPSK modulation scheme at low SNR, the optimised decision-directed ML FF algorithm has a superior noise performance compared to the original algorithm. Other algorithms considered were the decision-directed ML FB algorithm, the digitised version of the Costas loop, the non-data aided ML algorithm, the Viterbi and Viterbi algorithm and the decision-directed arctangent algorithm. At low SNR, the Viterbi-Viterbi algorithm with \( V(u) = u \) has the lowest noise variance. For medium to high values of SNR, the optimised decision-directed ML FF algorithm has the best performance; the performance approaches the Cramer-Rao bound. For QPSK modulation scheme, all the surveyed algorithms need only 1 sample per symbol to operate. From the

![Performance of the second-order loop with different \( \xi \).](image)

Figure 5.15: Performance of the second-order loop with different \( \xi \).

a frequency ramp combined with AWGN [Jaf55]. The experimental results show that in LMSS, third-order loops give better performance in the presence of Doppler shift, but they suffer from a higher sensitivity to oscillator phase noise. The model of the third-order filter used was implemented as the cascade of two proportional-plus-integral filters shown in Figure 5.14. From some preliminary design examples, satisfactory results with second-order PLLs were obtained. Therefore, third-order PLLs will not be used in the remaining of this thesis.
5.9 Summary

Noise performance simulation results presented in Figure 5.13, the two most suitable candidates qualified for further investigation in Chapter 6 are the digitised version of the Costas loop and the proposed algorithm. Although the noise performance of the decision-directed arctangent algorithm at medium to high SNR is approximately the same as that of the Costas loop, the algorithmic complexity of the Costas loop in terms of the number of additions and multiplications is less.

Table 5.2: Algorithmic complexity of the phase synchronisers surveyed.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>real multiplications</th>
<th>real additions</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decision-directed ML</td>
<td>9N</td>
<td>5N</td>
<td>FB</td>
</tr>
<tr>
<td>Costas loop</td>
<td>9N</td>
<td>5N</td>
<td>FB</td>
</tr>
<tr>
<td>Decision-directed arctangent</td>
<td>10N</td>
<td>6N</td>
<td>FF</td>
</tr>
<tr>
<td>Non data-aided</td>
<td>16N</td>
<td>8N</td>
<td>FF</td>
</tr>
<tr>
<td>V&amp;V, $V(u) = 1$</td>
<td>8N</td>
<td>4N</td>
<td>FF</td>
</tr>
<tr>
<td>V&amp;V, $V(u) = u$</td>
<td>10N</td>
<td>4N</td>
<td>FF</td>
</tr>
<tr>
<td>V&amp;V, $V(u) = u^2$</td>
<td>12N</td>
<td>4N</td>
<td>FF</td>
</tr>
<tr>
<td>V&amp;V, $V(u) = u^4$</td>
<td>16N</td>
<td>4N</td>
<td>FF</td>
</tr>
<tr>
<td>Proposed Decision-directed ML</td>
<td>10N</td>
<td>6N</td>
<td>FF</td>
</tr>
</tbody>
</table>

In Table 5.2 the computation requirement of each algorithm over an observation period of $N$ symbols have been tabulated. The following assumptions are made:

1. Modulation scheme is QPSK.
2. A first order loop has been used.
3. The results of trigonometric functions are found by look-up tables.

Furthermore, the tabulated results do not include the operations required to make a decision on the transmitted data.

The Costas loop has a slightly lower computational requirement than the proposed algorithm. More important than this marginal advantage in algorithmic complexity, is the poor acquisition time of the Costas loop.

The results presented in this chapter were based on using a single-channel modem under the assumptions of perfect carrier frequency and symbol timing synchronisation. In Chapter 6, the selected carrier phase recovery algorithms will be compared in an 8-channel MCDD whose received FDM signal is subject to both carrier frequency and symbol timing errors.

Chapter 5 Phase Synchronisation Algorithms
Chapter 6

An Optimised 8-Channel Recursive (IIR) MCDD

In the last four chapters, a survey of some of the available demultiplexing and synchronisation algorithms was carried out in order to identify the most suitable ones for comparisons with the self-proposed algorithms. In this chapter, the simulation results of an optimised 8-channel MCDD will be presented. For the purpose of resolving the phase ambiguity in coherent demodulators, a new heuristic technique has been developed.

In presenting the simulation results in the previous chapters, the following assumptions were made:

- Chapter 3 Perfect timing and phase synchronisation
- Chapter 4 Perfect frequency and phase synchronisation
- Chapter 5 Perfect frequency and timing synchronisation

In practice, a multicarrier demultiplexer demodulator (MCDD) must be designed to detect and correct the carrier and bit timing errors in the received signal. In the previous chapters, there was no coupling between the algorithms analysed. However, the aim of this chapter is to develop an MCDD in which various synchronisation algorithms interact. A frequency division multiplexer to multiplex 8 statistically independent channels at 2 different sampling rates will be developed in order to make a comparison between the proposed IIR demultiplexer [Dan94a, Dan94b] and a typically available FIR demultiplexer [Göc88, Qi 92]. Since coherent demodulation will be performed on the received signal, it is essential to address the problem of...
phase ambiguity. One method of resolving the phase ambiguity is by using unique words [Feh83]. In this chapter, a detailed heuristic method will be presented for detecting the unique words from the received data and the synchronisation (SYNC) sequence. This will be followed by the selection of the most suitable frequency error synchronisation algorithm with respect to acquisition time and tracking jitter. The candidates are the Alberty and Hespelt (A&H) dual filter discriminator [Alb89] and the self-proposed algorithm. The analysis in Chapter 3 showed that the A&H algorithm is self-noise free, and it has a constant gain over its acquisition range. Furthermore, it was shown in Chapter 3 that the proposed frequency error synchroniser has a relatively lower hardware and algorithmic complexity and a better noise performance. In this chapter, these two algorithms will be compared with respect to their speed of acquisition and their tracking jitter in the presence of Gaussian noise. It will be shown that the self-proposed algorithm has a superior performance with regard to acquisition and tracking of the frequency error.

In Chapter 4, a comparison between symbol timing error detection and correction algorithms was made. The Gardner algorithm [Gar86] and the self-proposed optimised modified Mueller and Müller (mM&M) algorithm were selected for further investigation in this chapter. It was shown that the Gardner algorithm can operate without a prior knowledge of the carrier phase error. However, the operation of the proposed algorithm is affected by the carrier phase error. A brief discussion on joint carrier phase and symbol timing recovery will be presented. It will be shown that under the same conditions, the proposed algorithm has a superior tracking performance compared to the Gardner algorithm.

The last synchronisers to be compared are the Costas loop and the proposed decision-directed maximum-likelihood (ML) feedforward phase error synchronisers, which were selected after a comprehensive survey of different phase error synchronisers. The main selection criterion was the noise performance. It was shown that the proposed algorithm has a near to optimum noise performance, while the Costas loop has the next best noise performance among other phase synchronisers. It will be shown that the proposed algorithm has a faster acquisition response and a noticeably lower tracking jitter in the presence of Gaussian noise.

Finally, the BER performance of the MCDD developed with the selected demultiplexing and synchronisation algorithms will be presented. 

Chapter 6 An optimised 8-channel IIR MCDD
6.1 Frequency Division Multiplexing

A simple method to perform frequency division multiplexing (FDM) of $K$ statistically independent, contiguous, uniformly spaced channels is shown in Figure 6.1. It is applicable to BPSK, (O)QPSK, and $\pi/4$-QPSK modulation schemes. The operation of the multiplexer is as follows:

The sources generate pseudo-random binary data with equal probabilities of ones and zeros. The shift register of each source has a unique starting value. Therefore, the channels are statistically independent. The pseudo-random binary data of each channel is fed to a modulator block which performs the basic operations such as Gray coding, signal mapping, up-sampling, and the bandlimiting of the signal before transmission. The output of the modulator blocks are at baseband with a sampling rate of $f_k = 2R$, where $R$ is the symbol rate. The channels occupy the same bandwidth, but are spatially separate and their output samples are generated at the same time.

To generate an FDM signal, a time domain to frequency domain translation is performed by the TDM-FDM block, which consists of a bank of rational sampling rate convertors and a bank of single-side band (SSB) modulators. By a rational sampling rate conversion, the sampling rate of each channel is increased to the
6.1 Frequency Division Multiplexing

The function of the SSB modulators is to shift the signal to the appropriate frequency band. Denoting the up-sampled real and imaginary components of the \( k \)-th channel by \( x_k(n) \) and \( y_k(n) \), respectively, the FDM signal is

\[
x_{fdm}(n) + jy_{fdm}(n) = \sum_{k=0}^{K-1} [x_k(n) + jy_k(n)] e^{2\pi jnWT_s}
\]

where

\[ x_{fdm} \text{ and } y_{fdm} \text{ are the real and imaginary components of the FDM signal.} \]

\[ K \text{ is the number of channels.} \]

\[ W \text{ is the channel spacing.} \]

\[ T_s \text{ is the FDM sampling period.} \]

\[ n \text{ is the index to samples.} \]

The average power of the FDM signal is \( K \) times higher than that of each individual channel. Therefore, the output of the TDM-FDM block is scaled by \( 1 / \sqrt{K} \) to normalise the power. In the next section the choice of the FDM sampling rate is discussed.

