

Spread Spectrum system analysis, development, and testing,
in a multipath UHF radio environment.

Terence Bayes
2

Submitted in accordance with the requirements for the
degree of PhD.

The University of Leeds

Department of Electrical and Electronic Engineering

January 1989

ABSTRACT

Direct Sequence Spread Spectrum systems presently find use in navigation and military communication systems. In recent years, however, interest has also been shown in the potential of such systems for Land Mobile Radio applications. In particular the broadband Spread Spectrum signal would be useful in combating the effects of narrowband fading experienced in the UHF radio environment.

In addition, vast improvements in the performance of digital integrated circuits is leading to the concept of the all-digital radio receiver. One such device made feasible by progress in miniaturisation is the Digital Matched Filter.

This work examines the operation of the Digital Matched Filter. It's applicability to Direct Sequence Spread Spectrum Systems is studied with a view to the possible implementation of such schemes in the Land Mobile Radio environment. Emphasis is placed on achieving maximum utilisation of the Digital Matched Filter resource. Information obtained from Digital Matched Filters in a complex baseband receiver is shown to assist rapid code synchronization, RF carrier recovery, and measurement of the multipath transmission channel profile.

A practical transmitter/receiver pair is described, and the results obtained suggest that future designers of Direct Sequence Spread Spectrum receivers may obtain benefits from the Digital Matched Filter approach, particularly in the area of rapid code synchronization.

ACKNOWLEDGEMENTS

I would like to extend special thanks to the following for their assistance during the course of this work:

Professor Peter Lawrenson and Professor Michael Howes, as heads of the Department of Electrical and Electronic Engineering, for their permission to use the facilities of the department.

Professor Peter Matthews for supervising the project, and providing much appreciated help and advice during the period of study and during the preparation of this thesis.

The Science and Engineering Research Council for providing financial support.

The Home Office Directorate of Telecommunications for providing funds for the project.

My wife, Sharon, for her patience during the preparation of this thesis.

CONTENTS

1	INTRODUCTION	1
1.1	History of the project	1
1.2	Background to the problem	2
1.3	Terminology used	3
	References	8
2	THE DIGITAL MATCHED FILTER	10
2.1	Introduction	10
2.2	DMF operation	11
2.3	DMF output statistics	18
2.3.1	Approximation for chip error probability with hard limiting	18
2.3.2	Approximation for DMF output - Bernoulli trials	21
2.3.3	DMF output - Code in phase	28
2.3.4	DMF output - Code out of phase	30
2.3.5	Noise only	32
2.4	Detection criterion	33
2.4.1	Code selection	33
2.4.2	Effect of modulation	33
2.4.2.1	Sequence inversion keying	33
2.4.2.2	On-off keying	34
2.5	Effect of quantization on DMF performance	37
2.6	Conclusions	44
	References	46
3	COMPLEX BASEBAND RECEIVER	58
3.1	Introduction	58
3.2	Receiver mixers	60
3.3	Lowpass filters	63
3.3.1	Choice of frequency response characteristic	64
3.3.2	Effect of filtering on correlation	64
3.3.2.1	Correlation of lowpass noise samples	64
3.3.2.2	Correlation of lowpass signal samples	67

3.3.3	Cutoff frequency and noise equivalent bandwidth	69
3.3.4	Filter loss relative to ideal chip matched filter	70
3.3.5	Filter phase and step response	75
3.3.6	Choice of practical lowpass filter	78
3.4	RF local oscillator and system clock relationship	80
3.5	Sample and hold and A/D converters	83
3.6	Digital matched filter	84
3.7	Combining algorithm	84
3.8	Conclusions	92
	References	96
4	SYNCHRONIZATION TECHNIQUES	110
4.1	Introduction	110
4.2	Synchronization and tracking schemes	111
4.2.1	Code synchronization acquisition	111
4.2.2	Code tracking	116
4.2.3	RF recovery	118
4.3	Digital synchronization system	119
4.3.1	DMF receiver code synchronization acquisition	120
4.3.1.1	Threshold detection on average algorithm output	123
4.3.1.2	Average of number of threshold crossings	124
4.3.1.3	Single threshold crossing with binary integration	129
4.3.2	Digital tracking and RF recovery loops	139
4.4	System study of control of a DMF based DSSS receiver	141
4.4.1	Conditions for code synchronization	145
4.4.2	Ability to achieve RF synchronization	148
4.4.3	Summary of properties of proposed receiver	153
4.5	Conclusions	155
	References	159
5	PROTOTYPE DIRECT SEQUENCE SYSTEM	172
5.1	Introduction	172

5.2	Prototype direct sequence transmitter	173
5.2.1	70 MHz oscillator and system clock generator	174
5.2.2	Code generators	175
5.2.3	Audio encoder	176
5.2.4	Delta modulator clock and synchronization generator	178
5.2.5	70 MHz limiter and mixer module	179
5.3	Prototype DSSS receiver	181
5.3.1	IF lowpass filter and amplifier	183
5.3.2	70 MHz oscillator and system clock generator	183
5.3.3	Power dividers and IF mixers	183
5.3.4	Baseband lowpass filters and video amplifiers	184
5.3.5	High speed comparators	184
5.3.6	Digital matched filters and arithmetic functions	185
5.3.7	Thumbwheel switch encoder	189
5.3.8	Digital comparator	190
5.3.9	Correlation processor and delta demodulator clock generator	191
5.3.10	Delta demodulator and lowpass filter	192
5.3.11	Receiver sequence generator	192
5.4	Conclusions	192
	References	195
6	PRACTICAL RESULTS	211
6.1	Introduction	211
6.2	Broadband noise generator	211
6.3	Chip errors	214
6.4	Symbol errors	217
6.5	DMF output probability distribution	219
6.6	Multipath propagation measurements	222
6.7	Examination of algorithm mean output	232
6.8	Probability algorithm exceeds thresholds	235
6.9	Conclusions	237
7	CONCLUSIONS	254

TABLES AND FIGURES

Table 2-1	Mathematical operations required to solve (2-9) and (2-10)	15
Table 2-2	Voltage transfer function for 3 bit quantizer	41
Table 2-3	Minimum and maximum loss due to multibit quantization	41
Table 3-1	Practical lowpass filter responses, ratio of noise equivalent bandwidth to 3dB bandwidth	70
Table 3-2	Practical lowpass filter responses, normalized risetime	78
Table 4-1	Probability that algorithm output exceeds threshold	126
Table 4-2	Mean number of synchronization searches, single threshold crossing with binary integration	134
Table 4-3	Mean number of synchronization searches, single threshold crossing with 1 and 4 binary integrators	137
Table 6-1	Predicted and measured DMF outputs with multipath components	228
Figure 2-1	Correlation Detector	48
Figure 2-2	m by n bit Digital Correlator	48
Figure 2-3	Canonical form of Binary Correlator	49
Figure 2-4	Binary Correlator with pipelined adders	49
Figure 2-5	Binary Correlator Implementation	50
Figure 2-6	Implementation of Binary Correlator from expression (2-10)	50
Figure 2-7	Gaussian probability density function	51
Figure 2-8	Cumulative Gaussian probability function	51
Figure 2-9	Hard limiter	51
Figure 2-10	Error due to approximation (2-21)	51
Figure 2-11	DMF output PDFs	52
Figure 2-12	Cumulative PDF of DMF output	53

Figure 2-13	Autocorrelation function of practical code for optimum detection	54
Figure 2-14	DMF output PDF, noise only case	54
Figure 2-15	DMF output PDF for code in phase and code out of phase cases	55
Figure 2-16	DMF output PDF for code in phase and noise only cases	56
Figure 2-17	Effect of quantization threshold on processing gain loss for constant input snr	57
Figure 3-1	Complex Baseband Receiver	98
Figure 3-2	Modulus of I and Q channel outputs with varying relative RF carrier phase	98
Figure 3-3	Modulus of correlation of lowpass noise samples for test filters	99
Figure 3-4	Modulus of correlation of lowpass signal samples for test filters	100
Figure 3-5	Loss relative to ideal chip matched filter for test filters	101
Figure 3-6	Lowpass filter risetime definitions	102
Figure 3-7	Lowpass filter risetime superimposed on mixer output null pattern	103
Figure 3-8	Algorithm to modify DMF output	104
Figure 3-9	DMF combining algorithms	104
Figure 3-10	Output statistics for algorithms, constant snr, varying phase	105
Figure 3-11	Output statistics for algorithms, $V/\sigma=0.5$	106
Figure 3-12	Output statistics for algorithms, $V/\sigma=1.0$	107
Figure 3-13	Output statistics for algorithms, increasing snr, constant phase (0 degrees)	108
Figure 3-14	Output statistics for algorithms, increasing snr, constant phase (45 degrees)	109
Figure 4-1	Maximum likelihood detector	161
Figure 4-2	Active correlator	161
Figure 4-3	Digital correlator	161
Figure 4-4	Stepping correlator	161
Figure 4-5	Delay lock loop	162
Figure 4-6	Dither loop	163
Figure 4-7	RF recovery loops	164

Figure 4-8	Algorithm output probabilities, $p_e=0.4437$	165
Figure 4-9	Algorithm output probabilities, $p_e=0.4007$	166
Figure 4-10	Algorithm output probabilities, $p_e=0.3274$	167
Figure 4-11	Production of average algorithm output	168
Figure 4-12	Production of sum of threshold crossings	168
Figure 4-13	Code epoch detection	169
Figure 4-14	Code synchronization detection by single threshold crossing with binary integration	169
Figure 4-15	Code synchronization detection by 4 single threshold crossing detectors with binary integration	170
Figure 4-16	DSSS Transmitter block diagram	170
Figure 4-17	DSSS receiver block diagram	171
Figure 4-18	DSSS receiver RF down conversion section	171
Figure 4-19	DSSS receiver baseband section	171
Figure 5-1	Practical DSSS transmitter block diagram	197
Figure 5-2	70 MHz oscillator	198
Figure 5-3	70 MHz divider and system clock generator	198
Figure 5-4	Transmitter code generator	199
Figure 5-5	Delta modulation	200
Figure 5-6	CVSD modulator	200
Figure 5-7	CVSD clock and synchronization generator	201
Figure 5-8	Transmitter IF modulator section	201
Figure 5-9	IF lowpass filter	201
Figure 5-10	Practical DSSS receiver block diagram	202
Figure 5-11	Receiver input filter and complex baseband converter	203
Figure 5-12	Baseband lowpass filter	204
Figure 5-13	Video amplifier	204
Figure 5-14	High speed comparators	205
Figure 5-15	Digital matched filter	206
Figure 5-16	Matched filter combiner, arithmetic functions, and digital comparator	207
Figure 5-17	Thumbwheel switch threshold encoder	208
Figure 5-18	Correlation pulse processor, CVSD demodulator clock generator, and receiver sequence generator	209
Figure 5-19	CVSD demodulator	210
Figure 5-20	CVSD demodulator audio lowpass filter	210

Figure 6-1	Noise generator amplitude probability distribution	242
Figure 6-2	Measured chip error rate	243
Figure 6-3	Measured symbol error rate	243
Figure 6-5	DMF output probability distribution, $p_e=0.365$	244
Figure 6-6	DMF output probability distribution, $p_e=0.434$	245
Figure 6-5	DMF output probability distribution, code out of phase and noise only	246
Figure 6-7	DMF mean output against delay with delayed path 6dB below main path, delayed path not inverted	247
Figure 6-8	DMF mean output against delay with delayed path 6dB below main path, delayed path inverted	248
Figure 6-9	DMF mean output against delay with delayed path 3dB below main path, delayed path not inverted	249
Figure 6-10	Square root algorithm mean output against relative carrier phase	250
Figure 6-11	Probability algorithm exceeds threshold output, $p_e=0.4437$	251
Figure 6-12	Probability algorithm exceeds threshold output, $p_e=0.4007$	252
Figure 6-13	Probability algorithm exceeds threshold output, $p_e=0.3274$	253

ABBREVIATIONS

ADC	Analogue to Digital convertor
AWGN	Additive White Gaussian Noise
BCD	Binary Coded Decimal
BPSK	Bipolar Phase Shift Keying
CDMA	Code Division Multiple Access
CPU	Central Processor Unit
CVSD	Continuously Variable Slope Delta modulator
δ	Dirac delta function
dB	Decibel
DMF	Digital Matched Filter
DSP	Digital Signal Processing
DSSS	Direct Sequence Spread Spectrum
ECL	Emitter-Coupled Logic
HF	High Frequency
I	In phase
IF	Intermediate Frequency
LMR	Land Mobile Radio
LSI	Large Scale Integration
MSB	Most Significant Bit
OOK	On Off Keying
Prob{x}	Probability that x is true
PROM	Programmable Read Only Memory
PSK	Phase Shift Keying
Q	Quadrature phase
RARASE	Recursion Aided Rapid Acquisition by Sequential Estimation
RASE	Rapid Acquisition by Sequential Estimation
RECT(x)	Rectangular function
RF	Radio Frequency
SIK	Sequence Inversion Keying
SINC(x)	Sin(x)/x
snr	Signal to noise r.m.s. voltage ratio
TDMA	Time Division Multiple Access
TTL	Transistor-Transistor Logic
UHF	Ultra High Frequency
VCO	Voltage Controlled Oscillator
VLSI	Very Large Scale Integration
Rect($\frac{t}{\tau}$)	= 1 $ t \leq \tau/2$
	= 0 otherwise

CHAPTER 1

INTRODUCTION

1.1 HISTORY OF THE PROJECT

The investigation of Direct Sequence Spread Spectrum (DSSS) systems began at Leeds in 1978 in collaboration with the Home Office Directorate of Telecommunications. The resulting broad paper study [MATTHEWS P.A. and BAJWA A.S 1979] identified the lack of published information in the public domain on the subject of practical DSSS systems. It concluded that there was a need for practical investigation into DSSS schemes, before further consideration could be given to the application of such techniques to the Land Mobile Radio (LMR) environment.

The work continued with a further study assessing the likely user capacity of such a scheme in a mobile radio system [DRURY 1981]. Whilst the user capacity did not compare favourably with proposed narrowband analogue schemes [FRENCH 1979], [TURIN 1984], the DSSS approach offered the potential advantage of some immunity to the problems of multipath fading. Further being a digital modulation scheme it naturally provided the facility for the inclusion of system control data alongside the digitized analogue 'message'. Drury concluded that the need to provide synchronization at the receiver would be a serious limitation of the active form of receiver correlation if long sequences were applied.

A further study then began in 1981, to investigate the application of Digital Matched Filters (DMFs) to receiver correlation, again in collaboration with the Home Office. The intention of the design was to avoid the need for synchronization by the use of matched filtering techniques, and to perform field trials on the system.

The study continued with support from the Home Office until March 1984, at which point several problems had been identified. The most serious problem was the requirement for receiver synchronization, a feature which had been considered undesirable at the specification stage. Furthermore difficulties in achieving operation at the desired chip rate had indicated the need for careful redesign, and highlighted potential problems in cascading short DMFs in an attempt to match to long code sequences.

Despite these difficulties it was felt that a detailed study of a receiver structure would be worthwhile, with particular emphasis on identifying losses incurred by the receiver due to non-ideal operation, and the possibility of achieving more rapid synchronization than could be achieved by an active correlator. This information could then be applied to the operation of a DSSS system in the LMR environment.

1.2 BACKGROUND TO THE PROBLEM

There is much literature on the subject of DSSS systems, and there has been interest in the topic since the early 1950's, following the pioneering work of Shannon. Areas of frequent interest are code properties e.g. [GOLD 1967], [SARWATE 1984], and spectral efficiency e.g. [GARDNER and ORR 1979], [HUI 1984]. In contrast few practical systems have been described, and this is indeed noticeable on the topic of practical implementations utilising the DMF as the correlation element.

At the commencement of the work in 1983 no references to practical DMF based DSSS receivers could be found. During 1984 two descriptions were given, [ASH 1984], and [SKAUG 1984], with a further paper on an analogue implementation presented by [YAMADA, DAIKOKU and USUI 1984]. The implementation described by Skaug did not include details of how the synchronization and tracking

functions were achieved as the work concentrated largely on the performance of the system in an HF radio environment. The receiver described by Ash used a delay lock loop as the synchronization scheme, a technique associated with active correlator designs and well documented as such.

Such an implementation is a somewhat inefficient use of the DMF as it does not provide a more rapid synchronization acquisition than can be achieved by the analogue delay lock loop. Clearly there was a requirement to investigate methods of obtaining rapid synchronization acquisition in DMF based DSSS receivers. This property could produce an increase in the favour of DSSS systems incorporating DMFs, which could lead to reduced system costs by the adoption of Large Scale Integration (LSI) techniques and volume sales.

One particular area in which an enhancement in synchronization performance would be a desirable property is burst transmission, particularly useful in covert applications or systems operating on an interference limited basis.

1.3 TERMINOLOGY USED

Where possible terminology has been chosen in accordance with that generally used in the literature. The meaning of terms used throughout this work is given here in the form of a brief introduction to Direct Sequence Spread Spectrum systems.

The term spread spectrum is taken to mean a signal which has been encoded in a manner such as to introduce considerable redundancy in the signal prior to transmission, and hence a considerable expansion in transmission bandwidth than is necessary to accommodate the signal prior to spreading. Compared to an equivalent power non-spread spectrum signal the spread spectrum signal will

be received at a lower snr, but over a wider bandwidth. This is a direct attempt to achieve improved utilisation of a transmission channel in terms of the maximum attainable channel capacity, as indicated by the work of Shannon (see for example [CARLSON 1975] pp350 to 371). Shannon's fundamental theorem states that the capacity of a channel to transmit information is a function of the channel bandwidth and the received snr. The information reception can be made at an arbitrarily low error rate. For a constant transmission rate, bandwidth expansion allows a decrease in transmitted power. Alternatively a reduction in transmission bandwidth must be accompanied by an increase in transmitted power. The exchange is approximately exponential, see for example [CARLSON 1975] for an excellent description, in favour of increasing bandwidth.

In a DSSS scheme direct bandwidth expansion is achieved by replacing each bit of digital message data by a sequence of 'm' bits, or 'chips'. The m bits constitute the 'code' sequence, or code, and are generally chosen from a set of orthogonal or nearly orthogonal codes. Near orthogonality is generally achieved by selecting binary codes which have low cross correlation properties. The digital message information is transmitted by modulation of the code sequence, most frequently by Sequence Inversion Keying (SIK). In this modulation scheme a code sequence is sent to represent a message bit equal to a logical true (1), and the code sequence logically inverted is sent to represent a message bit equal to a logical false (0). In this work a modulated code of length m is termed a message 'symbol'.

The received signal sequence can be correlated against the local reference code sequence to provide a detection capability. In this work the correlation process is performed by the Digital Matched Filter (DMF). A receiver can discriminate between a set of sequences received in the same bandwidth provided it has knowledge of the transmitted code sequence and the time of arrival of the end of the

code sequence. The property of signal discrimination is termed 'processing gain', and is a direct result from Shannon's law. If the receiver has knowledge of the transmitted code then each received data bit can be added coherently. The signal voltages will add directly, whereas the noise components will add on a power basis, since these will not be coherent. A net improvement in signal to noise ratio is achieved, by a factor $10\log(m)$ dB, and this is the familiar expression for processing gain.

To utilise the processing gain a knowledge of the transmitted code must be accompanied by knowledge of the time of arrival of the end of the code, such that correlation of the m received chips can be correctly made with the m reference chips. Knowledge of the time of arrival of the code start or end is termed 'code synchronization', and forms a considerable part of the work presented in this thesis. It is apparent from the above discussion that lack of code synchronization prevents access to the DSSS processing gain, and much effort has been spent in this area for this reason.

In terms of the operation within the DMF, code synchronization can be expressed in terms of the relative phase of the received code with the reference code. The code phase is the offset between a pair of identical codes in terms of the number of chips distance between them. This may be visualised as the state of a received code in a DMF relative to a static reference code. A code in phase condition occurs when the received code has a phase of 0 chips relative to the reference code. This is the condition required for code synchronization and only occurs at 1 of m possible code phases. Alternately code out of phase is all conditions where a received code does not have a phase of 0 chips relative to the reference code, and naturally occurs for $m-1$ code phases.

Finally signal to noise ratios (snrs) are quoted in terms of signal voltage amplitude to noise voltage rms

(V/σ) or chip energy to noise power density (E_c/N_o). Where comparisons are made Bipolar Phase Shift Keying (BPSK) modulation is assumed and the relationship

$$E_c/N_o = 10\log(V^2/2\sigma^2)$$

is assumed [CARLSON 1975].

1.4 SCOPE OF THE THESIS

Chapter 2 investigates the internal workings of the DMF and its performance in a statistical sense. The approximate loss incurred by using a single level of quantization is derived.