6.1.1 FDM Rate

Depending on the demultiplexing algorithm, the FDM signal is either critically sampled or oversampled. Figure 6.2(a) shows an FDM signal consisting of 8 channels which have been oversampled by a factor of two. The normalised rate is

\[
f_s = 2KW = 32R
\]

where \( R \) is the symbol rate. For this particular case, \( L = 16 \) and \( M = 1 \) in the TDM-FDM translator.
6.2 Proposed Recursive (IIR) Tree Demultiplexer

Figure 6.2: FDM signal, \( K = 8 \).

Figure 6.2(b) shows an FDM signal which has been critically sampled. The rate is

\[
f_s = KW = 16R
\]  

(6.4)

The parameters \( L \) and \( M \) for the above rate are \( L = 8 \) and \( M = 1 \). In the next section the demultiplexing of the FDM channels by a new scheme based on using a recursive (IIR) design algorithm to demultiplex the critically sampled QPSK channels for OBP satellites is presented.

6.2 Proposed Recursive (IIR) Tree Demultiplexer

The block diagram of the IIR Demultiplexer and the corresponding theoretical spectra at different points are shown in Figures 6.3 and 6.4. The basic building blocks of the demultiplexer are the halfband filters (HBF) and the spectral shifters. The IIR demultiplexer recovers the FDM channels in 3 stages. In general, with \( K \) channels, the demultiplexing is performed in \( \log_2 K \) stages\(^1\) in the binary tree structure. At every stage the spectrum is halved; the side channels are in one half and the middle channels are in the other half. The side channels occupy the spectrum in the frequency range from \( f = 0 \) to \( f = f_s/4 \) and from \( f = 3f_s/4 \) to \( f = f_s \). Similarly, the middle channels occupy the frequency band from \( f = f_s/4 \) to \( f = 3f_s/4 \).

\(^1\)With tree structures, \( K \) is a positive integer power of 2.
For filtering the side channels no spectral shift is required. However, for filtering the middle channels, the spectrum is shifted by $-0.5f_s^n$, where $f_s^n$ is the sampling frequency at the $n$-th stage. After filtering, the FDM signal is decimated by a factor of 2.

For the correct operation of the MCDD, it is necessary that the channels are arranged in odd stacking. The spectrum at point “A” has an even stacking arrangement. By shifting the FDM signal by half the channel spacing, $(-0.5W)$, the spectrum will have an even stacking arrangement at point “B”.

As an illustrative example of why the channel arrangement must be changed to an odd stacking, the spectrum at point “A” in Figure 6.4 is considered. With an even channel stacking, channels 3 and 7 will be distorted by the filtering action in the first stage of the demultiplexer and cannot be recovered. This filter has been shown with heavy lines. However, at point “B” channel 7 is fully recovered, while channel 3 is filtered out. The odd stacking arrangement is maintained throughout the demultiplexer up to the last stage. Before filtering is performed in the last stage,
6.2 Proposed Recursive (IIR) Tree Demultiplexer

Figure 6.4: Theoretical spectra at different points of the 8-channel IIR tree demultiplexer

the stacking is made even.

6.2.1 Design of the Prototype Recursive (IIR) Filter

The designed polyphase low-pass filter consists of first-order (in $z^L$) all-pass sections with the $z$-transfer function

$$H(z) = 1 + z(z^L)^{N-1} \prod_{i=1}^{N} \frac{c_i z^L + 1}{z^L + c_i}$$

(6.5)

where

$N = 2$ is the filter order.

$L = 2$ is the decimation factor.

$c_i$'s are the square of the filter complex poles that are found by the unconstrained optimisation of [Fle63].

Chapter 6 An optimised 8-channel IIR MCDD
Referring to the discussion in Chapter 2, other parameters used in the design of the prototype filter are:

Estimate of the poles: $c_1 = -0.1017$ and $c_2 = 0.1017$

The desired degree of accuracy $\epsilon$ with which the minimum point is determined: 0.01

Weighting function in the passband and stopband = 1

Pass-band frequency = 2.8R

Stop-band frequency = 5.2R

With the above design specifications, the optimised filter coefficients were found to be $c_1 = -0.579166$ and $c_2 = 0.121871$. The responses of the filter are shown in Figure 6.5.

The designed filter has a stopband attenuation of approximately 35dB. Although it is possible to increase the filter order to improve the frequency response, this will lead to additional demultiplexing delay. The designed filter has an approximately linear phase response. The peak-to-peak variation of the phase response in the passband is less than 0.0053 radian. The group delay variation in the passband is about $0.1047\frac{T}{R}$, where $T = \frac{1}{f_s}$ is the symbol period. It is near the passband edge that the filter shows a large variation in the phase response and the group delay. Therefore, with the IIAR design algorithm of [Ans83] the area near the passband edge must be avoided.

6.3 Nonrecursive (FIR) Tree Demultiplexer

The block diagram of the demultiplexer and the theoretical spectrum at different points of the demultiplexer are shown in Figures 6.6 and 6.3. The basic building blocks of the demultiplexer are the high-pass filters (HPF), the low-pass filters (LPF), the post-processing filters, and the spectral shifters. At each node of the tree there is a low-pass filter and a high-pass filter, which are followed by a sampling rate compression by a factor of two. The channels in the frequency band from $f = 0$ to $f = f_s/4$ are recovered by the low-pass filter, while the channels in the frequency band from $f = f_s/4$ to $f = f_s/2$ are recovered by the high-pass filter. The spectra at points "A" and "B" have not been shown. The spectra at these points are the same as those shown in Figure 6.4, except that the sampling rate is $32R$ and the spectrum from $16R$ to $32R$ is empty.

\[2\text{ The spectra at points "A" and "B" have not been shown. The spectra at these points are the same as those shown in Figure 6.4, except that the sampling rate is } 32R \text{ and the spectrum from } 16R \text{ to } 32R \text{ is empty.} \]
6.3 Nonrecursive (FIR) Tree Demultiplexer

Figure 6.5: Responses of the designed prototype IIR filter

frequency band from $f = f_s / 4$ to $f = f_s / 2$ are recovered by the high-pass filter. Furthermore, the transition bands of the low-pass filter distort the channels in the frequency bands from $f = f_s / 4$ to $f = f_s / 2$ and from $f = 3f_s / 4$ to $f = f_s$. Similarly, the channels in the frequency band from $f = 0$ to $f = f_s / 4$ and from $f = f_s / 2$ to $f = 3f_s / 4$ are distorted by the transition bands of the high-pass filter. After performing the high-pass filtering and compressing the sampling rate, the spectrum is shifted by $-0.5f_s^n$. This ensures that the channels which have been recovered by the high-pass filter will be within the filtering range of the low-pass and the high-pass filters at the next stage of the tree structure. Unlike the proposed IIR tree structure in which the channels are fully recovered in the 3rd stage, with the FIR demultiplexer post-processing is required to recover the channels and to correct the
6.3 Nonrecursive (FIR) Tree Demultiplexer

Figure 6.6: Block diagram of the 8-channel FIR tree demultiplexer

sampling rate. Referring to Figure 6.3 at points “I” to “P”, a post-processing filter is simply a low-pass filter to recover channels \( k_o \) from channels \( k_{o+1} \), where \( k_o \) is an odd channel number. The same post-processing filter can be used to recover channels \( k_e \) from channels \( k_{e-1} \), where \( k_e \) is an even channel number.

6.3.1 Design of the Prototype Nonrecursive (FIR) Filter

The design of the filters is based on the Parks-McClellan linear-phase, equiripple FIR filter design algorithm. The actual algorithm has been described in [Rab75, Opp89]. The algorithm has been implemented as part of a package called DFDP [Atl86]. Using this algorithm, FIR prototype filters for the tree structure have been designed. The designed halfband and post-processing filters have a length of 11 and 15 coefficients, respectively. With an 8-bit quantisation (including the sign bit), the stopband attenuations are about 37 dB and 32 dB for the halfband and post-processing filters, respectively.

For COSSAP simulations, the filters designed in DFDP must be reformatted.
Figure 6.7: Theoretical spectrum at different points of the halfband FIR demultiplexer.

The details are presented in Appendix D.

### 6.4 Simulation Results of Demultiplexing

To ensure that the developed demultiplexing COSSAP models recover the channels as expected, functional tests of the models were performed. By investigating the theoretical spectra in Figure 6.4, it can be seen that channel 4 is subject to the highest
adjacent channel interference (ACI) from either side\(^3\). Therefore the demultiplexing stages leading to the recovery of this channel are presented in Figures 6.8 and 6.9.

The power spectral density of a channel does not give any information regarding the possible change in the characteristics of the data by the demultiplexer. If the demultiplexer is not designed properly, it may, for example, rotate the phase of the data. To ensure that the data have not been corrupted, each channel was demodulated and BER tests were performed over a long observation period in the presence of Gaussian noise. The COSSAP simulation results for channel 4 have been shown in Figures 6.8(e) and 6.9(f). The “ideal” BER curves in these figures represent the probability of error \( P_e \) for uncoded QPSK coherent demodulation, which is given by [Mar93]

\[
P_e = 0.5 \text{erfc} \sqrt{\frac{E_b}{N_0}}
\]  

The noise model in COSSAP adds discrete-time zero-mean white Gaussian noise of variance

\[
\sigma^2 = 10^{-\text{SNR}/10} \times \text{mean signal power}
\]

(6.7)

to the input complex signal. The above expression must be modified to take into account other factors such as the number of samples per symbol, \( SPS \), the number of statistically independent frequency division multiplexed channels, \( K \), the overall coding rate, \( R_c \), and the ratio of symbol duration to bit duration, \( Z \). The modification is performed by scaling (6.7) by a factor \( F \) where

\[
F = \frac{SPS}{Z \times K \times R_c}
\]

(6.8)

For an 8-channel demultiplexer in which the channels are uncoded QPSK, the scaling factor \( F \) is 1 and 2 for the critical sampling and the oversampling cases respectively.

\(^3\)This is also true for channel 5.
6.4 Simulation Results of Demultiplexing

![Simulation Results of Demultiplexing](image)

(a) Point B  
(b) Point D  
(c) Point G  
(d) Point N  
(e) BER performance of channel 4

Figure 6.8: Simulation results of demultiplexing the proposed IIR tree structure.
6.4 Simulation Results of Demultiplexing

Figure 6.9: Simulation results of demultiplexing the FIR tree structure.
6.5 Effect of Quantisation on the Performance of MCDDs

The results in Figures 6.8(e) and 6.9(f) were obtained with unquantised filter coefficients. To investigate the performance of the IIR and FIR MCDDs with finite precision filter coefficients, the simulations in Figure 6.8(e) and 6.9(f) were repeated with different word-lengths ranging from 16 to 5 bits. The BER simulation results with filter coefficients quantised to 5 bits (excluding the sign bit) are shown in Figure 6.10(a). Comparing these results with those given in Figures 6.8(e) and 6.9(f), one can see that quantisation has resulted in a poor noise performance in the case of the FIR MCDD. At a BER of $10^{-3}$, the simulated FIR curve is about 0.67 dB worse than the theoretical results. However in the case of the IIR demultiplexer, the degradation in the BER performance is negligible. The superior noise performance of the IIR demultiplexer is due to the fact that the IIR prototype filter has a lower number of coefficients than the FIR prototype filter. Secondly, the analysis and simulations in Chapter 2 showed that by appropriate realisation of the IIR filters in the MCDD one can achieve a low noise variance.

The effect of quantisation on the BER performance can also be demonstrated by plotting the frequency response of the prototype filters. The responses before and after quantisation for each prototype filter have been shown in Figures 6.10(b) and 6.10(c). The reduction in the stopband attenuation for the case of the FIR filter is higher than that for the case of the IIR filter. From these frequency responses one can deduce that if nonlinearities are present in the channel causing spectral spreading, the IIR MCDD performs better than the FIR MCDD because the IIR prototype filter maintains its stopband attenuation.
6.5 Effect of Quantisation on the Performance of MCDDs

Figure 6.10: Effect of quantisation on the performance of MCDDs.
6.6 Comparison Between IIR and FIR Demultiplexers

Before the simulation results of the carrier and bit timing recovery in the MCDD are presented, it is interesting to make a comparison between the proposed IIR and FIR demultiplexers. The overall comparisons are as follow:

1. The IIR demultiplexer has 76 real multipliers, while the FIR demultiplexer has 356 real multipliers. This is a saving of 78.7% in the number of multipliers.

2. The IIR demultiplexer has 108 real adders, while the FIR demultiplexer has 500 real adders. This is also a saving of 78.4% in the number of adders.

3. The demultiplexing delay in the IIR demultiplexer is $9T$, while the delay in the FIR demultiplexer is $16T$. This is equivalent to 43.5% faster demultiplexing.

4. To demultiplex 8 channels, the channels must be sampled at $32R$ in the FIR demultiplexer, while the sampling frequency in the IIR DMUX is exactly half this value, i.e. $16R$. The advantages of halving the sampling rate, from $32R$ in (6.3) to $16R$ in (6.4), on the multiplexer are:
   (a) The multiplexer operates at half the speed.
   (b) The anti-imaging anti-aliasing low-pass filter (LPF) required in the TDM-FDM convertor has approximately half the length $N$. This can be easily verified from [Bel80]

\[
N \approx \frac{2}{3} \log \left( \frac{1}{10\delta_1\delta_2} \right) \frac{f_s}{\Delta f}
\]  

(6.9)

or

\[
N \propto f_s
\]  

(6.10)

Therefore, halving the sampling rate means that the length of the prototype filter is approximately halved.

5. By careful investigation of the spectra in Figure 6.2 it can be seen that half the spectrum from $f = f_s / 2$ to $f = f_s$ has been left empty in the case of oversampling the FDM signal. In general with an oversampling factor $J$, the spectrum from $f = f_s / J$ to $f = f_s$ remains unusable. If a channel is placed in this portion of the spectrum, it will be distorted by the transition bands of the demultiplexing filter and can not be recovered.
6. Since the IIR demultiplexer operates at half the sampling rate at which the FIR demultiplexer is operating, the IIR demultiplexer is more attractive for ASIC implementation.

7. The IIR demultiplexer has a superior BER performance compared to the FIR demultiplexer when the prototype filter coefficients have a finite word-length, and is hence more suitable for ASIC implementation.

All the research results 1–7 presented are strong evidence suggesting that the FIR demultiplexer is not as advantageous for implementation in the 8-channel MCDD. Therefore in the remaining of this chapter, only the proposed IIR demultiplexer will be used.

Before presenting the simulation results of synchronisation of the demultiplexed channels in a coherent demodulator, some aspects of the phase ambiguity in coherent receivers are discussed next.

6.7 Phase Ambiguity

A problem in coherent demodulators is that the oscillator used in the carrier recovery circuits may not be at the same phase as the phase of the transmitter oscillator. The carrier recovery loop not only attempts to correct the phase error in the received data, but also the output of the circuit may have an additional phase offset. If this phase offset is not corrected, there will be decision errors. The phase offset is ambiguous and must be detected and corrected. An example of the phase ambiguity is shown in Figure 6.11. With zero phase error and a pseudo-random binary source seed value set to 111111111, the phase offset is $\pi$ radian. When the actual error of 0.1 radian was introduced in the transmission channel model, the carrier phase synchronisation loop tracked an error of ($-\pi + 0.1$) radian. Therefore, a mechanism is required in the receiver to detect and remove the phase ambiguity. The phase ambiguity is also influenced by the choice of the source seed value. With the seed changed to 222222222, the phase ambiguity increased to $-3\pi / 2 = \pi / 2$ radian.

A simple method of resolving the phase ambiguity in simulations is to run the simulation once, find out how much the phase ambiguity is, and then rotate the phase of the data before making a decision. In the remaining of this section, a more practical approach to resolving the phase ambiguity is described.

The phase ambiguity may be resolved by employing a unique word (UW). The UW is a sequence of ones and zeros selected to exhibit good correlation properties.
To resolve the phase ambiguity by using UWs, the transmission must be in burst-mode. A UW is sent at the beginning of each data frame. UWs vary in length and may be as short as 10 bits or as long as 24 [Feh83]. A suitable UW is 011110001001 [Ahm92] whose length $N$ is 12 bits. Table 6.1 shows the possible received sequence at different phase ambiguities. The reason for underlining some of the bits will be made clear in sub-section 6.7.2.

Table 6.1: Phase ambiguity in QPSK demodulator

<table>
<thead>
<tr>
<th>Phase Ambiguity</th>
<th>Received UW Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>011110001001 100001110110</td>
</tr>
<tr>
<td>$\pi/2$</td>
<td>000111101100 111000010011</td>
</tr>
<tr>
<td>$\pi$</td>
<td>100001110110 011110001001</td>
</tr>
<tr>
<td>$-\pi/2$</td>
<td>111000010011 000111101100</td>
</tr>
</tbody>
</table>
6.7 Phase Ambiguity

The receiver identifies the start of each frame by detecting the UW. The UW detector establishes correlation between the received bits with the replica of the UW stored in the correlator. Only the received sequences which produce a correlation peak greater than a predefined threshold, $E$, are retained as UW. The performance of the UW detector is measured by the probability of the number of errors to be larger than the threshold, i.e. a miss, and the probability of falsely identifying the UW, i.e. a false alarm. The probabilities of miss $P_m$ and false alarm $P_{fa}$ are [Feh83]

$$P_m = \sum_{l=E+1}^{N} \frac{N!}{l!(N-l)!} p^l (1-p)^{N-l}$$

(6.11)

and

$$P_{fa} = \frac{1}{2^N} \left[ \sum_{l=0}^{E} \frac{N!}{l!(N-l)!} \right]$$

(6.12)

The variations in $P_m$ with respect to BER are shown in Figure 6.12(a). For the same number of errors in UW correlation threshold, as the UW length decreases, $P_m$ decreases. Any decrease in BER results in a decrease in $P_m$. Furthermore, as $E$ increases, $P_m$ decreases.