In chapter 3 the elements required to augment two DMFs in a complex baseband receiver are studied, with emphasis placed on the losses incurred by practical implementations. Choice of lowpass filters is shown to be a complex but important aspect of receiver design. New work is presented in the area of DMF output combining algorithms, and it is shown that an estimate of the received snr can be obtained.

Chapter 4 covers the important topic of receiver synchronization, with a summary of present techniques. New techniques for achieving rapid synchronization are given. These make maximum use of the information available from the DMFs.

Construction of a practical transmitter/receiver pair is described in chapter 5 and potential problem areas are identified. The descriptions and circuits presented indicate that there is still some way to go in terms of miniaturisation before hand held DSSS receivers are a practicality.

Chapter 6 presents the results obtained from the

practical receiver. Measurements have been made in the presence of noise in support of the ideas presented in chapters 2, 3, and 4.

Finally chapter 7 draws together the results to conclude the potential of rapid acquisition techniques in DSSS systems which use the DMF.

REFERENCES

- ASH C.P. 'Digital implementation of a narrowband satellite receiver', IEE Proceedings., Vol 131 Pt. F, No.6 October 1984, pp 570 to 576
- CARLSON A.B 'Communication Systems', McGraw-Hill electrical and electronic engineering series, 1975.
- DRURY D. 'Direct sequence spread spectrum techniques for land mobile radio applications', 1981, PhD Thesis, University of Leeds.
- FRENCH R.C. 'The effect of fading and shadowing on channel reuse in mobile radio', IEEE Trans. VT-28 (3), 1979, pp171-181.
- GARDNER C.S. and ORR J.A. 'Fading effects on the performance of a spread spectrum multiple access communication system', IEEE Trans. Com-27, 1979, pp143-149.
- GOLD R. 'Optimal binary sequences for spread spectrum multiplexing', IEEE Trans. IT-13, 1967, pp619-621.
- HUI J.Y.N. 'Throughput analysis for code division multiple accessing of the spread spectrum channel', IEEE Vol. SAC-2, No. 4, 1984, pp482-486.
- MATTHEWS P.A. and BAJWA A.S. 'Spread spectrum signalling in Home Office radio schemes', 1979, Phase 1 Final Report, Home Office Directorate of Telecommunications, Agreement No DT 122/180/12.
- SARWATE D.V. 'Mean square correlation of shift register sequences', IEE Proc. Vol. 131 Part F No. 2, 1984, pp 101-106.

- SKAUG R. 'Experiment with spread spectrum modulation on an HF channel', IEE Proc. Vol. 131 Part F No. 1 1984, pp 87-91.
- TURIN G.L. 'The effects of fading on the performance of direct sequence CDMA systems', IEEE Vol. SAC-2 No. 4, 1984, pp 597-603.
- YAMADA K. DAIKOKU K. and USUI H. 'Performance of portable radio telephone using spread spectrum', IEEE Vol. COM-32 No. 7, 1984, pp 762-768.

CHAPTER 2

THE DIGITAL MATCHED FILTER

2.1 INTRODUCTION

The main emphasis of this work is the study of the application of digital matched filters with single bit quantization to a DSSS system. There are several reasons for the adoption of such an approach, for example the minimization of receiver hardware, complexity, and cost. However general expressions defining the process of digital matched filtering (or digital correlation) will be determined for the general case of a length m bit by width n bit DMF. The single bit DMF ($n=1$) will then be studied in detail as a special case.

This chapter discusses the theoretical operation of a DMF and gives a formal definition for this. The single-bit (2-level) DMF is studied in detail and the output statistics for this filter are evaluated when the input data stream is corrupted with AWGN. Suitable approximations for the DMF output will be formulated as these are required later. A new approach to the calculation of the loss incurred by the DMF relative to the ideal matched filter is investigated for several levels of quantization. The results obtained are compared with those of other authors [BRUNO 1970], [TURIN 1976], [BAIER and BAIER 1984]. This technique can be used to extend the analysis beyond the limiting case of a data stream corrupted in AWGN provided the probability of quantizer level occupancy can be determined from the noise statistics.

2.2 DMF OPERATION

It was shown in chapter 1 that the main processing operation performed by a DSSS receiver is correlation. The primary function of the signal processing section of the receiver is to calculate the correlation between the received signal $w(t)$ and the reference signal $r(t)$ stored or generated in the receiver, such that a detection decision can be made with a finite error probability, figure 2-1.

We begin the analysis by observing that the reference signal $r(t)$ is periodic with period R , where $R=mT$, m is the number of chips in a given code, and T is the chip period. Therefore the correlation may be performed over one period R of $r(t)$ and can be defined as

$$v(y) = \int_0^R r(t) w(t-y) dt \quad (2-1)$$

It has been shown [STREMLER 1977] that the processes of correlation and matched filtering are related and therefore we begin by expressing the appropriate matched filtering operation as

$$v(t) = \int_0^R r(R-y) w(t-y) dy \quad (2-2)$$

We may now consider a digital approach to the matched filtering operation by taking sampled and quantized versions of $r(t)$ and $w(t)$ where $r_k=r(kT)$ and $w_k=w(kT)$ are

samples of $r(t)$ and $w(t)$ at $t=kT$ respectively, T is the chip period, and k is an integer. In most applications m will be equal to the number of samples in the reference signal $r(t)$. This may not be the case for applications such as ranging, where accurate chip alignment is required. In this case multiple samples of a chip may be required resulting in a proportional increase in the length of the matched filter.

Now (2-2) may be expressed in terms of the sampled quantities

$$v_k = \sum_{i=0}^{m-1} r_{k-i} w_{k-i} \quad (2-3)$$

where m is the code length.

We may write r_k as a j bit number such that

$$2r_k = \sum_{g=0}^{j-1} r_{k,g} 2^{-g} \quad (2-4)$$

and w_k as an n bit number such that

$$2w_k = \sum_{h=0}^{n-1} w_{k,h} 2^{-h} \quad (2-5)$$

The coefficients $r_{k,g}$ and $w_{k,h}$ take on the values $+1, -1$ only, and further r_k and w_k are restricted to lie in the range $\{+1, -1\}$. Therefore for the sampled and quantized signal and reference the expression (2-3) may be written

$$v_k = \sum_{i=0}^{m-1} \left\{ \sum_{g=0}^{j-1} r_{k-i,g} \right\} 2^{-g} \left\{ \sum_{h=0}^{n-1} w_{k-i,h} \right\} 2^{-h} \quad (2-6)$$

$$= \sum_{h=0}^{n-1} \left\{ \sum_{g=0}^{j-1} 2^{-(g+h)} \right\} \left\{ \sum_{i=0}^{m-1} r_{k-i,g} w_{k-i,h} \right\} \quad (2-7)$$

The full implementation of (2-7) would require a j bit by n bit by m bit correlator structure.

Such a scheme is practical in the sense that digital correlation leads directly to a bit-slice approach to implementation [CHRISTIE 1986]. However (2-7) defines the most general case of a quantized signal data stream correlated against a reference which has a set of coefficients r_k which may each take on any value in the range $0..2^{j-1}$. It has been shown [HOLMES 1982] that for optimum correlation the reference code should be filtered in a manner identical to the filtering imposed on the transmitted code by transmitter, channel, and receiver filtering. Therefore the coefficients chosen for r_k should represent the effect of this filtering on the reference code. The transfer functions of the transmitter and receiver filters can be determined but in practice the overall transfer function of the required filtering process

is not likely to be known at the receiver. This is due to the time varying response of the channel [STREMLER 1977]. Therefore the analysis proceeds with the assumption that reference code filtering is not practical.

We may thus write (2-4) as $\{r\} = \{+1, -1\}$. The reference can therefore be written in the same form as the transmitted code prior to modulation

$$r(t) = \sum_{j=0}^{m-1} d_j \cdot y(t-jT) \quad (2-8)$$

where $d_j = \{+1, -1\}$ for $j = \{0, \dots, m-1\}$

and $y(t) = 1$ for t in the range $0..T$

Having simplified (2-4) and (2-5) it is now possible to simplify (2-7) giving

$$v_k = \sum_{h=0}^{n-1} \left\{ \sum_{i=0}^{m-1} r_{k-i,0} w_{k-i,h} \right\} \quad (2-9)$$

It can be seen from (2-9) that the correlation result is formed from a scaled summation of n separate length m correlations, shown schematically in figure 2-2. Since the reference coefficients are not a function of h the order of summation may be reversed giving

$$v_k = \sum_{i=0}^{m-1} r_{k-i,0} \sum_{h=0}^{n-1} w_{k-i,h} \quad (2-10)$$

It is interesting to observe the difference in computational effort required to evaluate (2-10) relative to the evaluation of (2-9). Assuming that a summation of y elements requires y additions we may calculate the computational effort required by observing that (2-9) is n sets of m multiplications and m additions with a further n additions for the final summation and (2-10) is m sets of additions and 1 multiplication with a further m additions for the final summation. This gives the results of table 2-1.

TABLE 2-1

Mathematical operations needed to solve (2-9) and (2-10)

Equation	(2-9)	(2-10)
Additions	$n \cdot (m+1)$	$m \cdot (n+1)$
Multiplications	$m \cdot n$	m

In general multiplication of digital numbers is more computationally intensive than addition since multiplication is usually performed by a series of addition and shift operations. It would seem, therefore, that the calculation of (2-10) is preferable to that of (2-9) on the basis of computational complexity. However to test this hypothesis it is necessary to consider the way in which the

operations of addition and multiplication might be performed by a digital correlator. The summation over m in a length m bit by width n bit correlator can be performed canonically by an array of n adders, each with m inputs. This approach is likely to suffer from speed limitations at high bit rates [CHRISTIE 1986]. Figure 2-3 shows one of the n sections required for a full implementation.

Alternatively a pipelined approach can be adopted resulting in the need for n by m adders, figure 2-4, where the number of inputs per adder depends on the position in the pipeline but reaches a maximum of only 9 bits for a 511 bit correlator. This is a much more realisable technique and has been adopted in practical devices [TRW 1980], [MEDL 1986]. Initially the operation of multiplication appears more complicated than that of addition. However close examination of (2-9) shows that the multiplication of $r_{k-i,0}$ by $w_{k-i,h}$ is simply the multiplication of two single bit numbers. By suitable scaling of r and w , and accounting for a resulting offset in v , it is possible to replace the multiplication operation by a simple exclusive-NOR function. This simplifies the implementation considerably, and one section of the correlator corresponding to the input stream $w_{k-i,1}$ is shown in figure 2-5. Since only single bit data is being correlated only a single bit shift register is required for each element of w .

The final correlation result is calculated by taking a weighted sum of the outputs from each section of the correlator corresponding to each element of w . Therefore the evaluation of (2-9) can be performed by an array of n single bit correlators of length m followed by a suitable adder. Now the evaluation of (2-10) requires an array of m adders of width n plus suitable pipelined adders to form the result, figure 2-6. This is clearly a less efficient solution. Therefore correlator implementation according to the order of operations of (2-9) is more efficient than that of (2-10) despite the apparent number of

multiplications required. This is because the multiplications of (2-9) reduce to single bit multiplications. Further the binary representation of these numbers allows these multiplications to be implemented as the simple exclusive-NOR function. This is not the case in (2-10) where a single bit number must be multiplied by an n bit number.

Finally we consider the practical implications of taking a single bit approximation to $w(t)$. For this case $n=1$, and (2-9) and (2-10) become

$$v_k = \sum_{i=0}^{m-1} r_{k-i,0} w_{k-i,0} \quad (2-11)$$

It is interesting to observe that under this condition the theoretical limitation on the range of allowable values for $w(t)$ is removed, since $w_{k,h}$ is determined simply by the sign of $w(t)$. Further if it is necessary to constrain $w(t)$ to lie in the range $\{+1,-1\}$ then a knowledge of the expected amplitude of $w(t)$ must be assumed at the receiver. This is necessary such that the quantization levels may be determined, or automatic gain control implemented. The effectiveness of the receiver may thus be restricted and it will be shown that the potential performance improvement due to multibit quantization may not always be achieved in practice. This is particularly serious during the critical stage of synchronization, where it is unlikely that the receiver will have knowledge of the received snr.

It will be shown in section 2.5 that the additional expense and complexity of a multibit DMF structure may be wasted by incorrect choice of quantization levels for a given input snr. Although the single bit DMF may never

quite achieve the performance of the multibit DMF the relative performance loss is surprisingly low.

One advantage of the single bit DMF, therefore, is the removal of the need for automatic gain control of the received signal $w(t)$.

In summary it can be seen that the function of the DMF is to approximate the operation of an analogue matched filter, or correlator. This is achieved by performing summations of the signal and reference in sampled and quantized form. The DMF is constructed from an array of shift registers and adders. The number and complexity of the required adders is reduced by a parallel approach. This reduces many of the necessary addition operations to the simpler Exclusive-NOR operation.

2.3 DMF OUTPUT STATISTICS

We now examine the statistics of DMF operation. This will allow an estimation to be made of the performance of the DMF based DSSS receiver when signals are received in the presence of noise. This is important since DSSS systems are generally operated at low received snrs, either due to the desire for signal concealment, or as a result of operation at interference limited levels in overlay schemes.

2.3.1 APPROXIMATION FOR CHIP ERROR PROBABILITY WITH HARD LIMITING

We begin by examining the statistics of chip decisions. The Gaussian probability density function for a bipolar binary signal of equally likely amplitudes of $+V_c$ and $-V_c$ in bandlimited AWGN of zero mean and variance σ^2 is given by [PAPOULIS 1984]

$$q(V) = e^{-(V-V_c)^2/2\sigma^2} / ((2\pi)^{1/2} \sigma) \quad (2-12)$$

This is the probability that the instantaneous voltage will be in the range V to $V+dV$, figure 2-7. The cumulative distribution function, figure 2-8, or the probability that the instantaneous voltage will be less than some value x is given in normalised form as

$$Q(V_c) = \int_{V_c}^{\infty} q(x) dx \quad (2-13)$$

This then gives the error probability for a hard limiter operating under the given conditions, with $x=V_c$, and decision threshold set to 0 volts, figure 2-9, giving

$$\text{Pr(chip error)} = Q(V_c/\sigma) \quad (2-14)$$

The integral of (2-13) cannot be evaluated in closed form and therefore numerical techniques or tables are necessary to evaluate the function, [CARLSON 1975]. The function $Q(V)$ is identical to the complementary error function, $\text{Erfc}(V)$ [STREMLER 1977].

It is generally desirable to operate spread spectrum systems at low received snrs where advantage can be taken of the potential processing gain of the receiver. Consequently the expected value of V/σ at the receiver will be less than 1, and under this condition we may approximate the function $Q(V)$.

By substitution we obtain

$$Q(V) = (2\pi)^{-1/2} \int_V^{\infty} e^{-y^2/2} dy \quad (2-15)$$

Since terms in the expansion of e^x cannot be evaluated at infinity it is necessary to introduce the error function

$$\text{erf}(K) = (2/\pi)^{1/2} \int_0^K e^{-t^2} dt \quad (2-16)$$

which is related to the function $Q(K)$ by

$$Q(K) = (1 - \text{erf}(K/2^{1/2})) / 2 \quad (2-17)$$

Expanding (2-16) in terms of e^{-t^2} gives [PAPOULIS 1984]

$$\text{erf}(K) = 2e^{-K^2} / \pi^{1/2} \sum_{n=0}^{\infty} (2^n K^{2n+1}) / (1.3 \dots (2n+1)) \quad (2-18)$$

$$= 2e^{-K^2} / \pi^{1/2} (K + 2K^3/3 + 4K^5/15 + \dots)$$

$$= 2K/\pi^{1/2} \quad (2-19)$$

for small K . Therefore from (2-14), (2-17), and (2-19) we obtain

$$\text{Pr}(\text{chip error}) \approx 1/2(1-(2/\pi)^{1/2}K) \quad (2-20)$$

$$= 1/2 - K/(2\pi)^{1/2} \quad (2-21)$$

The error due to using this approximation compared to an accurate numerical technique is shown in figure 2-10 for V/σ in the range 0 to 0.5, (E_c/N_o less than -9dB for BPSK modulation). In this range the error is less than 3 percent.

2.3.2 APPROXIMATION FOR DMF OUTPUT - BERNOULLI TRIALS

We now extend the analysis to consider the case of m binary trials. Each trial is assumed independent but the error probability is constant for all trials. Such an event is generally termed a Bernoulli trial [PFEIFFER and SCHUM 1973]. We assume without loss of generality that the desired outcome is a string of m ones. If we now assign an error probability, p_e , to each event and further assume that p_e remains constant throughout m trials then we may calculate the pdf of the possible outcomes from the Binomial theorem [PAPOULIS 1984]

$$\Pr(\text{result}=K) = C \binom{m}{K} p_e^K (1-p_e)^{m-K} \quad (2-22)$$

where

$$C = \binom{m}{K} = \frac{m!}{(m-K)!K!} \quad (2-23)$$

For detection we are interested in finding the probability that the result exceeds a value K . Therefore from (2-22) we define the discrete output probability of an outcome K as

$$\Pr(\text{result}=K) = b(m, K, p_e) \quad (2-24)$$

Whence

$$\Pr(\text{result} < K) = \sum_{j=0}^{K-1} b(m, j, p_e) \quad (2-25)$$

$$= B(m, K-1, p_e) \quad (2-26)$$

Further the probability that the result exceeds K is found by subtraction from unity, this being the total cumulative probability, giving

$$\Pr(\text{result} > K) = 1 - B(m, K, p_e) \quad (2-27)$$

Again in (2-24) and (2-26) the numerical evaluation of factorials and summations becomes cumbersome, particularly for large values of m . However these functions may be approximated in a form more convenient for direct evaluation. Introducing the notation $p=p_e$, $q=1-p_e$, it can be shown [PAPOULIS 1984] that if $mpq > 1$ then

$$\sum_{K=0}^m p^K q^{(m-K)} \approx e^{-(K-mp)^2 / (2mpq)} \quad (2-28)$$

for K in the neighbourhood $(mpq)^{1/2}$ of mp . For p in the range of practical interest (0.3 to 0.5) this will imply large m .

Equation (2-28) is an application of the DeMoivre-Laplace theorem, and is based on Stirling's approximation

$$m! \approx m^m e^{-m} / (2\pi m)^{1/2} \quad (2-29)$$

Returning to the definitions of (2-12) and (2-13) in normalized form we have

$$q(y) = e^{-y^2/2}/(2\pi)^{1/2} \quad (2-30)$$

and

$$Q(x) = \int_x^{\infty} q(y) dy \quad (2-31)$$

We can equate terms from (2-12), (2-28) and (2-30) to show that for $0.3 < p < 0.5$

$$C_p^m q^{m-K} \approx (mpq)^{-1/2} q((K-mp)/(mpq)^{1/2}) \quad (2-32)$$

Therefore using (2-32) in (2-22) the evaluation of the probability that the result is K for a given value of p_e is reduced to the evaluation of the normal curve.

Further

$$\Pr(K_1 < K < K_2) = \sum_{K=K_1}^{K_2} C_p^m q^{m-K} \quad (2-33)$$

$$= (mpq)^{1/2} \sum_{K=K1}^{K2} q((K-mp)/(mpq)^{1/2}) \quad (2-34)$$

Now since the curve

$$q((K-mp)/(mpq)^{1/2}) = e^{-(K-mp)^2/2mpq} \quad (2-35)$$

is nearly constant over a unit interval for $mpq > 1$ the area in that interval is approximately equal to the ordinate, therefore

$$\sum_{K=K1}^{K2} q((K-mp)/(mpq)^{1/2}) \approx \int_{K1}^{K2} q((K-mp)/(mpq)^{1/2}) dK \quad (2-36)$$

$$= Q((K2-mp)/(mpq)^{1/2}) - Q((K1-mp)/(mpq)^{1/2}) \quad (2-37)$$

If the lower limit is zero then it can be shown [PAPOULIS 1984] that the second term of (2-37) is 0 giving

$$B(m, K, p_e) \approx Q((K-mp_e)/(mp_e(1-p_e))) \quad (2-38)$$

An improved approximation can be found by expressing $q(K)$ as the incremental slope of $Q(K)$ rather than it's derivative.

Putting

$$M = (mpq)^{1/2}$$

we obtain

$$q(K) \approx M \{Q((K+0.5)/M) - Q((K-0.5)/M)\} \quad (2-39)$$

and therefore

$$\sum_{K=K_1}^{K_2} C^m p^K q^{(m-K)} \approx Q((K_2+0.5-mp)/M) - Q((K_1-0.5-mp)/M) \quad (2-40)$$

Again if $K_1=0$ it can be shown that

$$B(m, K, p_e) \approx Q((K+0.5+mp)/M) \quad (2-41)$$

The similarity between the functions (2-12) and (2-38) can be investigated by evaluating the mean and variance for the Binomial distribution.