The variations in the probability of a false alarm with respect to the UW length are shown in Figure 6.12(b). $P_{fa}$ increases as $N$ decreases and decreases as $E$ decreases. Therefore there is a compromise between decreasing $P_m$ and decreasing $P_{fa}$.

To verify the probability of false lock, a simulation was run with $E = 0$ and $N = 24$. The theoretical $P_{fa}$ is $1 / 2^{24}$ or $P_{fa} = 0.596 \times 10^{-7}$. The simulation result was $0.3598 \times 10^{-7}$, close to the theoretical value.

For BER counting purposes the UW must be removed from the data stream before the count starts.

A method to generate UWs and detect them in COSSAP is by using discrete blocks available in the model libraries [Syn94]. However, this can be time-consuming and error-prone as the size of the UW changes. A more efficient method is to create models in one of the supported programming languages. For this purpose, a model to generate UWs and a model to detect and remove the UWs have been developed in the "C" programming language.

6.7.1 Frame Synchronisation

Assume the system is tracking the input errors and there is a sudden loss of lock due to, for example, excessive noise power over signal power. The only way that
the transmitter can find out that the satellite has lost tracking is by a signal from the satellite back to the station. In that case, the transmitter starts sending a sequence of frame synchronisation (SYNC) data. The frame synchronisation data pattern must show different phase variations to assist the satellite system to acquire the lost signal. For example, for BPSK the sequence of

\[ 0, 1, 0, 1, 0, 1, \ldots \]  

(6.13)
can be a good sequence [Wil93]. The phase angle changes by $\pi$ radian from one symbol to the other. For QPSK a suitable frame synchronisation pattern is

$$00, 10, 01, 11, 00, 01, 10, 11, 00, 10, 01, 11 \cdots$$

which shows $\pi/2, \pi, \pi/2, \pi$, $-\pi/2, \pi, -\pi/2, \pi, \pi/2, \pi, -\pi/2 \cdots$ radian phase variation from one symbol to the other.

### 6.7.2 Bit Stuffing

A pitfall in detecting the UW is that the transmitter may just happen to send the same data pattern as the UW. To prevent this coincidence the simple method of bit stuffing is used, in which some bits are inserted in the data. The process of bit stuffing increases the sampling rate by a factor $n/F$ where $n$ is the number of bits stuffed in a frame of length $F$. The stuffed bits are merely overhead and decrease the information-handling capacity of the transmission link. These bits must be removed at the receiver. Depending on one application to the other, one may choose different bit stuffing approaches. Referring to Table 6.1, the underlined sequence 011110 is common to all phase ambiguity possibilities. If the user data happens to have the above pattern, a "1" is stuffed after 011110. On the other hand if the sequence 011110 is received, a test is performed on the next three bits. If all three bits are one, it means that a "1" has been stuffed after 011110 and must be removed.

#### 6.7.2.1 An Improved Bit Stuffing Technique

Although the above approach is simple to implement, it has the disadvantage that the bit pattern is too short and the probability of its occurrence is high. This means that bit stuffing must be performed, on average, once every $2^6 = 64$ bits. A better method is to consider the whole UW as the desired pattern. If the data happens to have bit-by-bit correlation with one of the forms of the UW in Table 6.1, the complement of the UW, $\overline{UW}$, is stuffed before outputting the data. By this approach, the probability of data pattern to be exactly similar to one of the patterns in Table 6.1 is $1/2^{24}$. This is one occurrence in approximately every 16.8 million samples.

### 6.7.3 Detection of the Frame Overheads

As mentioned earlier, the incoming stream of data is multiplexed with a frame synchronisation sequence, UWs, user data, and possibly the stuffed UWs. How can
a coherent receiver distinguish between these UWs in order to resolve the ambiguity? A heuristic method has been shown in Table 6.2.

Table 6.2: Possibilities of detecting UWs by the receiver

<table>
<thead>
<tr>
<th>Case</th>
<th>Bits 0 – 23</th>
<th>Bits 24 – 47</th>
<th>Bits 48 – 71</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>UW not found</td>
<td>UW not found</td>
<td>UW not found</td>
</tr>
<tr>
<td>1</td>
<td>UW not found</td>
<td>UW not found</td>
<td>UW found (F)(S)</td>
</tr>
<tr>
<td>2</td>
<td>UW not found</td>
<td>UW found (F)</td>
<td>UW not found</td>
</tr>
<tr>
<td>3</td>
<td>UW not found</td>
<td>UW found (S)</td>
<td>UW found (D)</td>
</tr>
<tr>
<td>4</td>
<td>UW found (F)</td>
<td>UW not found</td>
<td>UW not found</td>
</tr>
<tr>
<td>5</td>
<td>UW found (F)</td>
<td>UW not found</td>
<td>UW found (S)</td>
</tr>
<tr>
<td>6</td>
<td>UW found (S)</td>
<td>UW found (D)</td>
<td>UW not found</td>
</tr>
<tr>
<td>7</td>
<td>UW found (S)</td>
<td>UW found (D)</td>
<td>UW found (F)</td>
</tr>
</tbody>
</table>

Referring to Table 6.2, the receiver performs a test on every 72 bits (this is the size of 3 UWs). The decisions made are shown in brackets, where

(F) is the the start-of-frame (SOF) UW.

(S) is the stuffed UW.

(D) is the user data whose all bits excluding $\gamma$ bits are similar to the SOF UW.

If a UW is not found, the sequence is either user data, or frame synchronisation sequence. In case 1, the receiver does not know whether bits 48 – 71 are the SOF UW or the stuffed UW. To resolve this ambiguity, 48 bits are read from the input port and a test is made on them. The result of the test will be one of the cases 4, 5, 6, or 7. In case 5, as bits 48-71 are the stuffed UW, there is a confidence that the next 24 bits are data and are sent to the output port without any further test.

To detect the frame synchronisation sequence, a correlation test is made on the decoded bits. If the correlation spike is higher than a predefined value, the
synchronisation sequence is found and will be discarded. Otherwise, it will be
deduced that these bits are the user data and will be processed for BER counting.

6.7.3.1 Simulation Results of Detecting the Frame Overheads

The simulation models were tested in a modem which consisted of a transmitter, a
transmission channel and a receiver with synchronisation loops. At the start of the
simulation, sequences of SYNC as given in (6.14) were transmitted. After trans­
mitting the SYNC sequence, frames of 1024 bits were transmitted. The overheads
at the beginning of each frame are 24 bits UW and 24 bits SYNC which occupy
4.6875% of the frame length. Different values of WGN were added to the channel. The simulation results are shown in Table 6.3. It was found that, with an
allowed error $E = 2$, the models could only work when $E_b/N_0$ was in the range of
4 – 6 dB. To overcome this problem, the complement of the UW and SYNC, were
added after the UW and the SYNC sequence, respectively. Results show that with
$1 \leq E_b/N_0 \leq 9$ dB the models have successfully detected, and corrected the phase
ambiguity.

Table 6.3: Simulation results of detecting the frame overhead

<table>
<thead>
<tr>
<th>$E_b/N_0$ (dB)</th>
<th>$E_{{UW,SYNC,DATA}}$</th>
<th>$E_{{UW,UW,SYNC,SYNC,DATA}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>–</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>–</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>–</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>–</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>–</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>–</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>–</td>
<td>2</td>
</tr>
</tbody>
</table>

6.8 Simulation Results of Doppler Error Synchronisation

In Chapter 3, different Doppler error synchronisation algorithms were analysed.
The Alberty and Hespelt (A&H) algorithm was identified as a suitable candidate
for comparison against the proposed rotational frequency error detector (FED) algorithm. In Section 6.4 the simulation results of the demultiplexing of channel number 4 were presented. Furthermore, in Section 6.7 the potential problem of phase ambiguity arising in coherent demodulators was dealt with. An improved technique for resolving the phase ambiguity was presented. With the given background, the Doppler shift can be added to the channel model.

![Figure 6.13: Simulation results of the Doppler error synchronisation by the Alberty and Hespelt algorithm.](image)

Figure 6.13 shows the acquisition and tracking performance of the A&H algorithm. Although the analysis in Chapter 3 showed that the algorithm is self-noise free in the absence of AWGN, Gaussian noise causes jitter during tracking. To overcome this problem, the loop must be disconnected when the frequency error has been acquired. Switching the input of the loop filter from the FED and connecting it to a zero output source at the optimum time is a side problem in the Doppler error synchronisation. This means that additional circuitry must continuously monitor the smoothed output of the error detector. An example of the implementation of the switching mechanism can be found in [Ahm92]. The simulation result of switching is shown in Figure 6.13. It is seen that the loop has acquired the error after about 300 symbols. With the Iridium satellites designed to transmit at 4.8 kbps [Iri93], assuming the transmission will be in burst mode with a frame length of 2400 bits, 300 symbols acquisition time means that only 1800 — \( N \) bits of data can be transmitted, where \( N \) is the UW length and the length of other frame overheads. To increase
this capacity, the acquisition time must be decreased. With a smaller acquisition
time, less SYNC sequence is transmitted. In the remaining of this section, the
performance of the proposed algorithm is presented.

![Graph](image)

(a) Acquisition and tracking performance

![Graph](image)

(b) BER performance

Figure 6.14: Simulation results of the Doppler error synchronisation by
the proposed algorithm.

In Figure 6.14(a), the simulation results of the Doppler error synchronisation
with the proposed FED is shown. The channel conditions are the same as those
used in the simulation of the A&H algorithm. There are two important features in
6.9 Joint Symbol Timing and Carrier Phase Synchronisation

the results:

1. The acquisition time is about 55 symbols as compared to about 300 symbols for the case of the A&H algorithm. This is a saving of 245 symbols period in acquisition time. This superior performance is due to the fact that ideally the acquisition time is only 2 samples period when the channel SNR is infinity. In the presence of AWGN, an averaging of the output must be performed. The above acquisition time of about 55 symbols is due to the averaging filter. Rather than using a fixed averager, an averaging filter whose $k$-th output sample is given by

$$y(k) = \frac{1}{K} \sum_{i=1}^{K} x(i), \quad k = 0, 1, \ldots$$

has been used.

2. The tracking performance is smooth enough that there is no need for a switching mechanism. Therefore, the overall hardware complexity of the proposed synchronisation algorithm is low. Furthermore, the proposed FED does not require any FIR filters, while the A&H algorithm requires FIR filters. This makes the proposed algorithm to be extremely computationally efficient compared to the A&H algorithm. Chapter 3 provides a detailed discussion on the implementation of the above algorithms.

The BER performance of the proposed FED in a Gaussian channel is shown in Figure 6.14(b). The results agree with the theoretical results of (6.6). For BER counting purposes, a second-order phase-locked loop (PLL) has been used before making a decision on the received data. With second-order PLLs very small amounts of frequency error, which has not been corrected by the frequency synchronisation loop, is reduced further. The performance of PLLs plus the design of the loop filter can be found in Chapter 5. The simulation results presented in this section were obtained by using the IIR demultiplexer.

6.9 Joint Symbol Timing and Carrier Phase Synchronisation

In Chapters 4 and 5 different symbol timing and carrier phase error synchronisers were surveyed. In presenting the simulation results of timing synchronisers in Chapter 4, it was assumed that there was no carrier phase error. Furthermore, in
Joint Symbol Timing and Carrier Phase Synchronisation

Chapter 5 it was assumed that there was no symbol timing error. In practice, the received signal is affected by both timing and phase errors. It is easy to verify that the performance of the timing (phase) synchronisers degrades in the presence of phase (timing) error. When the phase (timing) error becomes large, the timing (phase) synchronisers fail to operate. The only way the above algorithms can coexist and the existing hardware complexity will not increase, is by a joint operation. In joint synchronisation, the operation of each algorithm influences the operation of the other; the phase rotated signal is fed to the clock synchroniser and the output of the timing error correction device (interpolator) is fed to the phase synchroniser. Although each synchroniser has a fast acquisition time when used on its own, by joint synchronisation the overall acquisition time increases.

At the end of Chapter 4, the Gardner algorithm and proposed optimised mM&M algorithm were selected for further investigation in this chapter. The question which arises is “If the joint operation is possible, which timing synchroniser is more suitable to use? the Gardner algorithm which does not need to be implemented in a joint configuration to detect the timing error in the presence of the phase error, or the proposed optimised timing error detector which must be implemented in a joint configuration when there is a phase error?” The answer is in the tracking performance.

Looking carefully at the acquisition results shown in Chapter 4, it is evident that each result was obtained by using different loop gain factors $\beta$. In order to ensure a correct comparison is made between clock synchronisers, the gain of the algorithms must be normalised to unity. Referring to the S-curve of any algorithm, one can determine the gain by dividing the mean of the detector output by the normalised timing error. The output of the error detector is then scaled by the inverse of this gain. The acquisition performance for the Gardner algorithm and the proposed optimised mM&M algorithm are shown in Figure 6.15. In both cases the gain has been normalised to unity and the loop gain factor $\beta$ is also unity. It is quite clear that under the same condition, the Gardner algorithm has a large tracking jitter, which results in a poor noise performance. This tracking jitter was already identified from the curve of the noise performance of this synchroniser in Chapter 4. The tracking jitter can be decreased by decreasing the loop gain factor $\beta$. However, this in turn slows the acquisition time. The proposed optimised algorithm has no tracking jitter.

4The discussion here does not include the Gardner timing error detector which operates independent of carrier phase error.

5See Chapters 4 and 5.
6.9 Joint Symbol Timing and Carrier Phase Synchronisation

Figure 6.15: Acquisition and tracking performance of the Gardner algorithm (GA) and the optimised mM&M algorithm.

due to its optimum performance. For an analysis of this algorithm, see Chapter 4. From these evidences, it is clear that only the proposed mM&M algorithm is suitable for further consideration in the optimised IIR MCDD.

In the next sub-section, the choice of a suitable phase synchroniser for the optimised IIR MCDD is presented.
6.9 Joint Symbol Timing and Carrier Phase Synchronisation

6.9.1 Simulation Results of Carrier Phase Recovery

In Chapter 5, the noise performance of the algorithms surveyed was presented. The results showed that for QPSK, the proposed optimised decision-directed ML FF algorithm had the lowest noise variance at medium to high signal-to-noise ratio. Under the same channel conditions, the Costas loop had the next best noise performance. Therefore, these two algorithms were selected for further investigation. In this section, further developments on the final selection of the phase synchroniser are reported.

Both of the above algorithms have been simulated for their acquisition time and tracking performance. The results are shown in Figures 6.16 and 6.17 for two different channel SNRs. The Costas loop has clearly a slower acquisition response. Costas loop has a feedback configuration while the proposed algorithm has a feedforward configuration. The analysis in Chapter 5 showed that feedforward algorithms have a fast acquisition time, which is ideally one symbol period. To achieve such a fast acquisition, ideal conditions are assumed, i.e. the Doppler shift and the clock timing have been perfectly synchronised and the precision of the filter coefficients in the modem and the transmitted signal is infinite. Under imperfect conditions, the acquisition time must be increased to allow for a smaller tracking jitter.

![Figure 6.16: Acquisition and tracking performance of the Costas loop and the proposed PED algorithm at $E_b / N_0 = 5$ dB.](image)

Chapter 6 An optimised 8-channel IIR MCDD
6.9 Joint Symbol Timing and Carrier Phase Synchronisation

The acquisition time of the Costas loop is about 250 symbols. This slow acquisition is due to the fact that in this algorithm the error is found by summing the phase error estimates \( \hat{\theta} \), which are initially close to zero and take a number of symbols to approach the actual error. Unlike the proposed optimised algorithm, in which the phase error is directly measured by performing an arctangent operation on the sampled matched filter output, in Costas loop the error is estimated by calculating the phase difference between the constellation of the received signal and the ideal constellation points. For QPSK, the constellation points are ideally at \( \pm \frac{1}{\sqrt{2}} \). In simulations, the constellation points are at \( \pm \frac{1}{\sqrt{2}} + \epsilon \), where \( \epsilon \) is the error due to inaccuracies in arithmetic operations. The constellation points are on average at \( \pm \frac{1}{\sqrt{2}} \). But it is not desirable to read a large number of symbols, and perform averaging in order to find an estimate very close to \( \pm \frac{1}{\sqrt{2}} \). Therefore with the Costas loop, the phase error must be minimised recursively. This is the reason that the acquisition time of the Costas loop in Figures 6.16 and 6.17 is slow. The acquisition time can be decreased by increasing the bandwidth of the loop. However, by increasing the bandwidth, the tracking jitter increases and may drive the loop into instability. A suitable loop bandwidth can be found by simulation. In performing the simulations in Figures 6.16 and 6.17 the normalised bandwidth \( B_L T \) was 0.01.

From the above description and the results, it is clear that the proposed optimised DD ML algorithm is the most suitable candidate for phase synchronisation in the IIR
MCDD. Fast acquisition, low tracking jitter, good noise performance are the major features of this algorithm. Furthermore, the algorithmic complexity is reasonably low.