For this we have

$$\text{Mean}(K) = \sum_{K=1}^m K C_K^m p^K q^{(m-K)} \quad (2-42)$$

$$= \sum_{K=1}^m K \frac{m!}{((m-K)!K!)} p^K q^{(m-K)} \quad (2-43)$$

$$= mp \sum_{j=0}^{m-1} C_j^{m-1} p^j q^{(m-j)} \quad (2-44)$$

where $j=K-1$, from which

$$\text{Mean}(K) = mp \quad (2-45)$$

since the summation of (2-44) is equal to 1.

Similarly the variance may be determined

$$\text{Variance}(K) = \sum_{K=1}^m K^2 C_K^m p^K q^{(m-K)} \quad (2-46)$$

which by a similar method gives

$$\text{Variance}(K) = mpq \quad (2-47)$$

2.3.3 DMF OUTPUT - CODE IN PHASE

The previous discussion introduced the output pdf of a Bernoulli trial of m independent binary events or decisions. We may now consider the probability of a particular output from an m bit DMF for an input data code, where the data is a hard-limited estimate of a known binary code in AWGN, figure 2-9. First we consider the case where the sampled data in the on-line (received signal) register (figure 2-5) of the DMF is in alignment with the fixed reference register data. This condition is generally termed 'code in phase', or the code 'epoch'. We may evaluate the probability of a specific output value from the DMF for a given snr at the hard-limiter input by using expressions (2-21) and (2-24) of the previous section. The actual form of the code, that is the number of ones and their positions, is not relevant to this part of the argument since each data element is assumed to be independent of all others, and their value (zero or one) is based on the probability that any chip is in error from the corresponding element in the reference register. The in phase case is, therefore, a straightforward Bernoulli trial of m independent events. If SIK data modulation is used then, in the absence of noise or interfering signals and hence hard-limiter errors, the expected output from the DMF will be m for a data symbol 1 received and 0 for a data symbol 0 received, since in this case the data will be inverted. With a known input snr, and thus known chip error probability, the output pdf is Binomially distributed about the mean values $m(1-p_e)$ for a data 1 received, and about

mp_e for a symbol 0 received, figure 2-11. The pdfs of figure 2-11 are "mirror-images" about the mid-point $m/2$. This is simply due to the definitions designating a symbol 1 or 0 received.

If a symbol 1 is received but the pdf calculated on the number of chip disagreements rather than agreements then the resulting distribution is identical to that for a symbol 0 received but calculated on the basis of the number of chip agreements. Clearly the average ordinate or mid-point of these two, figure 2-11, is the value $m/2$, irrespective of the values of m or p_e . Now since the mean values are $m(1-p_e)$ and mp_e it can be seen that as p_e is reduced the distributions move away from the mid-point. As p_e is increased towards 0.5 the distributions move towards the mid-point, figures 2-11a, 2-11b, 2-11c. To determine if a symbol 1 or 0 is present in the on-line register we need a measure of discrimination between the two cases. To do this we examine the cumulative distribution of (2-27) rather than the discrete distribution of (2-24).

This is illustrated in figure 2-12 for a 511 bit DMF with chip error probability of 0.42 ($E_c/N_o = -17$ dB for BPSK modulation). The results show that for either symbol in the on-line register the cumulative probabilities rapidly approach zero, above a given ordinate for the in phase case, and below a given ordinate for the out of phase case. Further since the discrete pdfs are known to be symmetrical with corresponding amplitudes spaced a distance $m(0.5-p_e)$ from the mid-point $m/2$ the cumulative probabilities must also be symmetrically identical, with corresponding amplitudes spaced at $2m(0.5-p_e)$. Clearly if we know that the current DMF output corresponds to a symbol alignment (either symbol 1 or symbol 0) then it is only necessary to decide if the DMF output is greater than or less than $m/2$ to make the optimum detection estimate of a symbol 1 or 0. This will give minimum symbol error probability in the presence of chip errors, assuming equally likely symbols. Whilst this result may appear

obvious it can only be applied to the case where the symbol in the on-line register is known to be in alignment with the reference symbol, that is the symbol decision instant is code-synchronized. It will be shown that this has serious implications for the available processing gain available from the DMF during synchronization.

2.3.4 DMF OUTPUT - CODE OUT OF PHASE

For a binary code of m elements there are $m-1$ out of phase conditions between on-line and reference registers. In the absence of chip errors the DMF output will be determined by the out of phase autocorrelation function of the code. Values for some codes have been evaluated [GOLD 1967], [GOLOMB 1967], [BEALE 1982], but for the sake of clarity the discussion will be restricted at this stage to codes with uniform out of phase autocorrelation properties, figure 2-13. In particular the codes considered (m -sequences) have a uniform out of phase autocorrelation value of -1 , that is there is one more disagreement than agreement between the on-line and reference registers. This assumes a chip error probability of 0.

For a practical m -sequence of length m (where $m=2^K-1$, K is an integer) the error free out of phase condition will result in $(m+1)/2$ disagreements and $(m-1)/2$ agreements between on-line and reference registers. For a given value of p_e the pdf corresponding to the out of phase condition is similar in shape to that for the in phase condition. However the pdf is centred about the out of phase autocorrelation value, that is the shape of the pdf is determined by the value of p_e but the mean is always equal to the out of phase autocorrelation value. The mean can be evaluated by considering the length m code to consist of a pair of sub-sequences, sequence x of length $(m+1)/2$ corresponding to disagreements, and sequence y of length $(m-1)/2$ corresponding to agreements. Since the chips are assumed independent the sequence pair can be considered

independently on a statistical basis. Therefore considering the mean of each sequence we have from (2-45)

$$\text{Mean number of agreements in } x = (m+1) p_e / 2 \quad (2-48)$$

$$\text{Mean number of agreements in } y = (m-1) (1-p_e) / 2 \quad (2-49)$$

since p_e is common to both sequences.

The resulting mean is found by addition giving

$$\begin{aligned} \text{Mean output} &= (m-1) (1-p_e) / 2 + (m+1) p_e / 2 \\ &= (m-1) / 2 + p_e \\ &\approx m / 2 \end{aligned} \quad (2-50)$$

since $p_e \approx 0.5$.

The pdfs corresponding to in phase and out of phase conditions are shown in figures 2-15a to 2-15c for lowering snr values. The out of phase case appears as the centre distribution in 2-15a to 2-15c, the in phase case with code true appears to the right of the out of phase case, and for comparison the in phase case with code inversion appears to the left of the out of phase case. This shows that the pdfs are similar in shape but does not take into account the $m-1$ occurrences of the out of phase condition for each in phase condition. From the above it can be seen that if a symbol decision is made on the DMF output when the code is out of phase and with a decision threshold of $m/2$ then the symbol error probability will be approximately 0.5. This is the case for all values of p_e , assuming that m is much greater

than p_e .

Finally for the out of phase condition the variance is given by

$$\text{Total variance} = 1/2 (\text{Variance}(x) + \text{Variance}(y)) \quad (2-51)$$

which from (2-47) gives

$$\begin{aligned} \text{Total variance} &= 1/2 ((m-1)p_e q_e + (m+1)p_e q_e) \\ &= m p_e q_e \end{aligned} \quad (2-52)$$

where $q_e = 1-p_e$.

Hence the variance is the same as for the in phase case.

2.3.5 NOISE ONLY

If the input snr to the hard limiter of figure 2-9 is reduced to 0 (noise only) then from (2-22) the DMF output pdf is

$$\text{Pr}(\text{Output}=K) = C \frac{(0.5)^m}{K} \quad (2-53)$$

since $p_e = q_e = 0.5$

Clearly this is not changed by the code phase since the signal is zero, or by the noise amplitude providing this is greater than zero but not greater than would be acceptable

in a practical implementation. The mean value of this distribution is again $m(1-p_e)=m/2$, as for the out of phase case. Similarly the variance is mp_eq_e . The DMF output pdf for the noise only case is shown in figure 2-14. This does not vary with absolute noise level, within practical limits, provided the signal level is 0, since the resulting chip error probability will always be 0.5.

2.4 DETECTION CRITERION

2.4.1 CODE SELECTION

From the previous discussions and diagrams it can be seen that a knowledge of the time at which the code is in phase (code epoch) is necessary for the correct sampling of the DMF output, to minimize the symbol error probability, and hence optimize data demodulation. For the code out of phase condition the cumulative distribution is similar to that for the noise only case. The discrete probability distribution for the in phase case is approximately centred about the value $m(1-p_e)$ whilst that for the noise only case is centred about the value $m/2$. If a code is selected which has an out of phase autocorrelation value close to $m/2$ then the out of phase discrete probability distribution is also centred about the value $m/2$. Such codes are often termed 'pseudonoise' codes.

2.4.2 EFFECT OF MODULATION

2.4.2.1 SEQUENCE INVERSION KEYING

We now consider the effect of SIK modulation on the DMF output probability distribution. It has been shown that for a sequence inversion the DMF output distribution is reversed. For the out of phase condition the probability of correctly detecting a symbol 1 or 0 will be identical if

the decision threshold is set to the out of phase autocorrelation value, assuming equally likely symbols. For the in phase case the optimum threshold value is that which minimizes the probability of symbol errors under the same conditions. It can be seen from figures 2-11 and 2-12 that this must be the value $m/2$ by symmetry. Therefore to give an equal symbol error probability for the in phase case (regardless of the symbol) we must choose a symbol decision threshold at the DMF output of $m/2$. Furthermore to produce identical cumulative probabilities for the out of phase condition we must choose a code with an out of phase correlation value of $m/2$, for all code phases. In practice m may be an odd number and it may be necessary to choose a code with an out of phase autocorrelation value of $(m-1)/2$. The autocorrelation function of such a code, figure 2-13, is optimum in the sense that it maximizes the distance in DMF output between code in phase and code out of phase values in the absence of noise, and with noise present it is still optimum in the sense that it maximises the ratio of in phase to out of phase autocorrelation values for any particular DMF output K whilst still accounting for the two possible in phase cases due to modulation. For SIK modulation it is essential that such a code be used in order to obtain the maximum processing gain from the DMF.

2.4.2.2 ON-OFF KEYING

We now turn our attention to an alternative modulation scheme, ON-OFF keying (OOK), which may be useful in certain applications. With ON-OFF keying a sequence is transmitted to represent a symbol 1 and no sequence (no energy transmission) represents a symbol 0. If we assume coherent RF demodulation but assume an arbitrary phase offset of 0 or 180 degrees then the in phase probability distribution will be identical to the in phase case for SIK modulation as discussed. As we have no prior knowledge of the phase offset and assuming that phase offsets of 0 or 180 degrees are equally likely then the probability that the DMF output

probability distribution has mean $m(1-p_e)$ is equal to the probability that the output probability is equal to mp_e . Both cases, therefore, would need to be interpreted as receiving a symbol 1. For a symbol 0 received the DMF output probability distribution would be identical to that for the noise only case. The three probability distributions are shown in figures 2-16a to 2-16c for comparison with figures 2-15a to 2-15c, using the same error probabilities for the code in phase case in corresponding diagrams in 2-16 as used in 2-15. Again the in phase case with code true appears as the rightmost distribution for each value of snr, the in phase case with code inverted as the leftmost distribution, but the out of phase case has been substituted by the noise only case and appears as the centre distribution for each value of snr.

For the OOK case the optimum decision threshold is not $m/2$, but somewhere between $m/2$ and m , or $m/2$ and 0. Thus two decision thresholds must be accounted for and it is apparent that exceeding the upper threshold or failure to exceed the lower threshold defines a symbol one received, and failure to exceed the upper combined with exceedance of the lower threshold defines a symbol zero received, at the code epoch in the case of a symbol one received, and at an assumed code epoch (no signal received) in the case of a symbol zero received. Therefore the maximum distance between decisions of symbol ones and zeros has been drastically reduced and hence the sensitivity or processing gain of the DMF will also be lost.

A first order estimate of a possible choice of decision threshold is now given, the argument assumes a symbol 1 received as a positive DMF output.

For low snrs the DMF output pdfs for noise only and code in phase conditions are assumed to be similar in shape, only differing by their respective mean values. If the pdfs are similar then the cumulative pdfs will also be similar in shape, again only differing by the mean values

at which the respective curves reach the value 0.5. Therefore the probability of making a symbol decision error for the noise only case is the probability that the DMF output exceeds some threshold x , and for the symbol 1 case the error probability is the probability that the DMF output will be equal to or less than the threshold x . The optimum choice of threshold must be that which makes these two error probabilities equal, and assuming the DMF pdfs curves are equal in shape for noise and symbol 1 cases, then the value x will be the mid-point between the two pdfs. For noise only the mean DMF output was shown to be $m/2$, and for the code in phase condition it was shown to be $m(1-p_e)$.

Therefore the threshold x is the mean of these,

$$\text{Threshold} = m(3/4 - p_e/2) \quad (2-54)$$

Similarly it can be shown that if the symbol 1 is received such as to cause a negative DMF output then the threshold value should be set to

$$\text{Threshold} = m(1/4 + p_e/2) \quad (2-55)$$

This result highlights two serious problems with this type of modulation scheme. Firstly the choice of threshold is dependent upon the value of p_e and hence the received snr, this is unlikely to be known by the receiver prior to synchronization. Secondly the distance between deciding a symbol 1 and a symbol 0 received has been reduced by a factor of 2 and the resultant symbol error rate increased accordingly.

The problem is examined further in chapter 4 where the problem is addressed in terms of the performance of the DMF during code synchronization acquisition.

The main difference between the effect of OOK and SIK modulation on the DMF output probabilities has been introduced. Clearly if SIK modulation is used it is essential to maintain phase coherence between the transmitter and receiver local oscillators. If OOK modulation is used this condition can be relaxed to some extent at the expense of system sensitivity.

2.5 EFFECT OF QUANTIZATION ON DMF PERFORMANCE

The analysis of DMF performance for various quantization levels has been performed by several authors, e.g. [BRUNO 1970] for multilevel quantization, [TURIN 1976] and [BAIER and BAIER 1984] for single level quantization. An alternative technique is developed here, based on the statistics of the input samples. The samples are considered to belong to a sufficiently large set such that the probability of occupancy of a particular quantization level is determined solely by the probability density function of the input signal plus noise. The mean and variance of the quantizer output are then determined. The loss due to quantization can then be found by comparing these with the mean and variance at the output of the ideal matched filter.

It was shown in Chapter 1 that under ideal conditions and using the appropriate matched filter the processing gain available to a spread spectrum receiver is $10\log(m)$ dBs where m is the code length. We now consider the loss of processing gain due to quantization, since the DMF is an approximation to the ideal matched filter.

From the definition of processing gain

$$PG = (\text{mean snr})_{\text{output}} / (\text{mean snr})_{\text{input}} \quad (2-56)$$

it can be seen that we must examine the quantizer output statistics before the mean and variance can be determined. These will be independent of the DMF length, m , provided m is sufficiently large such that the quantizer input probability distribution can be considered to be equal to the expected or average probability distribution. This condition is also necessary when considering chip error probabilities since these are also assumed to be average error probabilities.

We begin by defining the mean and variance as in (2-42), (2-46) where

$$\text{Mean}(k) = \sum_{k=1}^L k p(k) \quad (2-57)$$

$$\text{Var}(k) = \sum_{k=1}^L k^2 p(k) \quad (2-58)$$

for an event with L outcomes, and probability of occurrence $p(k)$ for each unique outcome. For an n bit quantizer there are 2^n unique output states, giving $L = 2^n$.

Now if uniform and bipolar quantization (table 2-2) is assumed (2-57) and (2-58) cannot be applied directly since positive quantizer outputs are in the range 0 to $n-1$, and negative outputs are in the range -1 to $-n$. However without loss of generality the quantizer outputs can be assigned a new 'index' for k and an average weight, W_k , associated with each index. The new index is incorporated in table 2-2 for clarity.

Thus the mean may be written as

$$\text{Mean}(k) = \sum_{\substack{k=-n \\ k \neq 0}}^n W_k p(k) \quad (2-59)$$

$$= \sum_{k=1}^n W_k p(k) + W_{-k} p(-k) \quad (2-60)$$

and putting $V(k) = (W(k) - \text{Mean})$ the variance becomes

$$\text{Var}(k) = \sum_{k=1}^n (V_k)^2 p(k) + (V_{-k})^2 p(-k) \quad (2-61)$$

It now only remains to formulate the error probabilities $p(k)$. From figures 2-7 and 2-8 it can be seen that, assuming a rectangular pulse in AWGN, it is only

necessary to evaluate the normal curve at the appropriate quantization levels and by subtraction the average error probabilities can be found for each interval. However the probability associated with a given quantization interval for a given input snr is clearly determined by the choice of interval width, or quantization threshold, V_i . Therefore, using (2-60) and (2-61), the expression (2-56) has been evaluated for a range of quantization levels, figure 2-17, relative to a constant input snr of -17 dBs. Examining figure 2-17 in detail shows that a constant loss of almost 2 dBs can be expected from single-bit quantization (hard-limiting), whereas for 2 or more levels of quantization a minimum loss can be achieved for a particular choice of quantization threshold, table 2-3.

TABLE 2-2
Voltage transfer function for 3-bit quantizer
Uniform input and output quantization

Input Voltage (V_{in}) / Volts	Output	Index k	Weight W_k
$V_{in} > 3V_t$	3	4	3.5
$3V_t > V_{in} > 2V_t$	2	3	2.5
$2V_t > V_{in} > V_t$	1	2	1.5
$V_t > V_{in} > 0$	0	1	0.5
$0 > V_{in} > -V_t$	-1	-1	-0.5
$-2V_t > V_{in} > -2V_t$	-2	-2	-1.5
$-3V_t > V_{in} > -3V_t$	-3	-3	-2.5
$-4V_t > V_{in}$	-4	-4	-3.5

TABLE 2-3
Minimum and maximum loss due to multibit quantization

Quantization level/bits	Minimum loss/dB	Maximum loss/dB
1	1.98	1.98
2	0.78	1.98
3	0.30	1.98
4	0.09	1.98

This shows that a loss of only 2 dBs is incurred by the hard limiter based DMF, whereas a very good approximation to the ideal matched filter can be achieved with quantization of only 3 or 4 levels. The penalty of multibit quantization is increased DMF complexity and cost. Furthermore to obtain the minimum loss of table 2-2 the optimum quantization threshold must be known at the quantizer. This assumes that the input snr is known.

In general, during synchronization acquisition the received snr is unlikely to be known and the potential benefit of multibit quantization may not be realised in practise. For the single bit DMF the quantization loss is constant, and therefore independent of received snr. Figure 2-17 shows that the loss for multibit quantization at $V_t = 0$ is identical to that for the hard-limiter, regardless of the number of quantization levels. This can be seen intuitively since as V_t approaches 0 the width of the quantization interval approaches 0, and hence the probability of occupancy also approaches 0. Therefore the only significant contribution to the mean and variance of the quantizer output in (2-60) and (2-61) respectively is for $k = +n$ or $k = -n$, and these values correspond to outputs from the most significant bit. Clearly this is equivalent to the hard-limiter case and accounts for the curves of figure 2-17 approaching 2 dBs as V_t approaches 0. Similarly for low snrs it can be seen that as V_t becomes large relative to V_{in} then contributions to the quantizer output mean and variance can only be due to the least significant bit, since this is the only likely level of occupancy. Again this approximates the hard-limiter case, and explains why the curves of figure 2-17 approach 2 dBs as V_t becomes large relative to V_{in} .

In chapter 3 it is shown that the received snr can be measured by a complex baseband receiver based on single-bit DMFs. This could prove useful in a multibit DMF receiver where such information could be used to set up optimum quantization levels and hence allow the multilevel DMF to

operate under optimum conditions. In such a scheme the quantization threshold could be set to zero, allowing the DMF to operate in single bit mode. The received snr could then be measured, and the optimum quantization threshold determined from this. The DMF based on multibit quantization could then be operated under optimum conditions.

Finally from (2-60) we have for the hard-limiter

$$\text{Mean(output)} = 1 p(1) - 1 p(-1) \quad (2-62)$$

$$= 1 (1-p_e) - 1 (p_e)$$

$$= 1 - 2 p_e \quad (2-63)$$

and from (2-20) we have for low snr

$$p_e \approx 1/2 (1 - (2/\pi)^{1/2} v_{in}/\sigma_{in})$$

giving

$$\text{Mean(output)} \approx (2/\pi)^{1/2} v_{in}/\sigma_{in} \quad (2-64)$$

Therefore

$$(\text{Mean snr})_{in}/(\text{Mean snr})_{out} = (2/\pi)^{1/2} \quad (2-65)$$

since the output variance from (2-61) equates to 1.