The noise performance of a single-channel modem developed with a joint symbol timing and carrier phase recovery is shown in Figure 6.18. The simulation results have been obtained with fixed timing and phase errors of 0.25$T$ and 0.3 radian, respectively. The high BER at low signal-to-noise ratios is due to the fact that both algorithms use the receiver’s decision to detect the error $\Psi$, and at low SNR the decision errors are higher. As the SNR increases, the receiver’s decision errors decreases.

![Figure 6.18: BER performance of the joint symbol timing and carrier phase synchronisation.](image)

A question which arises here is “why 0.25$T$ symbol error was used in simulations?”. Assuming each channel transmits 2 samples per symbol, the maximum timing error which can be introduced is 0.5$T$ which can be introduced by deleting one sample from the beginning of the transmitted signal. Although the detection of this timing error in the absence of phase error is possible, in joint synchronisation this amount of error proved to be too high to detect and correct in the presence of Gaussian noise. Expecting a synchroniser to detect 0.5$T$ error is like expecting a QPSK phase synchroniser to detect $\pi / 4$ radian phase error. Such high errors cause unrecoverable decision errors. The next option is to increase the number of transmitted samples per symbol to 4, and to introduce 0.25$T$ symbol error. In the
next section, it will be shown that such a timing error can be introduced in an FDM signal even when each channel is transmitting at 2 samples per symbol.

6.10 Performance of the Optimised 8-Channel IIR MCDD

In this section the performance of the optimised MCDD developed is presented. There are 8 FDM channels to be demultiplexed. The demultiplexing scheme is based on the proposed IIR approach. The carrier and bit timing recovery techniques are based on the proposed synchronisation algorithms. Before presenting the results, it is useful to discuss how the timing error can be introduced in the case of multi-channel FDM signal.

Figure 6.19: Performance of the optimised 8-channel recursive (IIR) MCDD at $E_b/N_0 = 6$ dB.

Chapter 6 An optimised 8-channel IIR MCDD
6.10 Performance of the Optimised 8-Channel IIR MCDD

Referring to Figure 6.8, the normalised sampling rate at point "G" is 4, and there are two channels which must be demultiplexed. In order to allow for $0.25T$ timing error, a slight modification has to be made to the scheme proposed; after demultiplexing channel 4, no decimation is performed. Therefore, the sampling rate remains at 4 samples per symbol at point “N”. By deleting 4 samples from the beginning of the FDM signal before demultiplexing, each demultiplexed channel at points “I” to “P” in Figure 6.4 will have a timing error of 1 sample per symbol which consists of four samples. As a general rule, with $N$ channels, $N/2$ samples must be deleted from the beginning of the FDM signal so that each channel will have a timing error of $0.25T$.

The acquisition and tracking performance of the synchronisation algorithms, and the overall noise performance have been shown in Figure 6.19. The phase and timing errors can not be acquired unless the frequency error has been acquired. The acquisition time of the frequency error detector is fast; for a frequency error of $0.2\tau$, it takes about 50 symbols to acquire the error. From Figures 6.19(b) and 6.19(c), it can be seen that the timing and phase synchronisation loops have acquired the error soon after the frequency error has been acquired.

The BER results in the range from $E_b/N_o = 3.5 \text{ dB}$ to $E_b/N_o = 8 \text{ dB}$ have been shown in Figure 6.19(d). With simulations performed in the range of 0 to 8 dB, in steps of 0.5 dB, the MCDD could not synchronise the errors when the SNR was set to less than 3.5 dB. The failure in acquisition at low SNRs was also reported in [Ahm92] in which the carrier and bit timing error recovery algorithms were used in a single-channel OQPSK modem for application in the Inmarsat-M satellites. In order to achieve synchronisation at $E_b/N_o < 3.5 \text{ dB}$, the loop gain factor $\beta$ in the symbol timing recovery loop must be decreased, which in turn increases the acquisition time. With the configuration of the MCDD presented in this chapter, a decrease in $\beta$ will be necessary if error correction coding is used; coding will reduce the channel $E_b/N_o$ further by $10\log_{10}(R_c)$ where $R_c$ is the code rate.

In Figure 6.19 at any given SNR, 500 errors have been counted to simulate the BER. Simulations at higher SNRs, in general, take a considerable time to finish. This problem becomes worse when several channels are present. The degradation of the BER performance from the theoretical results is about 0.23 dB at BER=$10^{-3}$. Although 8 channels are present in the MCDD, such a low degradation in noise performance is due to the following reasons:

- Low quantisation noise in the IIR demultiplexer
6.11 Summary

- Near to optimum noise performance of the frequency synchroniser.

The main cause of the above degradation is due to the carrier and bit timing error recovery loops. This is not because these loops operate in a joint configuration in the MCDD developed. Such a problem, can also be seen when the algorithms operate in cascade [Ahm92].

The above MCDD uses coherent demodulation, in which there is a phase ambiguity introduced by the VCOs used for the carrier error correction. In simulations of the MCDD developed, a phase ambiguity of \( \pi \) radian was detected and corrected. This was achieved by using 24 bit unique words, and the complement of the unique words. Details of the implementation of the unique word detection can be found in Section 6.7.

6.11 Summary

In this Chapter, the final comparisons between different demultiplexing and synchronisation algorithms were made in order to find solutions that are superior to other algorithms. It was shown that, with the FDM channel specifications, the proposed IIR demultiplexer offered the following advantages over the FIR demultiplexer:

- 78.7% less real multipliers;
- 78.4% less real adders;
- 43.5% faster demultiplexing;
- 50% more spectral utilisation;
- 50% less coefficients in the anti-imaging anti-aliasing filter in the FDM multiplexer;
- better BER performance with quantised filter coefficients;
- more attractive for ASIC implementation.

The problem of phase ambiguity was addressed next. Unique words were used to resolve the phase ambiguity. To ensure that the unique words were accurately distinguished from the frame synchronisation sequence and the data, a detailed heuristic technique was presented.
To correct the Doppler shift, two candidate algorithms were considered. It was shown that the proposed algorithm has the following advantages over the Alberty and Hespelt dual filter discriminator:

- a faster acquisition time;
- no need for switching between the frequency synchronisation loop and the phase-locked loop. Therefore, the hardware complexity is low.

The symbol timing synchronisers were then considered. It was shown that the proposed optimised mM&M algorithm has the following superior features compared with the Gardner algorithm:

- An operation at half the sampling speed at which the Gardner algorithm operates;
- No tracking jitter.

The selection of the most suitable carrier phase recovery algorithm was then presented. It was shown that, under the same conditions, the proposed decision-directed maximum-likelihood feedforward phase error detection algorithm has the following advantages over the Costas loop:

- Faster acquisition of the error;
- Lower tracking jitter.

Finally, the performance of the MCDD developed was presented. It was shown that the acquisition time of all the synchronisation algorithms was fast. The degradation in the BER performance from the theoretical results was low. Fast acquisition of the signal is necessary for applications such as those in the Iridium or Inmarsat-P satellites, while a low degradation of the noise performance from the ideal response is a desirable feature in any application.
Chapter 7
Conclusions and Further Research

Global mobile communications using hand-held terminals will materialise with the advent of the proposed on-board processing low-earth orbit (LEO) satellites in the Iridium constellation and the Inmarsat-P intermediate circular orbit (ICO) satellites. The optimum design of the multicarrier demultiplexer demodulators (MCDDs) for these satellites is crucial to acquire the signal as fast as possible with the minimum design complexity.

In order to identify the most suitable design algorithms for MCDDs, an extensive survey on demultiplexing and synchronisation algorithms was carried out. To evaluate the performance of each algorithm, it was decided to work with 8 frequency division multiplexed (FDM) QPSK channels. The choice of 8 channels was purely for documentation purposes and the computer simulation time. Logically our first task was focused on the demultiplexing of the received FDM signals. It was found that the FIR binary tree structure was a suitable candidate because of its simple implementation. A complete tree demultiplexer was developed. My next task was to make an improvement to the demultiplexing approach. From our studies, it was clear that in the published literature only FIR demultiplexing techniques had been considered for on-board processing satellites. Therefore, we concentrated on exploring the feasibility of using an IIR demultiplexer. The nonlinear phase response was a disadvantage in using the conventional design methods which are widely available in textbooks. By further literature survey, it was found that the Ansari-Liu algorithm was attractive for our application. By this algorithm, prototype filters with approx-
Innately linear phase response can be designed. As the software for this complex algorithm was not available, a complete stand-alone package was developed which included a minimisation algorithm to optimise the designed filters. The designed IIR filters were used in a new demultiplexing scheme which took advantage of the frequency response of the prototype filters. Recovering the individual channels is important. However, the demultiplexing implications are more important. By comparisons with an FIR demultiplexer, it was shown that the self-proposed IIR approach had a considerably lower computational complexity in terms of arithmetic operations, and a lower demultiplexing delay. Such attractive features are due to the fact that a lower number of coefficients is required in designing the IIR filters to meet a given spectral specification. Further advantages in using the self-proposed demultiplexing scheme were reported. It was shown that, with the same channel specifications, the IIR scheme offered double the spectral utilisation which can be achieved by using the FIR demultiplexer. Further simulations demonstrated that the IIR demultiplexer had a superior noise performance when the filter coefficients were quantised. Since the IIR demultiplexer operates at half the sampling rate at which the FIR demultiplexer is operating, the IIR demultiplexer is more attractive for ASIC implementation. All these features showed that the self-proposed IIR demultiplexer was a suitable candidate for consideration in future on-board processing applications.