In dB notation the loss is $10\log(2/\pi) = 1.96$ dB. This is in agreement with the figures derived by other authors [BRUNO 1970], [TURIN 1976] and in close agreement with those of table 2-2. It should be noted that the analysis given assumes uniform quantization steps on both quantizer input and output. However it is possible to assign non-uniform quantizer input thresholds and different output 'weights' accordingly. The analysis given can be extended to the case of non-uniform quantization. By evaluating the terms $p(k)$ for the non-uniform input intervals, and using these with the output weights the quantizer output mean and variances can be found. However it has been shown [BRUNO 1970] that little benefit is obtained by using non-uniform quantizer intervals, and the penalty is increased receiver complexity.

2.6 CONCLUSIONS

The operation of the DMF has been studied in detail. The number of mathematical operations required in the DMF has been investigated. It has been shown that an attempt to reduce the number of correlation operations results in the need for a more complex system. This is because the number of correlation functions is balanced against the number of adders necessary to calculate the final correlation value. In practical systems a DMF structure is chosen whereby the majority of the addition functions are reduced to the simple Exclusive-NOR function.

The DMF output statistics have been examined. Approximations to these have been derived and will be used in the following chapters. The DMF output mean and variances have been evaluated and illustrated for various snrs. It was shown that the correct choice of code autocorrelation function will maximise the ratio of code in phase DMF output to code out of phase DMF output for SIK modulation. This led to the determination of the optimum code in phase decision threshold at the DMF output for SIK

modulation in a statistical sense. An alternative modulation scheme, OOK, was introduced in the context of DSSS systems, and an approximate threshold for optimum detection evaluated. It was shown that this type of modulation removed the need for carrier phase coherence at the receiver but the penalty was reduced sensitivity and ambiguity over the optimum decision threshold for an unknown input snr.

The loss due to quantization relative to an ideal matched filter was found by evaluating the mean and variances of the quantizer output statistics. The results obtained are in close agreement with those obtained by other authors but the technique is readily extended to the cases of non-white input noise, interfering signals, and distortion. Under these conditions it is necessary to evaluate the probability of occupance for each quantizer interval and from these the quantizer output mean and variances can be found.

REFERENCES

- BEALE M. (1982) 'Direct-Sequence Spread-Spectrum Multiple-access Systems', PhD Thesis, University of Kent.
- BAIER A. and BAIER P.W. (1984) 'Digital matched filtering of arbitrary spread-spectrum waveforms using correlators with binary quantization', Proceedings of the IEEE Military Communications Conference, April 1984.
- BRUNO F.J. (1970) 'Maximum likelihood quantizer design for Digital Matched Filter Receivers', Proceedings International Communications Conference.
- CARLSON A.B. (1975) 'Communication Systems', McGraw-Hill, 2nd edition.
- CHRISTIE R. (1986) 'Implementing Matched Filters in Direct Sequence Spread Spectrum Systems using the MA7170 Bit-slice Correlator, Application note, Marconi Electronic Devices Ltd, Doddington Road, Lincoln, England.
- GOLD R. (1967) 'Optimal binary sequences for Spread Spectrum multiplexing', IEEE Transactions on Information Theory, Volume IT-13, October 1967, pp619 - 621.
- GOLOMB S.W. (1967) 'Shift register sequences', Holden-Day.
- HOLMES J.K. (1982) 'Coherent spread spectrum systems', John Wiley and Sons.
- MEDL (1986) 'MA7170 Bit-slice Correlator', Application note, Marconi Electronic Devices Ltd, Doddington Road, Lincoln, England.
- PAPOULIS A. (1984) 'Probability, Random Variables, and

Stochastic Processes', McGraw-Hill, 2nd edition.

PFEIFFER P.E. and SCHUM D.A. (1973) 'Introduction to Applied Probability', Academic Press.

STREMLER F.G. (1977) 'Introduction to Communication systems' Addison-Wesley Publishing Company.

TRW (1981) 'TDC1023J Digital Output Correlator', Application note, TRW LSI products, P.O. Box 2472, La Jolla, California, 92038, USA.

TURIN G.L. (1976) 'An Introduction to Digital Matched Filters', Proceedings of the IEEE, Volume 64, Number 7, pp 1092 - 1112.

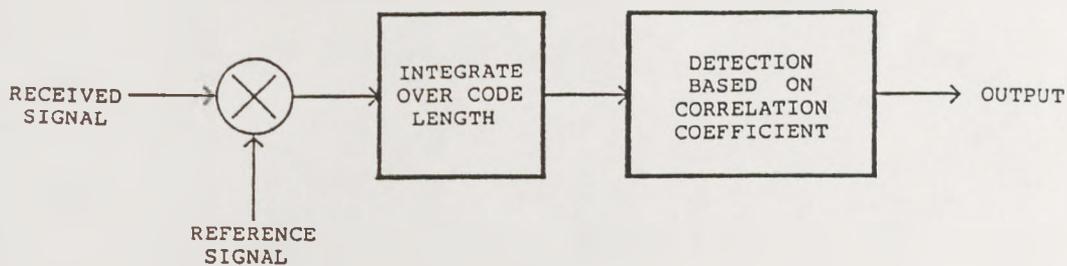


Figure 2-1. Correlation Detector.

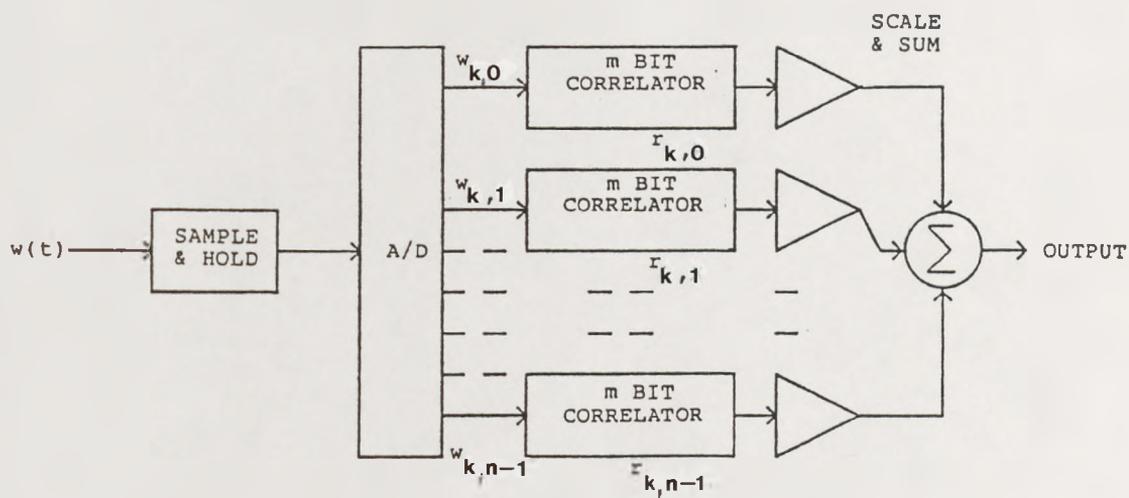


Figure 2-2. m by n bit Digital Correlator.

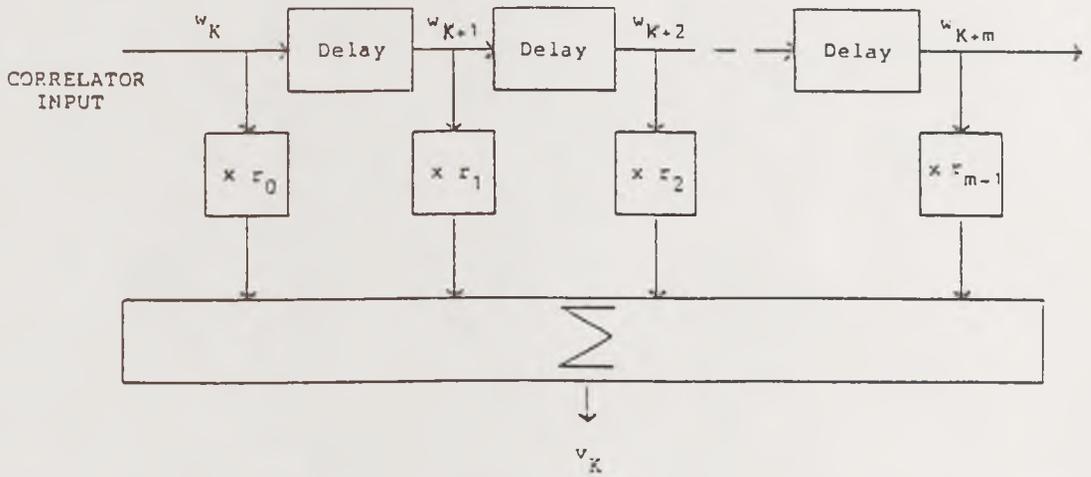


Figure 2-3. Canonical form of Binary Correlator

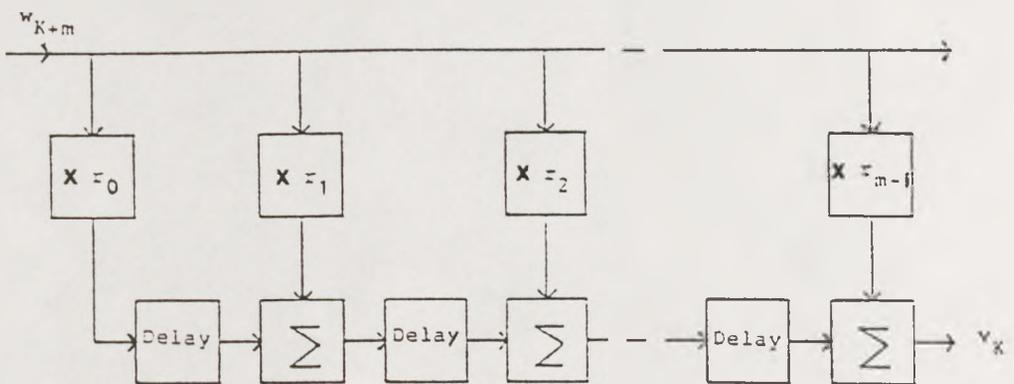


Figure 2-4. Binary Correlator with pipelined adders

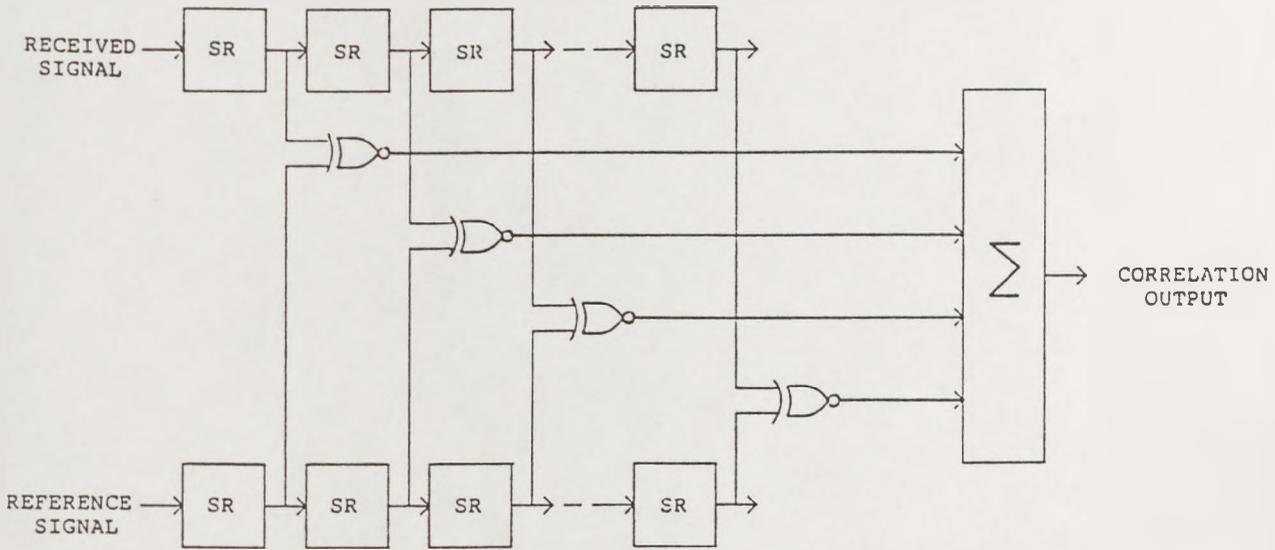


Figure 2-5. Binary Correlator Implementation.

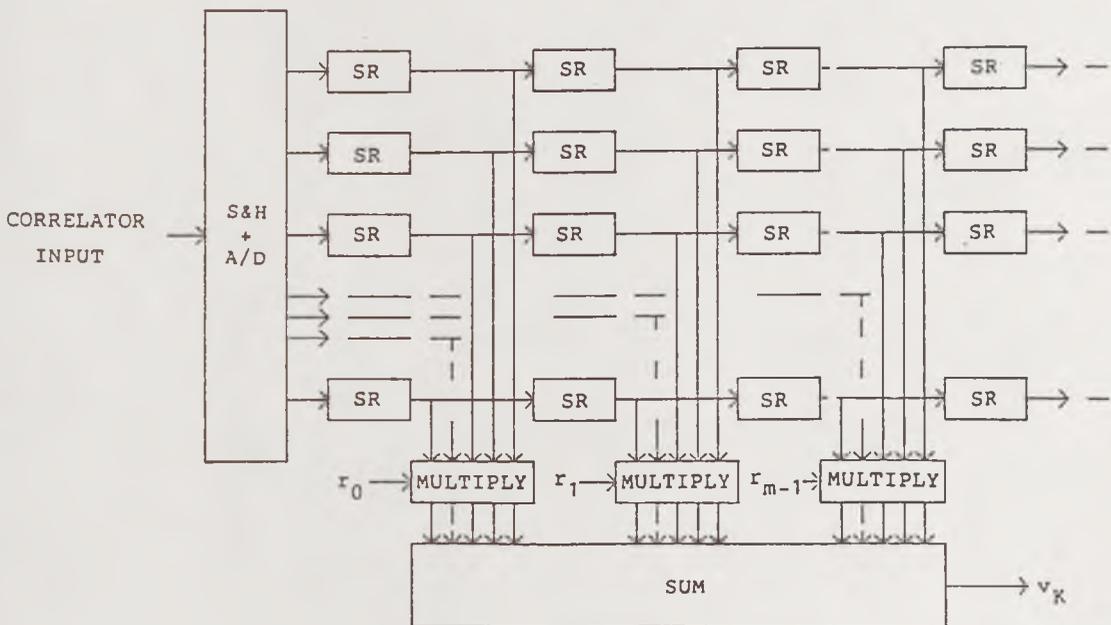


Figure 2-6. Implementation of Binary Correlator from expression (2-10).

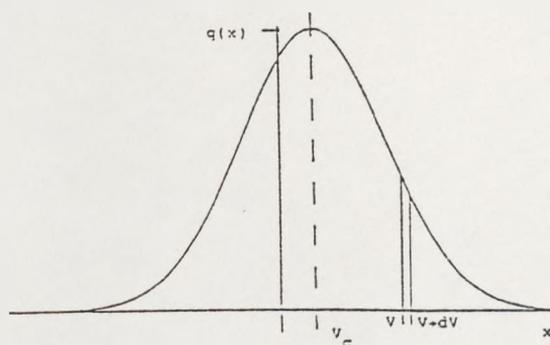


Figure 2-7. Gaussian probability density function.

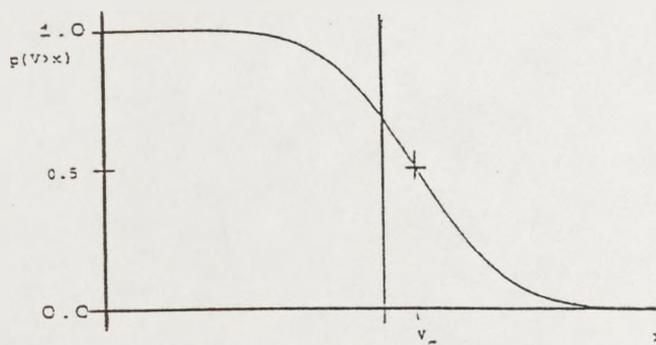


Figure 2-8. Cumulative Gaussian probability function.

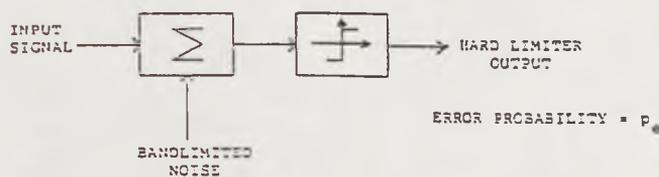


Figure 2-9. Hard-limiter.

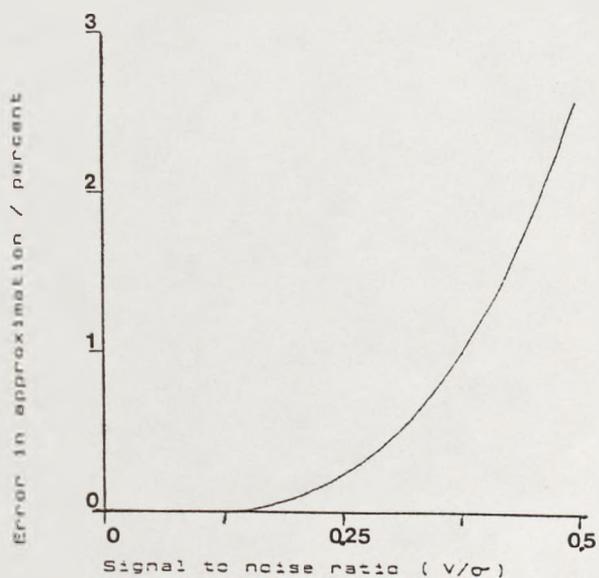


Figure 2-10. Error due to approximation (2-21)

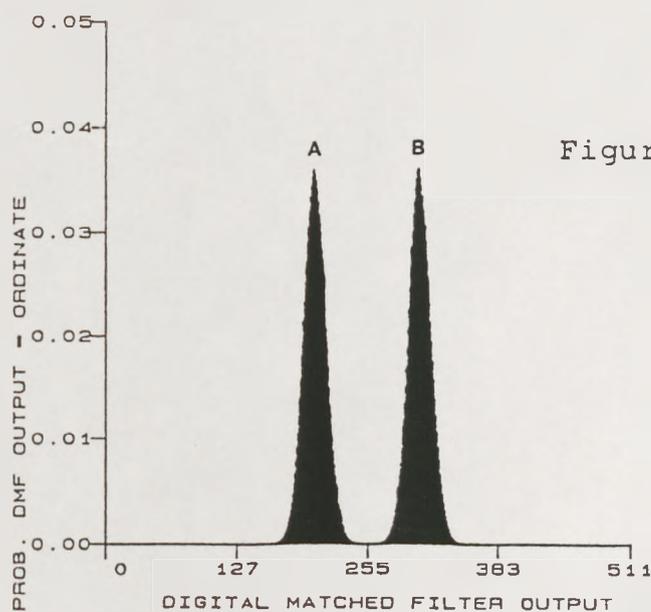
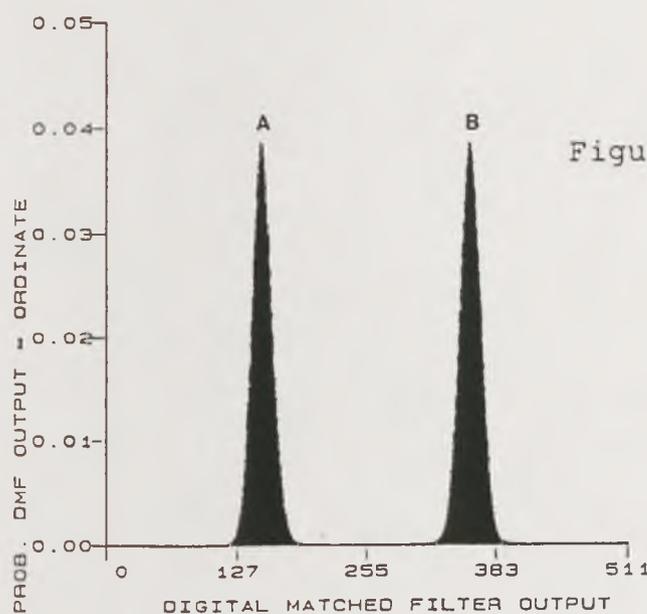
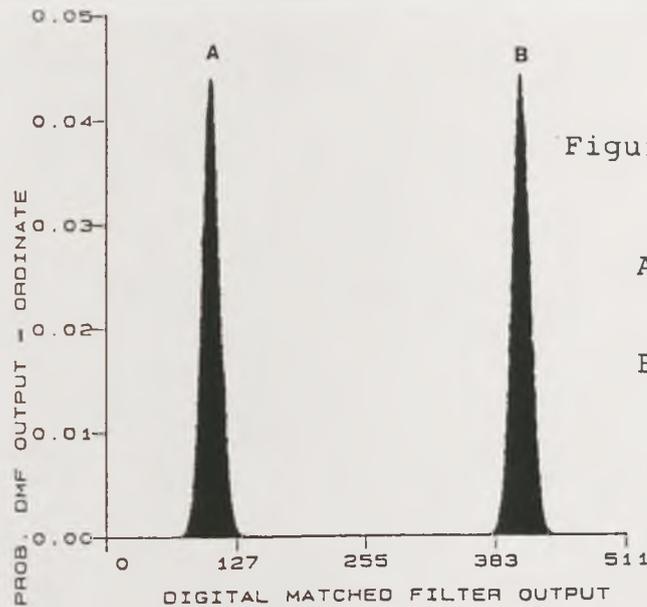


Figure 2-11. DMF output PDFs.

Figure 2-12a. Code in phase.

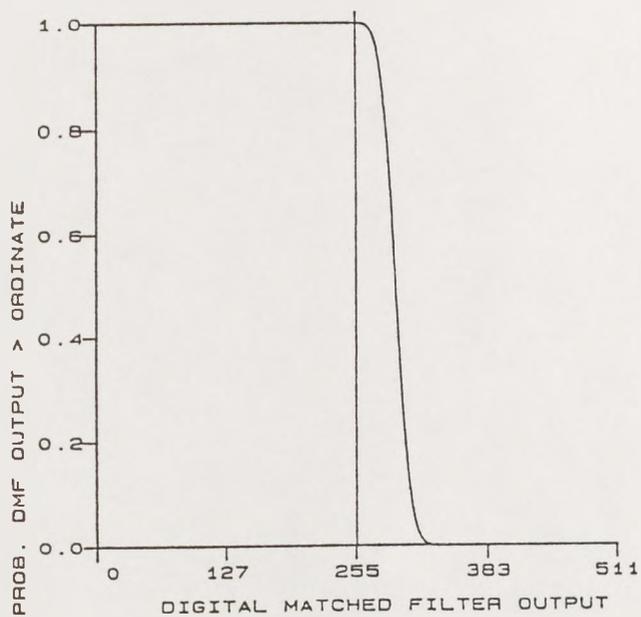


Figure 2-12b. Code in phase inverted.

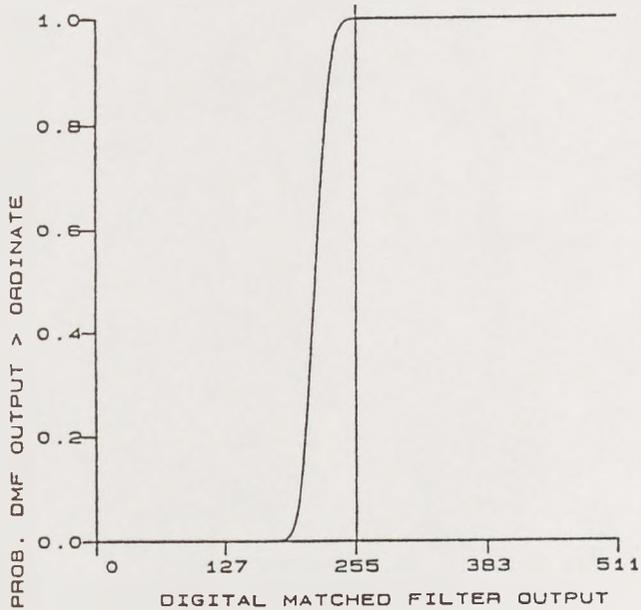


Figure 2-12. Cumulative PDF of DMF output.

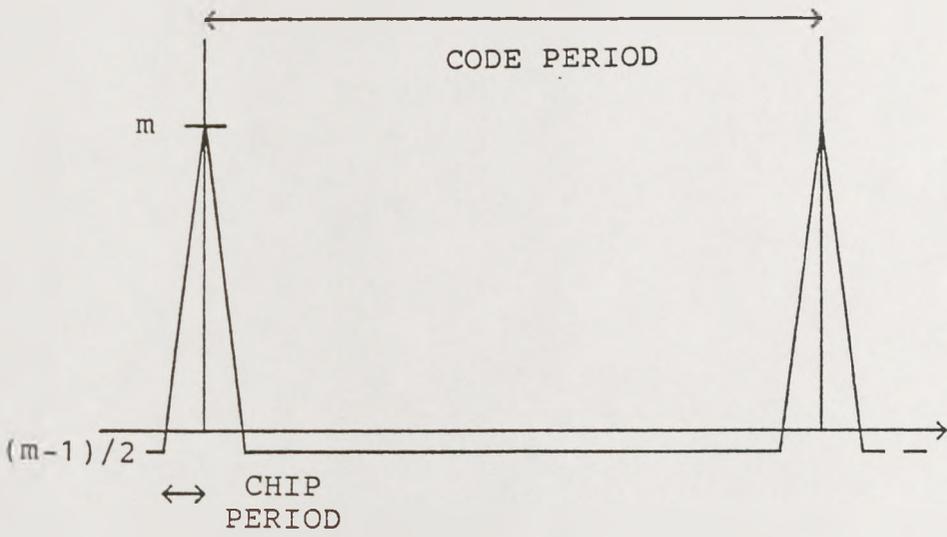


Figure 2-13. Autocorrelation function of practical code for optimum detection.

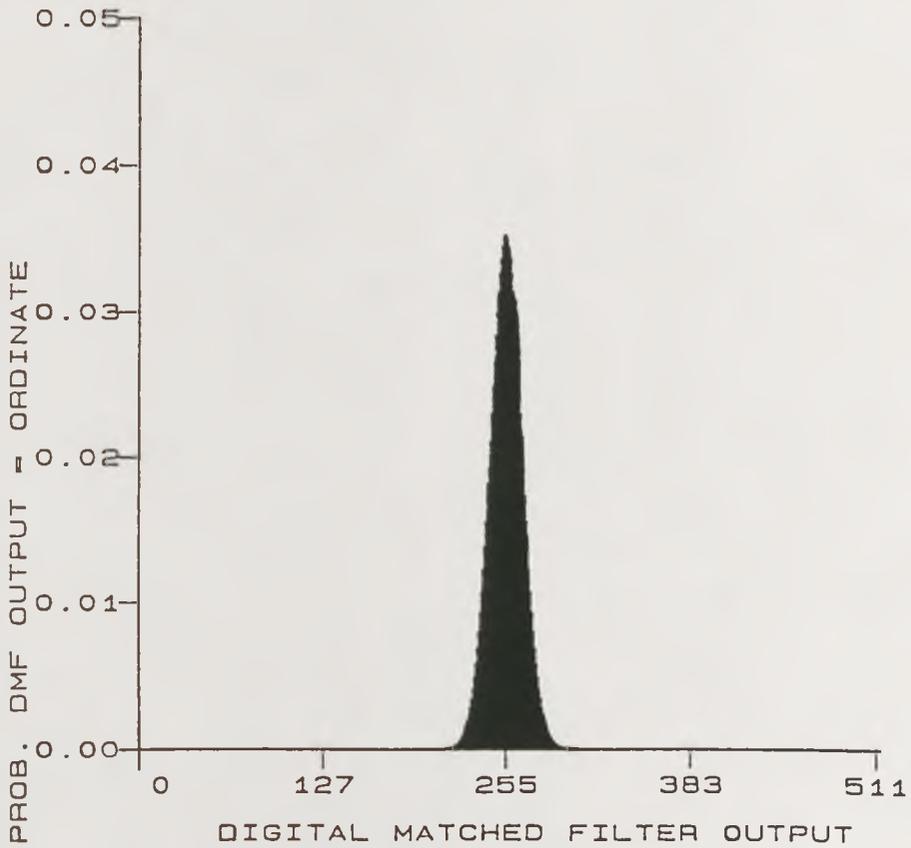


Figure 2-14. DMF output PDF, noise only case.

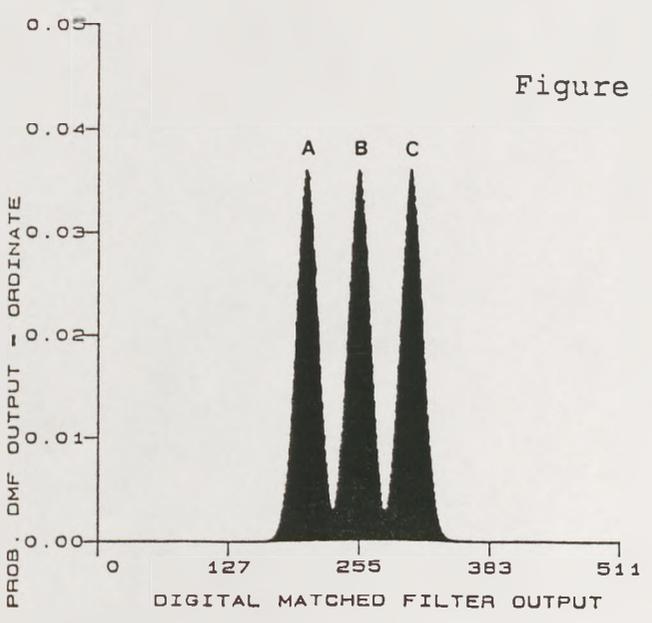
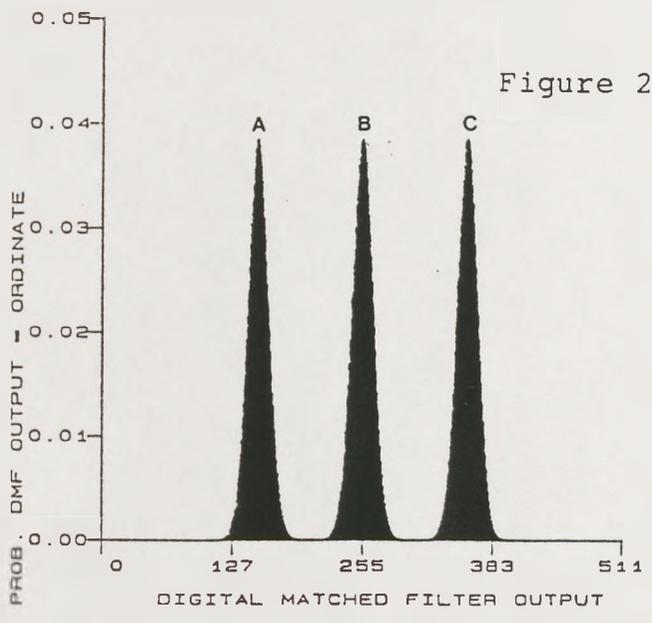
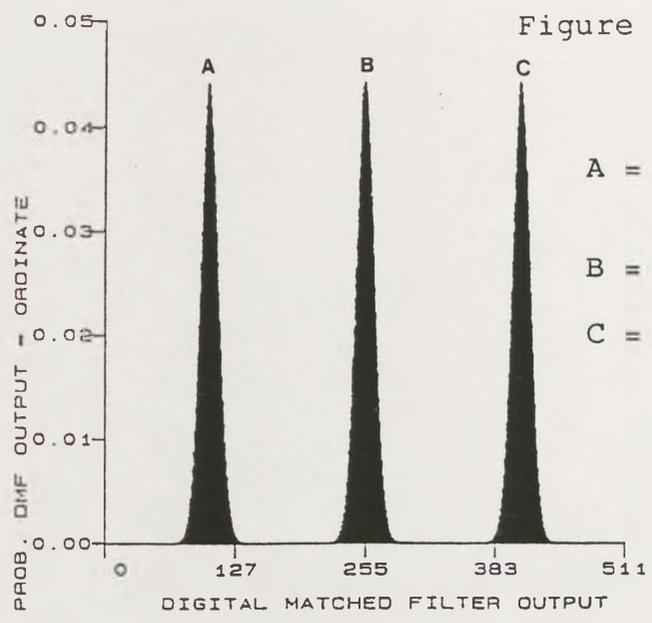


Figure 2-15. DMF output PDF for code in phase and code out of phase cases.

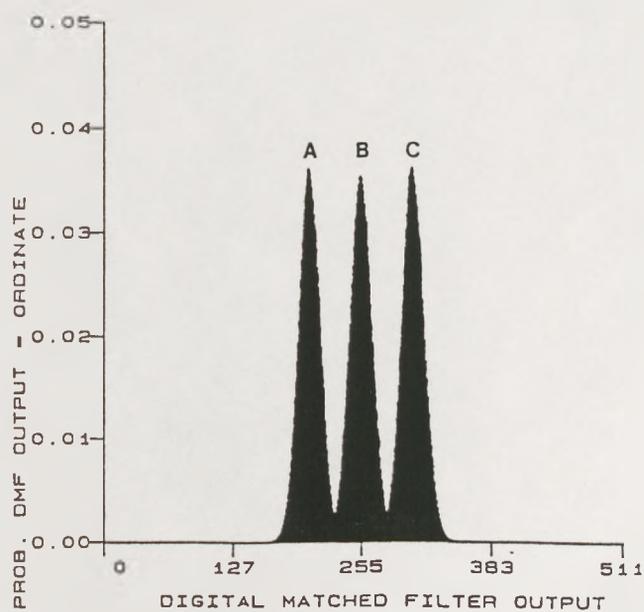
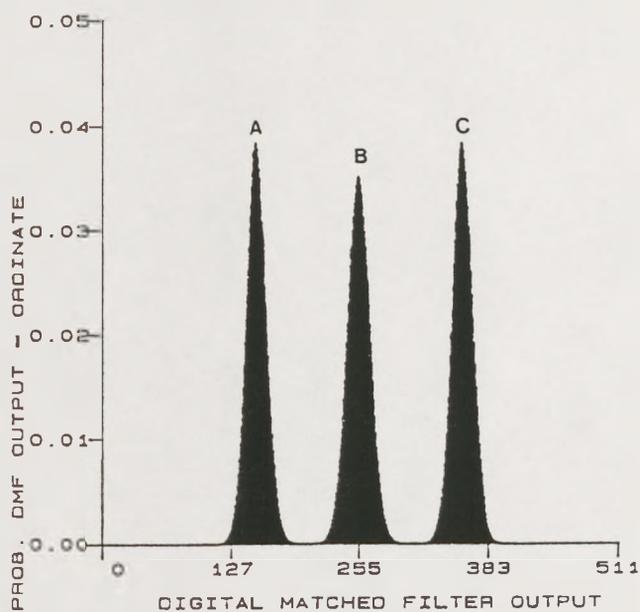
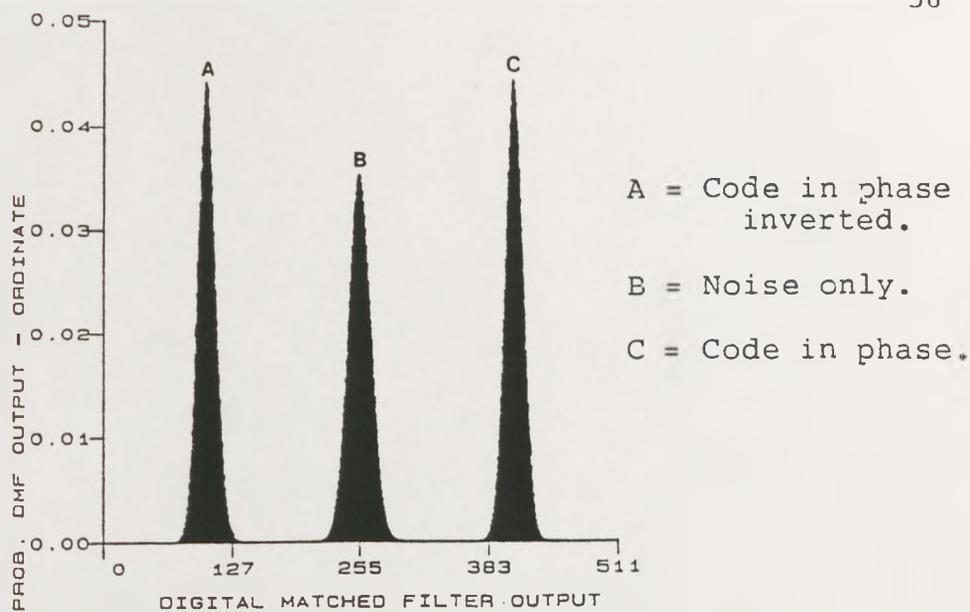


Figure 2-16. DMF output PDF for code in phase and noise only cases.

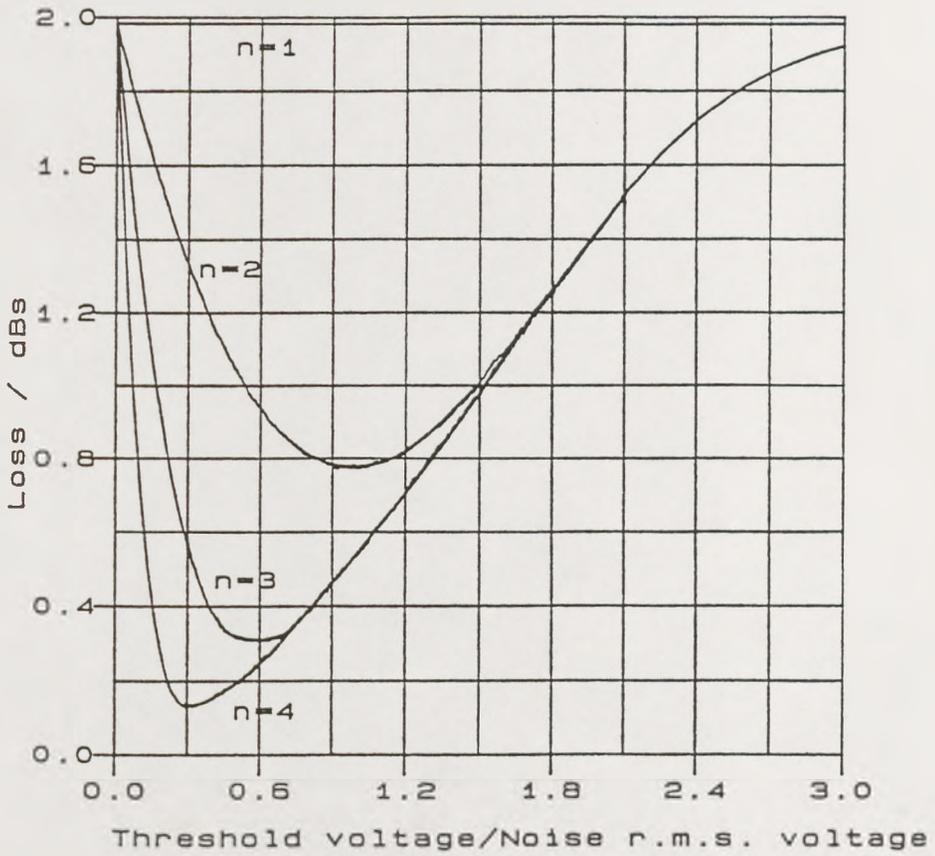


Figure 2-17. Effect of quantization threshold on processing gain loss for constant input snr.

CHAPTER 3

COMPLEX BASEBAND RECEIVER

3.1 INTRODUCTION

Due to speed restrictions of present digital matched filtering or correlation devices [CHRISTIE 1986] detection at baseband is necessary for DMF based DSSS receivers. Since the desired signal may be received in the presence of noise, other DSSS signals, or interfering signals, information regarding carrier phase coherence may not be available to the receiver prior to matched filtering [SKAUG and HJELMSTAD 1985]. The potential processing gain of the receiver is therefore required to obtain this information. Hence it is necessary to consider a complex baseband receiver structure, figure 3-1, with in phase (I) and quadrature phase (Q) baseband channels. The necessity for the Q channel can be seen by considering a single channel receiver. If the receiver local oscillator is frequency coherent with the received signal but in phase quadrature the lowpass filtered mixer output snr will be zero irrespective of the input snr. At first the complex baseband receiver may appear an inefficient solution since it introduces the need for an extra channel, and hence almost twice as much hardware compared to a phase coherent receiver. However, this approach is not unique to DSSS receivers and is, in general, the basis of many baseband digital receivers.

The function of the complex baseband receiver is to convert the received DSSS signal at RF to a baseband digital signal representing the receiver's estimate of the transmitted DSSS code. The important processes of symbol synchronization and detection, RF recovery and tracking, and data demodulation will be covered separately in chapter 4.

Each element of figure 3-1 will be described and analysed in detail. This is necessary since the overall performance of the system depends on the losses introduced by the various system elements.

It will be shown that the mixing, or down conversion process, is essentially ideal. Also the choice of optimum lowpass filter depends upon the DSSS application, since consideration must be given to stopband rejection, filter step response, and efficiency.

Several practical filter responses are considered. For these filters the correlation of lowpass noise and signal samples is evaluated. The results indicate that it is not possible to simultaneously obtain zero correlation of lowpass noise and signal samples. Therefore it is necessary to expect a degree of 'memory' between noise samples or a level of ISI between signal samples.