Covering the synchronisation aspects of the recovered channels in coherent demodulators was our next task. After a detailed literature survey, the Alberty and Hespelt (A&H) dual filter discriminator (DFD) was selected among other algorithms as suitable for further consideration. The presented analysis showed that this algorithm is always self-noise free at high SNR. However, the simulation results showed that the algorithm had a slow acquisition time, and the synchroniser had to be disconnected as soon as the frequency error had been acquired. The problem of when to disconnect the synchroniser meant an extra hardware complexity. Research to decrease the complexity and the acquisition time was therefore necessary. The result of our research in this area was a new heuristic frequency error detector. Compared to other algorithms, the self-proposed algorithm has a very low algorithmic complexity and a fast acquisition time. It was shown that at $E_b/N_0 = 6$ dB, the algorithm acquired the frequency error in about 50 symbols, which is a major improvement in the acquisition time which can be achieved by the Alberty and Hespelt dual filter discriminator. Furthermore, the computational and the implementation complexi-
ties of the self-proposed algorithm are considerably lower than that of the Alberty and Hespelt algorithm. The location of the frequency error detector is important. The self-proposed FED operates before the FDM demultiplexer, while the Alberty and Hespelt algorithm must be placed after demultiplexing. Therefore unlike the Alberty and Hespelt algorithm, the filtering operation due to demultiplexing has no influence on the self-proposed FED. From Bellanger’s FIR filter design formula\(^1\) it is evident that if the DFD is implemented before demultiplexing, a large number of filter coefficients is required which makes the frequency error detector inefficient.

The next part of our research was related to finding the best solution for symbol timing recovery. Many algorithms were analysed. Particular attention was paid to the Gardner algorithm and the modified Mueller and Müller (mM&M) algorithm. The Gardner algorithm is attractive to use due to its operation being independent of the phase error. However, it requires two samples per symbol to detect the error. On the other hand, the mM&M algorithm requires only one sample per symbol, but its operation is affected by the phase error. It was shown that the problem of the phase error in the mM&M algorithm can be solved by a joint synchronisation technique. The feature of this algorithm which prohibits its use in practical modems is that the algorithm shows symbol slips during tracking at high loop bandwidth. Symbol slips cause many unrecoverable errors. It was decided to explore ways to prevent the symbol slips by making the synchroniser self-noise free. An analysis was presented by which the algorithm was optimised. The result was that, even at high loop bandwidth, there are no symbol slips in the optimised algorithm during tracking. The optimised algorithm was then compared against the Gardner algorithm under the same loop bandwidth. It was shown that at high SNR there was no tracking jitter in the algorithm developed, while there was considerable tracking jitter in the Gardner algorithm. The excessively high tracking jitter, gives rise to a degradation in the noise performance of the MCDD. Furthermore, the simulation results of the self-proposed optimised mM&M timing error detector showed that at high SNR the acquisition time for the case of the QPSK modulation scheme, which will be used in satellites in the proposed Iridium constellation, was about 15 symbols. The other attractive feature of the algorithm developed is that only one sample per symbol is required, while the Gardner algorithm requires two samples per symbol. Therefore, the speed of the operation of the self-proposed algorithm is twice that of the Gardner algorithm.

\(^1\)See Chapters 2 or 6.
For the final part of our research, the carrier phase synchronisation was considered in order to find a suitable technique which offered a low tracking jitter and a fast acquisition. A survey of many algorithms were presented. Under the light of the survey carried out, the decision-directed maximum-likelihood feedforward algorithm was optimised. On the basis of the noise performance, the new phase error detector and the Costas loop were selected among other algorithms for further investigation in the MCDD. By analysis it was shown that the acquisition time of the self-proposed algorithm was faster than that which can be achieved by the Costas loop. Simulation results showed that the tracking jitter of the new algorithm was less than that of the Costas loop.

In summary, the major achievements of our research which shows a coherent and systematic approach in the development of an optimised 8-channel MCDD are as follows:

- a new recursive (IIR) demultiplexer;
- a new timing error detector;
- a new frequency error detector;
- a new phase error detector.

The emphasis on the research performed was on fast acquisition, low algorithmic complexity and a good noise performance.

Some suggestions for the continuation of our research are as follows:

1. In this thesis the performance of the MCDD developed was studied and simulated in the presence of Gaussian noise. It is important to investigate the performance under other channel conditions such as fading.

2. For documentation purposes, an 8-channel MCDD was developed. As in the Iridium satellites each MCDD must demultiplex 128 channels [Iri93], the simulation models developed must be extended to obtain the results for the above number of channels. There is a potential problem when large number of channels are simulated in COSSAP Version 6.6. To overcome memory problems, it is recommended to write programs to develop a single model for the FDM multiplexer and a single model for the IIR demultiplexer.

3. The operation of the optimised mM&M algorithm proposed depends on how well the phase error has been corrected. It would be useful to study the
possibility of extending this algorithm to make its operation independent of phase error. With such an extension, the BER performance would improve; the tracking jitter of the phase synchroniser would then have no influence on the performance of the timing synchroniser.
Appendix A

Background to Norms in the Ansari-Liu Algorithm

Norm of $x$, written as $||x||$, is a real number satisfying the following conditions:

1. $||x|| > 0$, unless the elements of $x$ are zero.

2. For any scaler $\lambda$, and any vector $||x||$

   $$||\lambda x|| = |\lambda| \cdot ||x||$$  \hspace{1cm} (A. 1)

3. For any vectors $x$ and $y$

   $$||x + y|| = ||x|| + ||y||$$  \hspace{1cm} (A. 2)

The most common norms for a continuous function $f$ on a closed interval $[a, b]$ are

1. Maximum norm:

   $$||f||_{\infty} = \max |f(x)|, \quad x \in [a, b]$$  \hspace{1cm} (A. 3)

2. Weighted Euclidean norm:

   $$||f||_{2,w} = \left( \int_a^b |f(x)|^2 w(x)dx \right)^{\frac{1}{2}}, \quad w(x) > 0$$  \hspace{1cm} (A. 4)

where $w(x)$ is the weighting function. The weighted Euclidean norms belong to a family of norms called the $l_p$ norms:

$$||f||_p = \left( \int_a^b |f(x)|^p w(x)dx \right)^{\frac{1}{p}}, \quad p \geq 1$$  \hspace{1cm} (A. 5)
or in discrete form

\[ \|f\|_p = \sqrt{w(a)|f(a)|^p + w(a+1)|f(a+1)|^p + \cdots + w(b-1)|f(b-1)|^p + w(b)|f(b)|^p} \]  
(A.6)

or

\[ \|f\|_p = \left[ \sum_{i=a}^{b} |f(x)|^{2p} \right]^{\frac{1}{2p}} \]  
(A.7)

Rather than using the above form, it is more convenient to use

\[ [\|f\|_p]^{2p} = \sum_{i=a}^{b} |f(x)|^{2p} \]  
(A.8)

The above norm has been used to design filters in the Ansari-Liu algorithm.
Appendix B

IIR Polyphase Filters with a Nonlinear Phase Response

The $q^{th}$ weighted norm is

$$e_2^q(q) = \sum_{k=1}^{K} W_k \left| H_k - |H_k^d| \right|^{2q} \quad (B.1)$$

Take the partial differentiation with respect to $\beta_p$

$$\frac{\partial e_2^q(q)}{\partial \beta_p} = \sum_{k=1}^{K} 2q \cdot W_k \left| H_k - |H_k^d| \right|^{2q-1} \cdot \frac{\partial |H_k|}{\partial \beta_p}$$

$$= \sum_{k=1}^{K} 2q \cdot W_k \left| H_k - |H_k^d| \right|^{2q-1} \cdot \frac{\partial |H_k|}{\partial \beta_p} \quad (B.2)$$

Expand $\frac{\partial |H_k|}{\partial \beta_p}$ in Equation (B.2)

$$\frac{\partial |H_k|}{\partial \beta_p} = \frac{\partial}{\partial \beta_p} \left[ \left( H_k \cdot H_k^* \right)^{1/2} - \left( H_k^d \cdot H_k^* \right)^{1/2} \right]$$

$$= \frac{1}{2\sqrt{H_k H_k^*}} \frac{\partial H_k}{\partial \beta_p} \cdot H_k^* + \frac{1}{2\sqrt{H_k H_k^*}} \frac{\partial H_k^d}{\partial \beta_p} \cdot H_k$$

$$= \Re \left[ \frac{1}{|H_k|} \cdot \frac{\partial H_k}{\partial \beta_p} \cdot H_k^* \right]$$

$$= \Re \left[ \frac{H_k}{H_k} \cdot \frac{\partial H_k}{\partial \beta_p} \right] \quad (B.3)$$