The efficiency of the practical filters relative to an ideal chip matched filter is evaluated. For each filter there is an optimum choice of cutoff frequency which minimizes the loss expected for an input pulse of given duration. Whilst this approach is most suitable for examining the loss expected in radar signalling it is also a useful concept in DSSS applications. Attempting to match the lowpass filter to the DSSS code will result in the derivation of the code matched filter characteristic. Since the DMF is a suitable approximation to this it is clear that the role of the lowpass filter is to maximize the snr obtainable from each chip, and is therefore a close approximation to the chip matched filter.

The practical filters are compared on the basis of risetimes, since long risetimes lead to reduced sampling snr if the optimum sampling instant is unknown. This is likely to be the case for the DSSS receiver operating prior to synchronization.

The effect of a non-coherent carrier is considered. It is shown that a preferred relationship exists between the RF carrier frequency and the baseband sampling frequency. This leads to the possibility of synchronizing the periodic variation in lowpass filter outputs due to carrier frequency offsets with the variation in sampling snr obtained due to the lowpass filter risetime pattern.

Two algorithms for combining the outputs of the I and Q channel DMF outputs are considered. For each algorithm the output mean and variances are evaluated and the effects of changing snr and carrier phase are investigated.

3.2 RECEIVER MIXERS

Referring to figure 3-1 it can be seen that the two arms of the complex baseband receiver are identical except for a phase offset of 90 degrees between the carrier input of the Q channel mixer relative to the I channel mixer. It is reasonable to assume that the mixers are ideal in operation. Distortion can be minimized since automatic gain control (AGC) can be used to ensure the mixers are not driven into saturation by the combination of desired signal and noise. The AGC can also be used to ensure a sufficient drive level into the mixers, although it is clear that this may consist mainly of noise in the DSSS signal. However if the signal level is too low relative to the noise level, such that mixer non-linearities or noise figure problems arise, it is likely that the signal could not be recovered from the noise in the practical system to be described, since there will be insufficient processing gain. The important criterion would then be the dynamic range of the mixer which for practical devices is of the order of 60 dBs, far in excess of the processing gain available in the practical system.

We begin the analysis by considering the mixer outputs for an input signal of the form

$$r(t) = s(t) \cos(2\pi f_c t + \phi) \quad (3-1)$$

where $s(t)$ is the spreading code, f_c is the carrier frequency, and ϕ is an arbitrary carrier phase offset. The mixer outputs will be

$$M_I(t) = s(t) \cos(2\pi f_c t + \phi) \cos(2\pi f_c t) \quad (3-2)$$

$$M_Q(t) = s(t) \cos(2\pi f_c t + \phi) \sin(2\pi f_c t) \quad (3-3)$$

where $M_I(t)$ and $M_Q(t)$ are the I and Q channel mixer outputs respectively. Now $s(t)$ may be an arbitrary pseudonoise code of m chips, but for the sake of clarity and without loss of generality the following discussion will consider the case of a code comprising m 1's. This is not an unreasonable case since later in the discussion the modulus of each chip is considered. Under this condition we may replace $s(t)$ by a constant A . Expanding (3-2) and (3-3) gives

$$M_I(t) = A/2 \cos \phi + A/2 \cos(4\pi f_c t + \phi) \quad (3-4)$$

$$M_Q(t) = A/2 \sin \phi + A/2 \sin(4\pi f_c t + \phi) \quad (3-5)$$

Since the mixers are followed by lowpass filters the double frequency terms will be removed and, therefore, taking the modulus of the lowpass filter outputs for the input signal $A \cos(2\pi f_c t)$ we have

$$|\text{LPF } I_{\text{out}}(t)| = |A/2 \cos\theta| \quad (3-6)$$

$$|\text{LPF } Q_{\text{out}}(t)| = |A/2 \sin\theta| \quad (3-7)$$

shown in figure 3-2. It can be seen from (3-6) and (3-7) that, for any arbitrary phase offset, the modulus of the I channel or the Q channel must be greater than zero. This is the fundamental principle on which the complex baseband receiver is based. Further if a frequency difference f_d is present between the local oscillator and the received signal then we may replace $\cos(2\pi f_c t + \theta)$ by $\cos(2\pi(f_c + f_d)t)$. The modulus of the I and Q channel mixer outputs after lowpass filtering is

$$|\text{LPF } I_{f_o}| = |A/2 \cos(2\pi f_d t)| \quad (3-8)$$

for the I channel and

$$|\text{LPF } Q_{f_o}| = |A/2 \sin(2\pi f_d t)| \quad (3-9)$$

for the Q channel.

Therefore the mixer outputs will be sinusoidal with amplitude $A/2$ and frequency equal to the difference or beat frequency. Further a constant phase offset of 90 degrees at the beat frequency will be maintained between the I and Q outputs. The importance of this will be discussed in section 4.2.3 where carrier recovery loops are considered.

3.3 LOWPASS FILTERS

The lowpass filters are an important element in the complex baseband receiver. Careful consideration must be given to each of the filter parameters before an optimum filter can be chosen for a given DSSS receiver application.

The optimum lowpass filter depends on the character of the channel interference, the effect of transmitter filtering, and the receiver down conversion process. Furthermore filtering distorts the waveform and may reduce the snr at the digitizer input unless accurate sampling synchronization can be maintained.

In general a compromise is reached whereby the lowpass filter is matched to the elementary 'chip' waveform. The suitability of this approach is examined.

The main function of the lowpass filter is to reject unwanted signals resulting from down conversion and to limit and define the noise bandwidth into the sampler and digitizer network. For the practical system it is assumed that image interference will be at a frequency sufficiently above the passband that the choice of lowpass filter will not seriously degrade the system performance due to image frequency interference. Special filtering or the processing gain of the DMF may be required to combat interference within the signal passband. Therefore the most important consideration is to correctly choose the filter parameters such that the filter output snr is maximized for a given input snr. The filter parameters will be studied in detail with a view to obtaining this.

3.3.1 CHOICE OF FREQUENCY RESPONSE CHARACTERISTIC

If it is necessary to attenuate strong unwanted signals above the signal passband then a response characteristic with good out of band rejection will be required. In this case a high order Chebychev or Elliptic response might be chosen in preference to a Bessel or Gaussian response. For the purpose of this analysis we will restrict the argument to the reception of a DSSS signal in the presence of AWGN or other DSSS users. In this case it will be assumed that strong out of band signals are not present and therefore we may compare filter characteristics on the basis of phase response, step response, and efficiency relative to an ideal chip matched filter.

Whilst the frequency responses of popular filter types, such as Chebychev, Butterworth, and Bessel filters, are known to differ the important criterion in this application is the snr at the filter output for constant input snr. The choice of optimum cutoff frequency is determined by the noise equivalent bandwidth of the filter. This is also an important factor in determining the loss relative to an ideal chip matched filter. Before these parameters are investigated a fundamental property of sampling a lowpass signal is investigated.

3.3.2 EFFECT OF FILTERING ON CORRELATION

3.3.2.1 CORRELATION OF LOWPASS NOISE SAMPLES

The analysis of the DMF output statistics of Chapter 2 was based on the assumption that the samples into the DMF were statistically independent. It has been shown [PFEIFFER and SCHUM 1973] that samples of a signal with Gaussian statistics will be independent provided the correlation between the samples is zero. It is therefore necessary to investigate the effect of filter cutoff frequency on the correlation between noise samples, since these are assumed

to have Gaussian statistics. To do this we use the Fourier transform relationship between the autocorrelation function $R(t)$ and power spectral density $G(f)$ [CARLSON 1975] where

$$R(t) = F^{-1} (G(f)) \quad (3-10)$$

For the practical filter with transfer function $H(f)$ this may be written

$$R(t) = \int_{-\infty}^{\infty} |H(f)|^2 S(f) e^{j2\pi ft} df \quad (3-11)$$

where $S(f)$ is the power spectral density of the signal at the filter input. For AWGN of double sided power spectral density $N_0/2$ we may write

$$S(f) = N_0/2 \quad (3-12)$$

and for the ideal rectangular response lowpass filter with bandwidth B we may write

$$H(f) = \text{RECT}(f/2B) \quad (3-13)$$

From (3-10), (3-12), and (3-13) we may write

$$R(t) = N_0 B \text{SINC}(2\pi Bt) \quad (3-14)$$

From (3-14) it can be seen that $R(t)$ is a continuous function of time t , but is zero for certain values of t . For a given filter bandwidth B we may select a sampling interval t_s such that the correlation between successive noise samples is zero, i.e. $R(t_s)=0$. Alternatively for a given chip rate and hence sampling interval t_s we may select the appropriate bandwidth B to give the same effect, i.e. $\text{SINC}(2\pi Bt_s)=0$. In this case the sampling interval is fixed since we perform single chip sampling at the chip rate. This gives a sampling interval of $t_s=1/2B$ and we may write $B=f_s/2$, where f_s is the sampling frequency. Therefore for the ideal response characteristic we obtain zero correlation between noise samples if the cutoff frequency is chosen to be equal to half the sampling frequency.

The above analysis has been performed for several practical filter responses. The modulus of the autocorrelation function for these filters is shown in figures 3-3a to 3-3e. The ideal response gives the familiar $\sin(x)/x$ shape, with high side lobes. The Bessel response falls rapidly to a low value but does not come close to zero until Bt is greater than 2. Therefore the Bessel response is poor in the sense that a high Bt product is required for low correlation values. The Gaussian response also falls rapidly towards zero with increasing Bt , and at a Bt value of about 0.7 the correlation becomes approximately 10^{-3} . Both Bessel and Gaussian responses fall monotonically reaching zero at infinity, but never actually reach zero for the range of Bt values shown. The Butterworth and Chebychev responses are somewhat similar in shape to the ideal response. This is a possible indication of the rectangular frequency transfer approximation used for these filter types. For the Butterworth response zero correlation is obtained at a Bt value of approximately

0.57, and for the Chebychev response zero correlation occurs at a Bt value of approximately 0.47.

These results indicate that the cutoff frequency of the filter must be carefully chosen to prevent correlation of lowpass noise samples. This may be likened to the problem of ISI regularly encountered in the design of digital communication systems. In this case it is equivalent to a degree of 'memory' existing in the sampler between successive noise samples if the correlation value is not equal to zero. Having found the required cutoff frequency for each filter type we may now examine the effect that filtering to these bandwidths will have on the correlation of the lowpass signal samples.

3.3.2.2 CORRELATION OF LOWPASS SIGNAL SAMPLES

The power spectral density for an m -sequence with periodic autocorrelation function, as shown in chapter 2 figure 2-13, is a line spectrum with line spacing $1/t_s$, where t_s is the code sequence period. This is bounded by a $\sin^2(x)/x^2$ envelope function [CARLSON 1975] to give

$$E(f) = (1+m)/m^2 \sin^2(x)/x^2 \left\{ \sum_{\substack{k=-\infty \\ k \neq 0}}^{\infty} \delta(f - (k/t_s)) \right. \\ \left. + \delta(f)/m^2 \right\} \quad (3-15)$$

where $x = \pi f t_c / 2$, t_c is the code chip period, m is the code length, and hence $m t_c = t_s$. Having found the optimum

lowpass filter 3 dB bandwidths for low or zero correlation of lowpass noise samples in section 3.3.2.1 we may now extend the argument to the correlation of lowpass signal samples taken at the filter output when the input signal is of the form (3-15). Such analysis is not generally possible for digital systems. However the DSSS signal is a special case for which the expected structure, and hence the autocorrelation function, of the transmitted code is known. For each response considered the 3 dB frequency has been selected on the basis of zero or low correlation of lowpass noise samples. The 3 dB frequency is expressed as a fraction of the sampling frequency, this being the inverse of the sampling period. The values chosen are 0.5, 0.75, 0.7, 0.57, and 0.47 for the ideal, Bessel, Gaussian, Butterworth, and Chebychev filters respectively. The correlation functions are shown in figures 3-4a to 3-4e. From these it can be seen that at the sampling rate, corresponding to sample time = 1.0, correlation exists between lowpass signal samples for all responses considered. These results, and those of section 3.3.2.1, suggest that for filters with these responses it is not possible to obtain zero correlation of lowpass noise and signal samples at the same 3 dB bandwidth. Correlation between lowpass signal samples is more commonly known as intersymbol interference (ISI).

It is possible that a filter response exists which would satisfy the correlation criterion for both noise and signal samples. Alternatively it may be possible to choose the cutoff frequency of a practical lowpass filter on the basis of zero noise sample correlation and then use equalization to reduce the effects of ISI. Whilst some authors have considered this problem [SHIMBO and CELEBILER 1971] it is clearly an area worthy of closer examination. However in view of the need to concentrate on the properties of the DMF within the DSSS system it was felt that further effort could not be spent on this topic.

3.3.3 CUTOFF FREQUENCY AND NOISE EQUIVALENT BANDWIDTH

The calculation of the efficiency of a practical filter response relative to an ideal chip matched filter response requires the calculation of the noise equivalent bandwidth of the filter. For a practical lowpass filter with frequency transfer function $H(f)$ we may define

$$B_n = \int_{-\infty}^{\infty} 1/2 |H(f)|^2 / |H(0)|^2 df \quad (3-16)$$

$H(0)$ is the filter response at $f=0$ Hz and is generally normalized to unity. The noise equivalent bandwidth, B_n , corresponds to the bandwidth of an ideal rectangular magnitude response lowpass filter, with gain $H(0)$, which would give an equivalent output noise power to the practical filter. The ratio of B_n to the 3 dB cutoff frequency, f_c , is tabulated for several filter types and orders, table 3-1.

TABLE 3-1
 PRACTICAL LOWPASS FILTER RESPONSES
 RATIO OF NOISE EQUIVALENT BANDWIDTH TO 3 dB BANDWIDTH
 [BLINCHIKOFF and ZVEREV 1976]

Filter type	Order	B_n / f_c
Bessel	3	1.07
"	5	1.04
Butterworth	3	1.05
"	5	1.02
Chebyshev	3	1.04
(0.1dB)	5	1.01
Gaussian	3	1.11
"	5	1.01

It is clear from table 3-1 that for the practical filters considered the noise equivalent bandwidth is very close to the 3 dB cutoff frequency. Therefore in terms of noise equivalent bandwidth many practical filter responses can be considered to have the same response as a rectangular transfer function.

3.3.4 FILTER LOSS RELATIVE TO IDEAL CHIP MATCHED FILTER

In this section the efficiency of the practical filters relative to the ideal chip matched filter is investigated and the expected loss is calculated. This loss is defined as the difference in snr at the output of the practical filter when compared to the snr at the output of the ideal chip matched filter, and is determined from the following ratio, which is normalized to unity noise power [SCHWARZ 1970]

$$L = \frac{\left| \int_{-\infty}^{\infty} P(w) H(w) e^{jwT} df \right|^2}{\int_{-\infty}^{\infty} |P(w)|^2 df \quad N_0/2 \int_{-\infty}^{\infty} |H(w)|^2 df} \quad (3-17)$$

Here $P(w)$ is the spectrum of the input signal, an isolated rectangular pulse of width τ in this case, $H(w)$ is the frequency response of the practical filter, AWGN is assumed with power spectrum $N_0/2$, and $w = 2\pi f$. Evaluation of (3-17) gives the efficiency of the filter with response $H(w)$ relative to the ideal chip matched filter with response $K e^{jwT} \sin(w\tau/2)/(w\tau/2)$ [SCHWARZ 1970].

Thus (3-17) becomes

$$\eta = \frac{\left| \int_{-\infty}^{\infty} \sin(w\tau/2)/(w\tau/2) H(w) e^{jwT} df \right|^2}{2\pi/\tau N_0/2 \int_{-\infty}^{\infty} |H(w)|^2 df} \quad (3-18)$$

The second term of the denominator of (3-17) is the noise equivalent bandwidth of the practical filter discussed in section 3.3.3. This term determines the noise output power from the practical filter.

The calculation of the loss of the practical lowpass filter relative to the ideal chip matched filter is important for three reasons. Firstly it is known [SKOLNIK 1962] that the ideal matched filter for an arbitrary time waveform has an impulse response which is identical in shape to the signal to which it is matched, but reversed in time. For a rectangular pulse this would imply a rectangular impulse response, and a $\sin(x)/x$ frequency response. Since this function is anticipatory it cannot be achieved in practice. Secondly if the filter introduces excessive loss in the signal relative to the loss in the noise the snr ratio into the sampler will be seriously degraded. Therefore high efficiency relative to the ideal chip matched filter is also required. Thirdly to prevent or minimize correlation between noise samples the 3 dB bandwidth must be correctly chosen for a given sampling rate, as shown in section 3.3.2.1 It is therefore necessary to calculate the loss for the practical filters considered so far. This will indicate which response can best satisfy the above conditions. Several authors have performed this analysis [SCHWARZ 1970], [SKOLNIK 1962]. However the analysis is generally restricted to ideal, single stage RC, or Gaussian responses. The analysis is extended here to practical responses such as Bessel, Butterworth, and Chebychev.

The efficiency relative to the ideal matched filter has been calculated from (3-18) for several responses, figures 3-5a to 3-5e. The results show the loss for a given pulse width \times 3 dB bandwidth (Bt) product. For a given value of t each response has an optimum value of B which minimizes the relative loss. It can be seen that for all responses the minimum loss at the optimum value of Bt is less than 1 dB. Also if the bandwidth is chosen to be small

in order to reduce the noise power, the loss rapidly becomes large. This is because the $\sin(x)/x$ spectrum of the signal has a considerable proportion of the total available energy within the main lobe. If the bandwidth is made large to include other sidelobes the noise power increase becomes larger relative to the signal power increase since the noise spectrum is assumed constant with increasing frequency. Therefore the relative loss increases.

We may now compare the lowpass noise sample correlation results obtained in section 3.3.2.1 with the results obtained above to determine the loss incurred by each practical filter at the optimum bandwidth. For the Bessel filter, figure 3-3b, the correlation value is low at high values of Bt , but does not approach zero until Bt is much greater than 2. At these values of Bt the loss relative to the chip matched filter is high, figure 3-5b. The Gaussian correlation value falls more rapidly, and reaches a value very close to zero, approximately 10^{-3} , at a Bt value of approximately 0.7, figure 3-3c. At this point the loss relative to the chip matched filter is about 1.5 dBs. The Butterworth response has zero correlation at a Bt value of about 0.57. The loss relative to the chip matched filter is slightly higher at about 1.2 dB. On this basis the Chebychev response is best, since it has a loss of 0.7 dB at a Bt value of about 0.47.

This analysis assumes the case of an isolated rectangular pulse in AWGN. It is acceptable for applications such as radar, where pulses are in general isolated and the lowpass filter can be followed by envelope or modulus detection.

For a digital system involving the transmission and detection of a sequence of data bits the minimum bit error probability will be achieved by sampling the pulse at peak snr. Therefore a knowledge of the correct sampling instant must be known at the receiver if the error probability is to be minimized. However when account is made of the

autocorrelation function of the DSSS signal it is clear that the effects of adjacent chips, and hence ISI, must be considered.

Analysis along these lines will lead to the derivation of the optimum matched filter for the entire code. Since we already have a suitable approximation to this in the DMF it is clear that measuring the performance of the lowpass filter relative to the ideal chip matched filter does provide information about the efficiency of the lowpass filter. Some authors have obtained results by computer simulation [HOPKINS and SIMPSON 1975]. Here the filter bandwidth was chosen to be half the chip rate. In this study it was found that an average loss of about 0.75 dBs could be expected at the optimum sampling instant from a 4th order Butterworth filter relative to the ideal chip matched filter. The figures were obtained using an m-sequence as the input data stream. It was found that the loss reduced as the code length was increased, although no explanation for this effect was given. Also the loss increased with increasing filter order, and this was almost certainly due to the increased ISI expected in this case. Unfortunately figures for other filter types were not presented. However the results available showed good agreement with those obtained in this section, where the lowpass filter was considered to be a chip matched filter.

Since the lowpass filter bandwidth is selected to give zero noise sample correlation we must obtain some information on the effects of signal ISI. For this the lowpass filter step response will be considered.

3.3.5 FILTER PHASE AND STEP RESPONSE

The effect of lowpass filtering prior to sampling is often measured by eye diagrams. In this technique an experiment or simulation is performed whereby a pulse or data stream is fed into a lowpass filter. A diagram is then constructed where the vertical scale corresponds to the signal at the filter output whilst the horizontal scale corresponds to the time instant relative to the start of the test pulse or data stream. The optimum filter output for systems which sample at random, such as would be the case during synchronization of a DSSS receiver, is that which gives the largest eye opening over the widest sampling range. For analogue integration receivers the average loss due to filtering may be determined by the average eye opening.

Eye diagrams are normally generated using a long random or pseudorandom sequence. This is necessary since eye patterns are sensitive to the data pattern used for testing, and the length of the test pattern must be greater than the expected length of the ISI tails. Measurements on practical filters or computer simulation are frequently used as these give an immediate insight into the performance of potential filter structures. For a DSSS signal of length m chips it has been shown [GOLOMB 1967] that there are approximately $m/2$ transitions from either 0 to 1 or from 1 to 0. Therefore examination of the lowpass filter risetime will allow the average loss in signal to be found for a given sampling error. This avoids the need to construct eye diagrams and is only possible since the statistics of runs of 1's and 0's within the DSSS code are known. This might not be the case for a general data transmission network. The lowpass filter risetime will now be considered.

Lowpass filters with constant time delay (linear phase) have low or zero transient overshoot characteristics [BLINCHIKOFF and ZVEREV 1976]. This property arises from

the symmetric impulse response obtained with these filters. Gaussian and Bessel filter characteristics are approximations to a linear phase response and hence have low overshoot characteristics. Butterworth, Chebychev, and filters based on constant magnitude frequency response approximations generally have non-linear phase responses and can thus exhibit considerable transient overshoot. If a pulse is sampled, overshoots and ringing, due to the transient response of the filter, may lead to sampling errors. Also if the optimum sampling instant is not known the sampling error probability may be reduced by selecting a lowpass filter response which exhibits a short risetime. This will then minimize the fractional chip period sampled at reduced snr. Therefore to minimize sampling errors the lowpass filter should have a short risetime and zero or low overshoot.

The step response of an analogue filter is also linked to the phase response. The phase and amplitude responses are linked as these arise from the frequency domain characteristic approximation chosen for the filter. The filter impulse response is the inverse Fourier transform of the frequency domain complex amplitude response. Since the step response is the time integral of the impulse response filters with impulse response overshoots will also have step response overshoots [BLINCHIKOFF and ZVEREV 1976]. This is a common feature of all non-equalized filters which attempt to approximate the ideal frequency domain amplitude response of infinite stopband attenuation and zero passband to stopband transition. The filter which provides the fastest monotonic step response for a given bandwidth is the prolate filter [BLINCHIKOFF and ZVEREV 1976]. A practical realization with zero overshoot can be obtained by using a filter with equiripple stopband attenuation. However forcing the step response to be monotonic results in a large risetime relative to the equiripple passband realization. Therefore a compromise must be made between transient response, and hence risetime, and stopband attenuation.

Traditionally the risetime has been defined as the time necessary for the step response to go from 10% to 90% of its final steady state value, figure 3-6a. An alternative definition is the time required for the steady state value to be reached at a rate equal to the maximum slope of the risetime, figure 3-6b. Further definitions [SU 1971] attempt to minimize the least square errors in the risetime relative to a straight line 'ideal' risetime, figure 3-6c. Comparison of filter risetimes can therefore be misleading since filters with very different responses may have the same risetime. A further complication is added when the frequency of normalization is not chosen to be the 3 dB cutoff frequency. One definition [ZVEREV 1967] normalizes the risetime to the 40 dB cutoff frequency. If Gaussian and third order Chebychev filter risetimes are compared on the basis of maximum slope normalized to the 3 dB cutoff frequency then the Gaussian risetime is lower by a factor of approximately 0.5, for a given 3 dB cutoff frequency. If the comparison is based on the maximum slope definition normalized to the 40 dB cutoff frequency then the Gaussian risetime is higher by a factor of approximately 0.5 [BLINCHIKOFF and ZVEREV 1976]. In many practical applications the problem will be resolved by specifying factors such as overshoot and settling time.

The risetime of the lowpass filter is important when considering the DSSS receiver operating prior to synchronization. In this condition the optimum chip sampling instant will be unknown. If sampling occurs on the rising edge of the chip the effective snr will be lower than when sampling the chip at a time instant after the risetime transition, ignoring any effects due to overshoot. Therefore the snr at the sampling instant is affected by the sampling instant relative to the start of the received chip and the filter risetime. Table 3-2 gives risetime figures by two definitions for the practical filters. From this it can be seen that the risetimes of Gaussian and Bessel responses are lower than those of Butterworth and

Chebyshev responses, according to these risetime definitions.

Since Bessel and Gaussian responses are approximations to linear phase responses, and thus have the added benefit of low overshoot, they are preferable to Butterworth and Chebyshev responses since they minimize the fractional chip period sampled at reduced snr.

TABLE 3-2
PRACTICAL LOWPASS FILTER RESPONSES
NORMALIZED RISE TIME ($t_{\text{rise } w_{3\text{dB}}}$)
[ZVEREV 1967],[BLINCHIKOFF and ZVEREV 1976]

Filter type	Order	Risetime 10-90%	Risetime (SU)	Risetime Max. sl.
Bessel	3	2.2	2.6	2.1
"	5	2.2	2.7	2.2
Butterworth	3	2.3	2.8	2.5
"	5	2.6	3.1	2.7
Chebyshev (0.1dB)	3	2.7	3.3	2.6
	5	2.9	3.5	3.0
Gaussian	3	2.2	2.6	2.1
"	5	2.2	2.7	2.1

3.3.6 CHOICE OF PRACTICAL LOWPASS FILTER

For the practical system the filter structure chosen was the Gaussian to 12 dB approximation. The three factors considered to be most important in this case were noise correlation, transient overshoot, and risetime. Since the effects of noise correlation are difficult to quantify it was felt that attempting to avoid this problem entirely would be sensible in view of the main objective of the

study. The best way to achieve this was to select a filter type which gave a zero, or very close to zero, correlation of noise samples at a suitable bandwidth. Further since transient overshoot and ringing could lead to sampling errors the required filter should exhibit a linear phase response. Finally since only a single sample per chip would be taken it was felt that minimization of filter risetime was important, since this would minimize chip sampling errors prior to synchronization and tracking.

Also whilst other filter types offered potentially lower losses relative to the ideal chip matched filter the realisation of such potential would not be achieved unless the optimum filter sampling instant was known. This arises since the minimum loss relative to an ideal chip matched filter is only achievable at 1 instant in time, the optimum sampling instant. Prior to synchronization this information would not be known.

It is possible that a superior choice of filter exists for this particular application. The use of equalizing networks to reduce the effects of channel distortion in telephone networks was suggested some time ago [LUCKY 1965]. This could also be applied to the equalization of lowpass filters. Alternatively different filters could be 'switched in' at different points in the operational cycle to optimize the receiver sensitivity. Advances in digital signal processing techniques would undoubtedly prove valuable in this area. However it is clear that choosing the optimum lowpass filter for a baseband DSSS receiver is not a trivial task. At best a loss in effective snr of between 0.5 and 1 dB will be experienced, a figure not so far below the loss incurred by hard limiting. Failure to consider and optimize all aspects of the lowpass filtering process could lead to a reduction in overall system performance. At worse this could lead to the situation whereby the processing gain available within the DSSS receiver is less than the signal loss due to the selected LPF, and the entire DSSS process would have been wasted.

3.4 RF LOCAL OSCILLATOR AND SYSTEM CLOCK RELATIONSHIP

It was shown in section 3.2 that the modulus of the received chip snr was a function of the relative phase between the carrier of the received data and the receiver local oscillator. Further considering the lowpass filter risetime discussed in section 3.3.5 it is clear that there are two independent variables which determine the effective sampling snr for a given received snr. The interaction between these two variables, the lowpass filter risetime and the effect of the carrier phase on the modulus of the received chip must now be considered. Clearly if these can be linked such that only one variable need be considered synchronization at the receiver will be simplified.

This is possible if the carrier frequency is linked to the sampling clock frequency. If the transmitted signal is of the form

$$x(t) = s(t) \cos(2\pi K f_{tr} t) \quad (3-19)$$

where $s(t)$ is the spreading code, f_{tr} is the transmitter code generator clock rate, and K is a multiplicative factor, and the receiver down converts with local oscillator frequency $K f_{re}$, then the mixer outputs ignoring the double frequency terms for the I and Q channels will be

$$M_I(t) = s(t) \cos(2\pi K (f_{tr} - f_{re}) t) \quad (3-20)$$

$$M_Q(t) = s(t) \sin(2\pi K (f_{tr} - f_{re}) t) \quad (3-21)$$

These are derived from equations (3-8) and (3-9). If we

begin by considering the case where the receiver is sampling at the beginning of the chip we may find the time taken for the sampling to slip by one complete chip. At this point the sampling will have slipped to the beginning of the preceding or following chip. The transmitter sends gf_{tr} bits in g seconds, and the receiver samples gf_{re} times in g seconds. For one chip difference we have

$$gf_{tr} - gf_{re} = 1 \quad (3-22)$$

thus

$$g = 1/(f_{tr} - f_{re}) \quad (3-23)$$

If sampling is at the correct rate $f_{tr}=f_{re}$ there is no slip and g is infinite. We may now write (3-20) and (3-21) as

$$M_I(t) = s(t) \cos 2\pi Kt/g \quad (3-24)$$

$$M_Q(t) = s(t) \sin 2\pi Kt/g \quad (3-25)$$

Thus at the receiver there are K/g beat cycles per second in each channel. Alternatively in g seconds there are K beats or $2K$ nulls in each channel. If K is an integer then in g seconds the receiver will slip by one chip in sampling and pass through K nulls at the mixer outputs. At time $g+g_d$ seconds the relative sampling instant and the null pattern will be identical to that at g_d seconds*. Thus it is possible to lock the null pattern to the lowpass filter risetime pattern by selecting an integer ratio between the

* where g_d is an arbitrary time instant

RF local oscillator and the baseband sampling frequency. The modulus of the lowpass filter outputs will therefore be the superposition of the lowpass filter risetime and the beat pattern due to the RF oscillator offset, figure 3-7, giving after lowpass filtering

$$\text{LPF } I_{\text{out}}(t) = s(t) r(t) \cos(2\pi Kt)/g \quad (3-26)$$

$$\text{LPF } Q_{\text{out}}(t) = s(t) r(t) \sin(2\pi Kt)/g \quad (3-27)$$

where $r(t)$ a function defining the amplitude of the risetime of the lowpass filter at time t .

The effect of the lowpass filter risetime can be appreciated by comparing figures 3-2 and 3-7. From figure 3-7 it can be seen that linking the RF and sampling frequencies by an integer ratio causes the basic null pattern due to RF phase offsets to be linked to the relative sampling instant. In the diagram shown the ratio between RF frequency and sampling frequency is 8, and a sampling slip of one chip occurs. The ratio has been set low to exaggerate the effect in the diagrams. Therefore in the time taken to slip by one chip in sampling the RF beat pattern has passed through 8 cycles, or 16 nulls, figure 3-7a. In this time sampling will have occurred at all relative phases of the lowpass filter risetime curve, and this can clearly be seen superimposed on the basic beat pattern. If sampling continued to slip in the same relative time direction the result would be the superposition of risetime patterns over the basic null pattern whenever a chip transition occurred. This also highlights the need for minimum lowpass filter risetime, since this minimizes the fractional chip period sampled at reduced snr under non-synchronized conditions. A further observation that can be made from figure 3-7 is that with single chip sampling in noise a reduced sampling snr may occur if the relative sampling instant is close to the beginning of a chip

transition. This could result in the receiver failing to achieve synchronization, even with an adequately high received snr at the lowpass filter inputs.

This important effect is entirely due to the risetime of the lowpass filters and is generally omitted in the theoretical analysis of most authors when synchronization techniques are presented. From figure 3-7 it can be seen that a very simple technique can be used to overcome this problem. If a receiver failed to synchronize the sampling instant could be changed by $1/2$ of a sampling clock cycle, and the synchronization procedure could then be repeated. If sampling was originally occurring on the rising edge of the lowpass filter risetime response, and consequently at reduced snr, then skipping half a sampling period would result in sampling in or close to the region of maximum output.

Thus linking the RF and sampling frequencies has benefits in receiver design and analysis and is particularly important when synchronization and RF recovery are considered.

3.5 SAMPLE AND HOLD AND A/D CONVERTERS

These elements convert the analogue output from the lowpass filters into digital form. Samples are taken at a rate f_{re} , this being termed the chip sampling rate. The number of levels of quantization required in the A/D section is determined by the DMF structure, as described in chapter 2. In the receiver under discussion the sample and hold and A/D section is replaced by a hard limiter. This is equivalent to a sample and hold device followed by a single bit A/D converter. Since these elements are assumed to be almost ideal in practice further discussion will be left to chapter 5, where a practical implementation is discussed.

3.6 DIGITAL MATCHED FILTER

The operation of the DMF was discussed in detail in chapter 2. In the complex baseband receiver two DMFs are required. These are identical, and provide outputs at the chip sampling rate. Each DMF gives an output in the range $\{0..m\}$ where m is the DMF length. If the received chip stream is sampled once per chip then the DMF length will be equal to the code length. If the chip stream is oversampled then it is necessary to extend the length of the DMF by the oversampling factor. In the practical system to be described the DMF length is 511 chips giving a maximum processing gain of 27 dBs, less the loss of 2 dBs due to the use of single bit quantization, giving an effective processing gain of 25 dBs.

3.7 COMBINING ALGORITHM

It was shown in section 3.1 that a complex baseband structure is necessary for a digitally implemented DSSS receiver. A theoretical receiver study has been performed [TURIN 1976] using such a receiver structure. However in this study the DMF outputs were converted back into analogue form, squared, and summed prior to envelope (modulus) detection. Since we are considering a digital approach to the DSSS receiver it is necessary to consider a digital algorithm for combining the outputs from the two DMFs. Such an algorithm should give an output similar to that obtained from an ideal coherent receiver structure operating with a single DMF. Therefore it is desirable that, for a constant input snr, the algorithm should give a constant mean output as the carrier phase is varied, that is the algorithm output should not reflect the value of the relative phase of the carrier. This is important since varying carrier phase may occur during synchronization acquisition.

The first stage in considering the combining algorithm

is to remove the offset inherent in the DMF output. It was shown in section 2.3.5 that for a hard-limiter input of noise alone the DMF output probability distribution has a mean value of $m/2$. Since the DMF output probability distribution is symmetrical in this case, and further if SIK modulation is used, then the signal plus noise DMF output probability distribution is also symmetrical about the DMF output value of $m/2$, as shown in section 2.3.3. Therefore it is possible to subtract $m/2$ from the DMF output and the result is a measure of distance of the DMF output from the mean output, assuming equally likely symbol 0's and 1's. Therefore the modified output comprises two parts. The first part is the sign bit which determines whether the output is greater than or less than the value $m/2$. The second part is the distance between the output and the value $m/2$.

The first algorithm to be considered is a direct summing algorithm. Here the modified DMF outputs are simply added together. The result does not have a constant mean value as the carrier phase is varied. This can be seen by considering the case where the input snr is sufficiently high such that the contributions to the output mean made by DMF output values less than $m/2$ is zero, that is the DMF output probability only has significant terms for output values greater than $m/2$ in both the I and Q channel DMFs. From chapter 2 we have

$$\text{Mean DMF output} = m(1-p_e) \quad (3-28)$$

and also

$$p_e = 1/2(1-(2/\pi)^{1/2}V/\sigma) \quad (3-29)$$

whence

$$\text{Mean DMF output} = m(1/2 + (2/\pi)^{1/2} V/\sigma) \quad (3-30)$$

For the I channel we have

$$\text{Mean } I_{\text{DMF}} \text{ output} = m(1/2 + (2/\pi)^{1/2} V/\sigma \cos\phi) \quad (3-31)$$

and for the Q channel

$$\text{Mean } Q_{\text{DMF}} \text{ output} = m(1/2 + (2/\pi)^{1/2} V/\sigma \sin\phi) \quad (3-32)$$

Now since the value $m/2$ is subtracted prior to summing (3-31) and (3-32) may be written

$$\text{Mean } I_{\text{sub}} \text{ output} = m(2/\pi)^{1/2} V/\sigma \cos\phi \quad (3-33)$$

$$\text{Mean } Q_{\text{sub}} \text{ output} = m(2/\pi)^{1/2} V/\sigma \sin\phi \quad (3-34)$$

Summing gives

$$\text{Mean sum output} = m (2/\pi)^{1/2} V/\sigma (\cos(\phi) + \sin(\phi)) \quad (3-35)$$

The mean output varies with carrier phase, and reaches a peak value at 45 degrees. At this point the mean output will be approximately 1.414 times greater than the value at 0 or 90 degrees.

The second algorithm to be considered calculates the

modulus of the modified DMF outputs by squaring, summing, and taking the square root of the modified DMF outputs. From (3-33) and (3-34) we obtain

$$(\text{Mean output})^2 = (m/(2\pi)^{1/2})^2 (V/\sigma)^2 (\sin^2\phi + \cos^2\phi) \quad (3-36)$$

$$\text{Mean output} = m/(2\pi)^{1/2} V/\sigma \quad (3-37)$$

In this case the dependence on the carrier phase has been removed and the algorithm gives a mean output which is a direct measure of the input snr.

Whilst these expressions may provide some idea of the likely trends their accuracy is limited, as a result of the approximations made. Firstly (3-28) is based on the assumption that the snr is low, as shown in section 2.3.1. Secondly it was necessary to assume that for both I and Q channels the DMF output probability distributions were entirely to one side of the value $m/2$. This allowed the output mean to be determined by subtracting the value $m/2$ from the mean output of a Binomial distribution. For this to be true it is necessary for p_e to be low, which implies a high value of snr. Thirdly it is clear that if the carrier phase is 0 degrees the input snr to the Q channel will be 0, irrespective of the snr into the Q channel mixer. Therefore the Q channel output probability distribution cannot lie entirely to one side of the value $m/2$, but must be symmetrical about the value $m/2$. Consequently the approximations cannot be true for carrier phase values close to 0 or 90 degrees, or multiples thereof. Therefore both algorithms have been evaluated numerically. This involves calculating the algorithm output mean and standard deviation.

Before these can be found it is necessary to consider

the effect of subtracting the offset from the DMF output. This is important since in practice integer arithmetic will be used in the DMF and algorithm operations. Since the DMF length m is odd (511 for the practical system considered) it is necessary to round off the value of $m/2$, since this will not be an integer. In practice subtracting an offset maps the DMF output from the range $\{0..m\}$ to the range $\{0..(m-1)/2\}$. This is shown in figure 3-8. It can be seen that if the DMF output is greater than or equal to $(m+1)/2$ the offset subtracted is $(m+1)/2$. If the DMF output is less than or equal to $(m-1)/2$ the offset subtracted is $(m-1)/2$. In the second case the modified output will be in the range $\{0..(m-1)/2\}$ but this case is identified from the first case by the sign bit. Subtraction of different offsets according to the size of the DMF output causes a slight discontinuity in the modified output. This arises since DMF outputs of $(m-1)/2$ or $(m+1)/2$ are both mapped to a modified output value of 0. This is necessary for practical reasons and is discussed in chapter 5. For large values of m this discontinuity is unlikely to cause significant problems.

The algorithm output statistics can now be considered. The I and Q channel DMF output statistics are considered independent. This is a reasonable assumption since there is no interaction between the two DMFs, although there is a relationship between the I and Q channel chip error probabilities. The modified I and Q channel DMF output probability distributions are found in a similar way to the Binomial distribution discussed in section 2.3.2 of chapter 2.

For the I channel we have

$$\text{Prob}\{I_{\text{out}}=i\} = \sum_m^j C p_I^j (1-p_I)^{m-j} + \sum_m^k C p_I^k (1-p_I)^{m-k} \quad (3-38)$$

and for the Q channel

$$\text{Prob}\{Q_{\text{out}}=i\} = \sum_m^j p_Q^j (1-p_Q)^{m-j} + \sum_m^k p_Q^k (1-p_Q)^{m-k} \quad (3-39)$$

where

$$j = (m+1)/2 + i \quad , \quad k = (m-1)/2 - i$$

and p_I and p_Q are the I and Q channel chip error probabilities.

The algorithm output mean and standard deviation can be found from the above probability distributions once the effect of the algorithm mapping has been considered. This is shown in figure 3-9 for the summing and square root algorithms. Figure 3-9a shows how the summing algorithm maps the modified I and Q channel output distributions, each in the range $\{0..(m-1)/2\}$, to a single distribution with an output range of $\{0..(m-1)/2\}$. The probability that the summing algorithm output has a particular output x is given by

$$\text{Prob}\{\text{sum}=x\} = \sum_{i=0}^{(m-1)/2} \text{Prob}\{I_{\text{out}}=i\} \text{Prob}\{Q_{\text{out}}=x-i\} \quad (3-40)$$

Figure 3-9b shows the mapping for the square root algorithm. It can be seen that the I and Q channel output distributions are first mapped by a squaring operation. The sum is then formed. Finally the square root of the sum of

the squares is taken to give the algorithm output. For this algorithm the output statistics are more difficult to write down in an explicit form. This arises since not all algorithm output values can exist in practice. This is apparent by considering the algorithm output to be the sum of two integer values squared. For example the output value 5 can exist, since this is the sum of 4 and 1 which are the integers 2 and 1 squared. However the value 6 cannot exist, since it is not possible to find a pair of integers which will give this value when squared and summed. Further taking the square root would require a transfer to floating point arithmetic. This is undesirable from a practical viewpoint and therefore the algorithm output is rounded to the nearest integer. These conditions preclude the formulation of a simple expression of the form (3-40) for the square root algorithm. In practice the algorithm output mean and standard deviations have been found numerically.