Substitute (2.49) in (B.3)

$$\frac{\partial |H_k|}{\partial \beta_p} = \Re \left[ \frac{|H_k|}{H_k} \cdot G_{pk} \cdot \frac{\partial \ln f_k(\beta_p)}{\partial \beta_p} \right] \quad (B.4)$$

Substitute (B.4) in (B.2)

$$\frac{\partial e_2^q(q)}{\partial \beta_p} = \sum_{k=1}^{K} 2q \cdot W_k \left| H_k - |H_k^d| \right|^{2q-1} \cdot \Re \left[ \frac{|H_k|}{H_k} \cdot G_{pk} \cdot \frac{\partial \ln f_k(\beta_p)}{\partial \beta_p} \right] \quad (B.5)$$

where $G_{pk}$ is given by (2.51). The design of IIR filters with a non-linear phase response is similar to that of the approximately linear phase design. However, the filter $H_0(z)$ is no longer constrained to be 1 to preserve the input samples.
Appendix C

General Features of All-Pass Filters

All-pass filters have a magnitude response which is unity for all frequencies:

\[ |H(\omega)| = 1, \quad \text{for all } \omega \quad (C.1) \]

The transfer function of an all-pass filter is of the form:

\[
H(z) = \frac{\sum_{k=0}^{N} a_k z^{-N-K}}{\sum_{k=0}^{N} a_k z^{-k}} = \frac{a_0 z^{-N} + a_1 z^{-N+1} + \cdots + a_N}{a_0 + a_1 z^{-1} + \cdots + N z^{-N}} = \frac{a_0 + a_1 z + \cdots + a_N z^N}{a_0 z^N + a_1 z^{N-1} + \cdots + a_N} \quad (C.2)
\]

All the coefficients \( a_k \) are real. The numerator and denominator coefficients are the same except that their order is reversed. To see that Equation (C.2) implies Equation (C.1), (C.2) is written as

\[
H(z) = \frac{z^{-N} \sum_{k=0}^{N} a_k z^k}{\sum_{k=0}^{N} N a_k z^{-k}} = z^{-N} \frac{D(z^{-1})}{D(z)} \quad (C.3)
\]

whose magnitude is unity. The filter is a pure phase shifter.
Appendix D

Formatting the Filters Designed by DFDP for FIR Demultiplexers

In COSSAP there are no facilities to design prototype filters. The package used in designing various filters is called Digital Filter Design Package (DFDP) and runs on IBM compatible PCs. The designed filter output file contains more information than that required. The file is first converted to a special COSSAP format by a developed utility called flt21og (FiLTer to LOG). To extract and reformat the filter coefficients the following information from the DFDP file are used

<table>
<thead>
<tr>
<th>Number of Entries</th>
<th>Name</th>
<th>Purpose and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ID</td>
<td>Design method flag:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 ⇒ IIR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 ⇒ KFIR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 ⇒ PMFIR</td>
</tr>
<tr>
<td>1</td>
<td>NF</td>
<td>Filter length variable.</td>
</tr>
<tr>
<td>NF</td>
<td>IX</td>
<td>Integer array containing a 16-bit representation of the impulse response coefficients This array is meaningless if the coefficients have not been quantised.</td>
</tr>
<tr>
<td>1</td>
<td>IPX</td>
<td>Power of two for scale factor of integer coefficients.</td>
</tr>
<tr>
<td>1</td>
<td>IQ</td>
<td>Quantisation flag:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 ⇒ no quantisation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 ⇒ quantised</td>
</tr>
<tr>
<td>1</td>
<td>NBITS</td>
<td>Number of bits in quantised coefficients.</td>
</tr>
<tr>
<td>1</td>
<td>FS</td>
<td>Sampling rate (KHz).</td>
</tr>
</tbody>
</table>

The utility program opens the filter coefficient file and from the second line the number of the filter coefficients is found. The program then moves down $NF + 4$ lines and starts formatting the next $NF$ values. The formatted file is then converted to a binary file by the COSSAP diterm (DIrect from TERMinal) program.
Appendix E

An Overview of the Features of COSSAP

COSSAP is a software tool for simulating the transmission system layer, which consists of the channel, modulators, demodulators, encoders, decoders, filters, etc. The indispensable components of this package are a configurator, a simulator, a chart viewer and a report generator. The simulation configurator is used to select a set of models of functional blocks from the model library and connect them in a desirable topology. Parameters of the various models are specified while the system is being configured or during the simulation.

The simulator produces results and the relevant statistics. The models required by the configurator and the simulator are stored in model library databases. Post processing is performed on the results to display them in a variety of formats. The report generator gives error messages and some statistics such as the number of samples processed throughout the simulation.

Some of the main features of this package are

- Arbitrary number of feedback loops.
- On-line and off-line documentation for library models which gives information such as input/output ports and data types, parameters and their data types, external data file formats, history of changes, etc.
- Support of hierarchical models which consist of primitive models or other hierarchical models.
- Support of xdbx symbolic debugger for debugging the primitive model the user is developing.
An Overview of the Features of COSSAP

- Predefined I/O macros to assist the user in developing new models.
- Various libraries of preprogrammed models of functional blocks.

Indeed some of the features mentioned above can also be found in other computer-aided analysis and design (CAAD) packages such as Signal Processing Workstation (SPW) [Bar91][Kur93]. However, there are other features which make COSSAP more suitable for multirate digital filtering. These features are the stream-driven nature of this package and the automatic scheduling.

COSSAP has a stream-driven nature in which after the simulation control is passed to a model, the model checks its input buffer(s). If there is at least the minimum required number of signal samples available at the input port(s), signal processing starts. Otherwise, the simulation control is passed to the next model. This is an advantage over time-driven CAAD tools in which the simulation control is passed from one model to the other only after an increment in time.

Automatic scheduling deals with how many signal elements are processed at any time the models become activated. This is important in multirate digital filtering, in which the input and output sampling rates of one or more models are different. The input sampling rate of models is set automatically by the package. In packages, such as SPW, which do not support the self-scheduling, the user must manually set the input and output rates.
References


References


References


References


Index

\(\pi/4\)-QPSK, 16

AFC loop, 68
BQC, 79
CDP, 81
DFD, 73
gardner, 69
Ansari and Liu algorithm, 42
all-pass filters, 44
design, 43
group delay, 45
phase response, 45
baseband switching, 5
bit stuffing, 174
improved technique, 174
BPSK, 12
CDMA, 7
COSSAP, 19
self-scheduling, 19
DFDP, 162
digital structures, 28
FIR, 28
IIR, 30
direct method DMUX, 9, 31
Doppler shift, 62
curve, 63
rate of change, 63
even stacking, 31
first-order section, 52
noise variance, 57
FDMA, 7, 155
rate, 156
Fletcher-Powell algorithm, 46
procedure, 47
frequency search, 66
parallel, 66
serial, 67
half-band FIR filter, 33
Inmarsat-P, 1
Iridium, 1
MCDD, 167
8-channel, 185
ML estimation, 97
OBP, 1
odd stacking, 31
OQPSK, 12
phase ambiguity, 170
phase error synchroniser, 130
acquisition time, 139
arctangent algorithm, 136
decision-directed ML, 131
noise performance, 148
non data-aided, 137
optimised FF algorithm, 144, 192
V&V algorithm, 141
PLL, 68
loop filter, 149
polyphase DFT structure, 11, 35
polyphase matrix structure, 35
Powell function, 49
QPSK, 12
recursive MCDD, 20
Rosenbrock function, 49
rotational FED, 82
new method, 84, 191
sampling rate, 31
critical, 31
oversampling, 31
sampling rate conversion, 23
decimation, 26
interpolation, 24
rational factor, 27
second-order section, 53, 57
noise variance, 59
synchronisation, 17
  carrier phase, 182
  Doppler, 176
  frame, 172
  joint, 179
  symbol timing, 180
TDMA, 7
timing error correction, 124
  polyphase filters, 124
timing error detection, 101
  DTTL, 115
  Gardner algorithm, 109
  Kobayashi algorithm, 102
  mM&M algorithm, 117
    optimised, 121, 191
  MMSE, 113
  two-point tracker, 107
tree demultiplexer, 9
  FIR, 10, 160, 190
  IIR, 11, 157, 189
WOAS, 39
Published Papers


8. **Self-Noise Cancellation of the Modified Mueller and Müller Algorithm.**

Abstract has been submitted to IEEE 3rd International Symposium on Communication Theory & Applications, Lake District, July 1995.
MATERIAL REDACTED AT REQUEST OF UNIVERSITY