The output mean and standard deviation has been found for each algorithm from

$$\begin{array}{l} \text{Max.} \\ \text{output} \end{array}$$

$$\text{Alg. Mean output} = \sum_{i=0} i \text{ Prob}\{\text{Alg}=i\} \quad (3-41)$$

and

$$\begin{array}{l} \text{Max.} \\ \text{output} \end{array}$$

$$\text{Alg. Output Var.} = \sum_{i=0} (i-\text{mean})^2 \text{ Prob}\{\text{Alg}=i\} \quad (3-42)$$

Figures 3-10a, 3-11a, and 3-12a show these against carrier phase, for input snr values of 0.1, 0.5, and 1.0. These correspond to input values of E_c/N_o for BPSK modulation of -23 dBs, -9 dBs, and -3 dBs respectively. This was considered to cover the range of values likely to be of interest in the test system.

These figures show that the summing algorithm mean output varies with carrier phase, for a constant input snr. The mean output reaches a peak value at 45 degrees, as predicted by the approximation (3-35). The square root algorithm mean output is almost constant with varying carrier phase, again with constant input snr. For this algorithm the mean output becomes less uniform at higher snrs. Also, for constant input snr, the summing output produces a higher mean output than the square root algorithm as the carrier phase is varied. However the ratio of mean to standard deviation gives an indication of how 'noisy' the algorithm output is. Figures 3-10b, 3-11b, 3-12b show the ratio of algorithm output mean to standard deviation against carrier phase for the same input snrs as before. These show that the square root algorithm has higher output mean to standard deviation than the summing algorithm, as the carrier phase is varied, and for the three test snr values. Thus the square root algorithm gives a less noisy output.

At a carrier phase value of 45 degrees the summing algorithm does not have a significantly lower mean output to standard deviation ratio. The mean output is, however, significantly greater than the mean output of the square root algorithm. This would suggest that at carrier phase values in the region of 45 degrees the summing algorithm gives a better output than the square root algorithm. At carrier phase values close to 0 or 90 degrees the square root algorithm has a higher output mean to standard deviation ratio, with only a slightly lower mean output.

Therefore at these carrier phases the square root algorithm gives the best output.

Finally for each algorithm the mean output and ratio of mean output to standard deviation has been calculated for increasing input snr, and at constant carrier phase values of 0 degrees, figure 3-13, and 45 degrees, figure 3-14. From figures 3-13a and 3-14a it can be seen that above an input snr value of about 0.1 ($E_c/N_o = -23$ dBs) each algorithm gives a mean output which is approximately linear with input snr. This is significant since it leads to the possibility of using the algorithm to measure the input snr. Knowledge of the received snr could be useful in systems employing power control, or covert systems where it may be desirable to maintain the lowest transmitted power level for acceptable reception. Figures 3-13b and 3-14b indicate that for each algorithm the ratio of output mean to standard deviation is also approximately linear with increasing input snr, and for constant carrier phase. This suggests that the output standard deviation does not increase with input snr.

3.8 CONCLUSIONS

In this chapter a complex baseband receiver structure is proposed and analysed in detail. The function of this sub-system is to convert the received data code at RF to a baseband data stream from which synchronization, tracking and RF recovery information can be obtained. It was shown that a dual channel structure is necessary at baseband. This has carrier in phase (I) and quadrature phase (Q) baseband channels. This prevents total loss of signal due to carrier phase offsets, and is necessary to recover carrier phase information.

The system elements were considered separately, and their influence on the overall performance of the complex baseband receiver was discussed. The process of down

conversion, or mixing, was assumed to be ideal. This assumption was based on manufacturers' information which suggested that distortion due to non-linearities and noise would be negligible provided the mixing devices were operated within their rated specifications.

Lowpass filtering was considered in detail. The choice of lowpass filter response is based on parameters such as stopband attenuation, step response, and efficiency. Five filter responses were considered, one ideal and four realizable in practice. The effect of sampling at the filter output was considered. It was shown that for the five filter types examined it was not possible to obtain zero correlation of lowpass noise and signal samples in a given filter bandwidth. Equalization of the filter output was proposed as a technique to reduce the consequent ISI.

The efficiency of the lowpass filter relative to an ideal chip matched filter was found to be a useful parameter once synchronization has been achieved, as it is then possible to sample the filter output at the optimum sampling instant. At best an effective loss in snr of about 0.5 to 1 dB would be experienced by the use of a Butterworth approximation to the chip matched filter, a figure comparable with the loss incurred by hard limiting.

In view of the main objective of this work it was felt that further investigation into this problem was beyond the scope of this thesis. For the practical system a Gaussian to 12 dB lowpass filter was selected. This filter was chosen on the basis of correlation properties, risetime, and transient response.

For conversion from analogue to digital format single bit quantization (hard limiting) was chosen. Further a single sample per chip will be taken, as this approach requires the minimum size of DMF for a given code length. Therefore it is possible to investigate experimentally a system with the highest processing gain available for a

given size of DMF. As shown in chapter 2 a minimum loss in processing gain of about 2 dB can be expected when using this level of quantization, assuming ideal sampling conditions.

It was shown that linking the RF frequency to the sampling frequency by an integer ratio was preferable to allowing these frequencies to be independent. The effects of the lowpass filter risetime and relative carrier phase can then be evaluated for each relative sampling instant. Linking these frequencies is not necessarily restrictive since adjacent DSSS carriers can be spaced a simple multiple of the chip bandwidth apart.

To combine the two modified DMF outputs two algorithms were considered. The simplest technique formed the sum of the two outputs. This algorithm was shown to give an output which varied in amplitude as the carrier phase was varied relative to the phase of the received RF signal. The second technique summed the square of the modified DMF outputs. The square root of this summation was then taken. This algorithm gave an output which was largely independent of the relative carrier phase. Each algorithm gave an output which was shown to be approximately proportional to the input snr. This leads to the possibility of using the algorithm to measure the input snr. The availability of such information could be useful in systems employing power control or where the need to maintain minimum transmitted power was desirable and feedback was available. If the simple summing algorithm was used it would be necessary to maintain a constant relative phase at RF to eliminate the variation in algorithm output with relative carrier phase.

The complex baseband receiver is a sub-system of the DSSS receiver which provides a digital output representing the correlation between the input data stream and the local replica code. An output is available from the complex receiver on every sampled chip. To demodulate the received data it is necessary for the receiver to acquire code lock,

and if SIK modulation is used the receiver must also maintain RF phase coherence. These problems are discussed in chapter 4.

REFERENCES

- BLINCHIKOFF H.J. and ZVEREV A.I. (1976) 'Filtering in the Time and Frequency Domains', John Wiley and Sons.
- CARLSON A.B. (1975) 'Communication Systems', McGraw-Hill, 2nd edition.
- CHRISTIE R. (1986) 'Implementing Matched Filters in Direct Sequence Spread Spectrum Systems using the MA717 Bit-slice Correlator', Application note Marconi Electronics Devices, Doddington Road, Lincoln, England.
- GOLOMB S.W. (1967) 'Shift register sequences', Holden-Day.
- HOPKINS P.M. and SIMPSON R.S. (1975) 'Probability of error in PN-modulated Spread Spectrum Binary Communication Systems', IEEE Transactions on Communications, April 1975, pp 467 - 472.
- LUCKY R.W. (1965) 'Automatic Equalization for Digital Communication', Bell Systems Technical Journal, Volume 44, April 1965, pp 547 - 588.
- PFEIFFER P.E. and SCHUM D.A. (1973) 'Introduction to Applied Probability', Academic Press.
- SCHWARZ M. (1970) 'Information Transmission, Modulation, and Noise', McGraw-Hill, 2nd edition.
- SHIMBO O. and CELEBILER M.I. (1971) 'The Probability of error due to ISI and Gaussian Noise in Digital Communication Systems', IEEE Transactions on Communications, April 1971, pp 113 - 119.

- SKAUG R. and HJELMSTAD J.F. (1985) 'Spread spectrum in communication', Peter Peregrinus.
- SKOLNIK M.L. (1980) 'Introduction to Radar Systems', McGraw-Hill.
- SU K.L. (1971) 'Time Domain Synthesis of Linear Networks', Prentice Hall.
- TURIN G.L. (1976) 'An Introduction to Digital Matched Filters', Proceedings of the IEEE, Volume 64, Number 7, pp 1092 - 1112.
- ZVEREZ A.I. (1967) 'Handbook of Filter Synthesis', John Wiley and Sons.

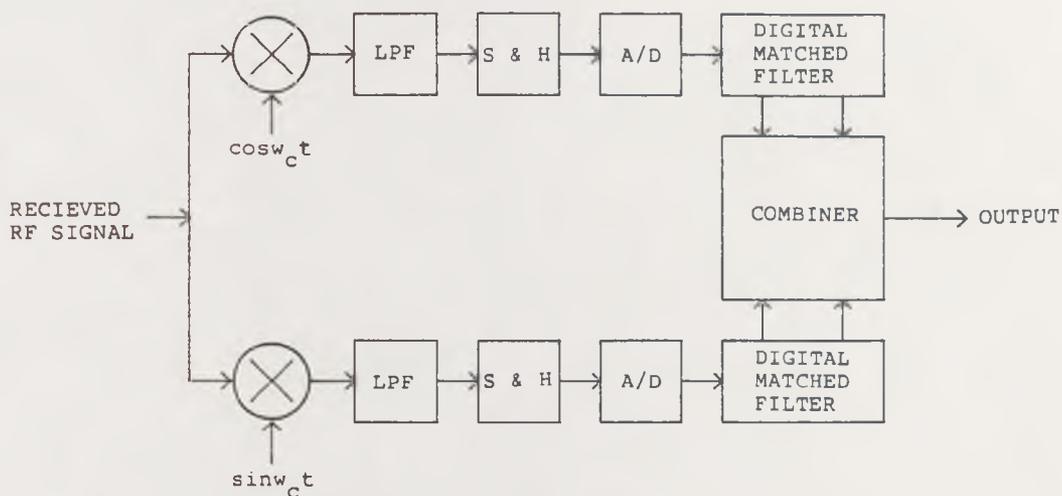


Figure 3-1. Complex Baseband Receiver.

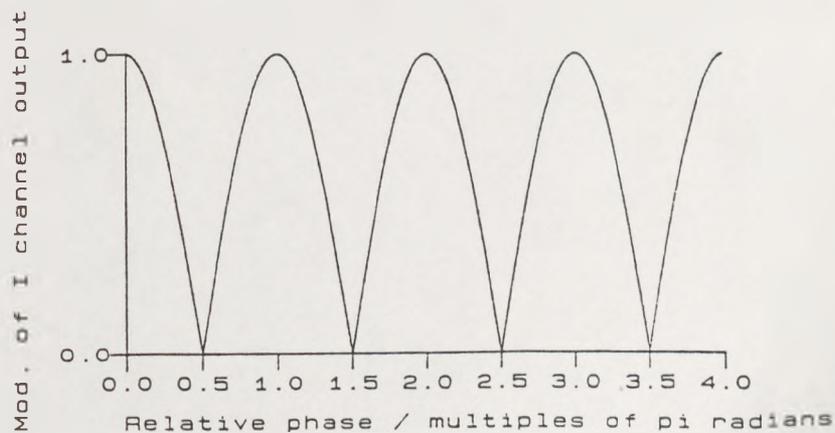
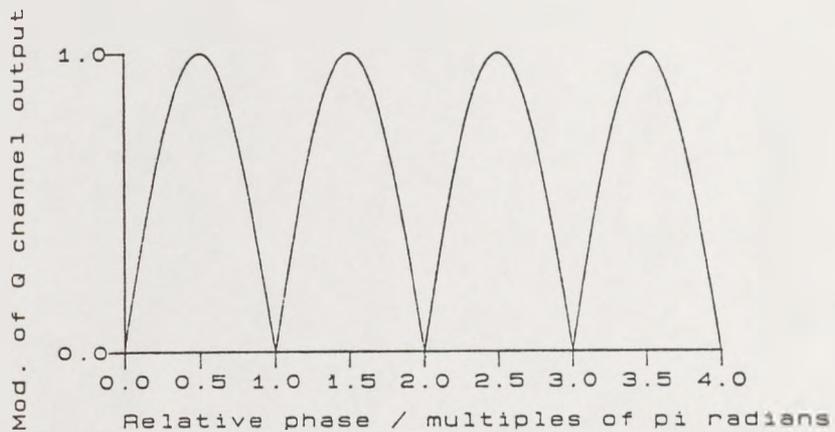


Figure 3-2. Modulus of I and Q channel outputs with varying relative RF carrier phase.

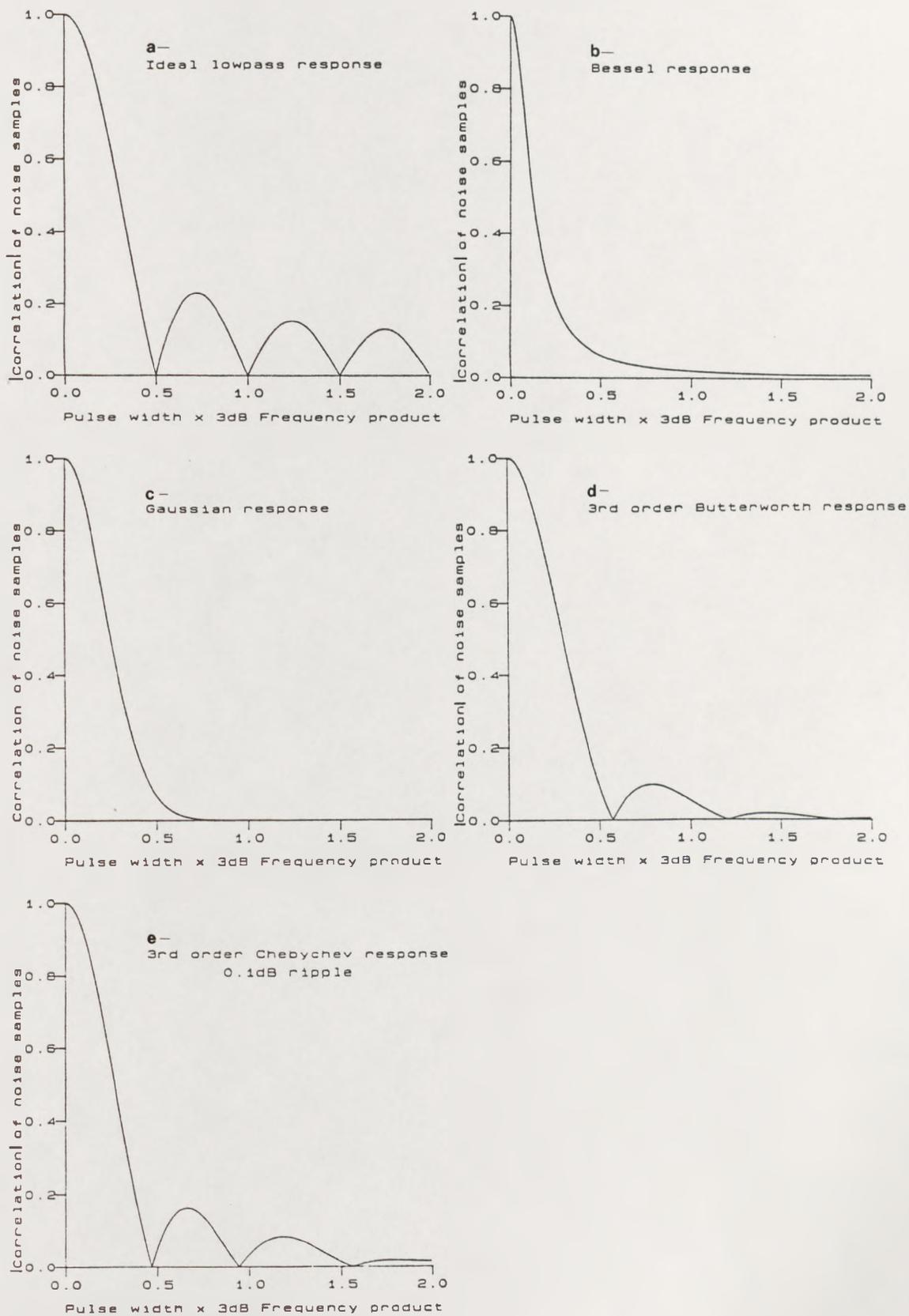


Figure 3-3. Modulus of correlation of lowpass noise samples for test filters.

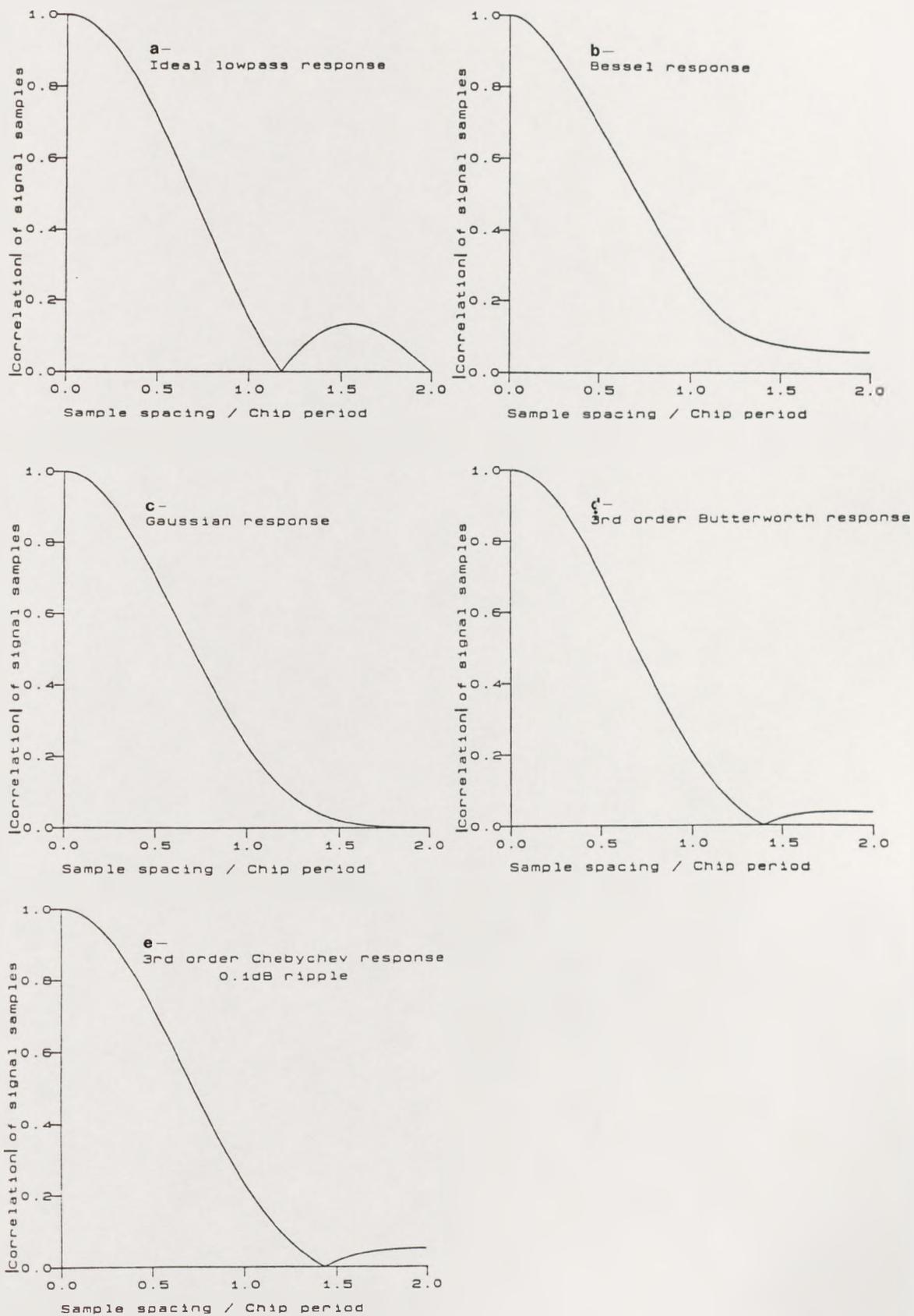


Figure 3-4. Modulus of correlation of lowpass signal samples for test filters.

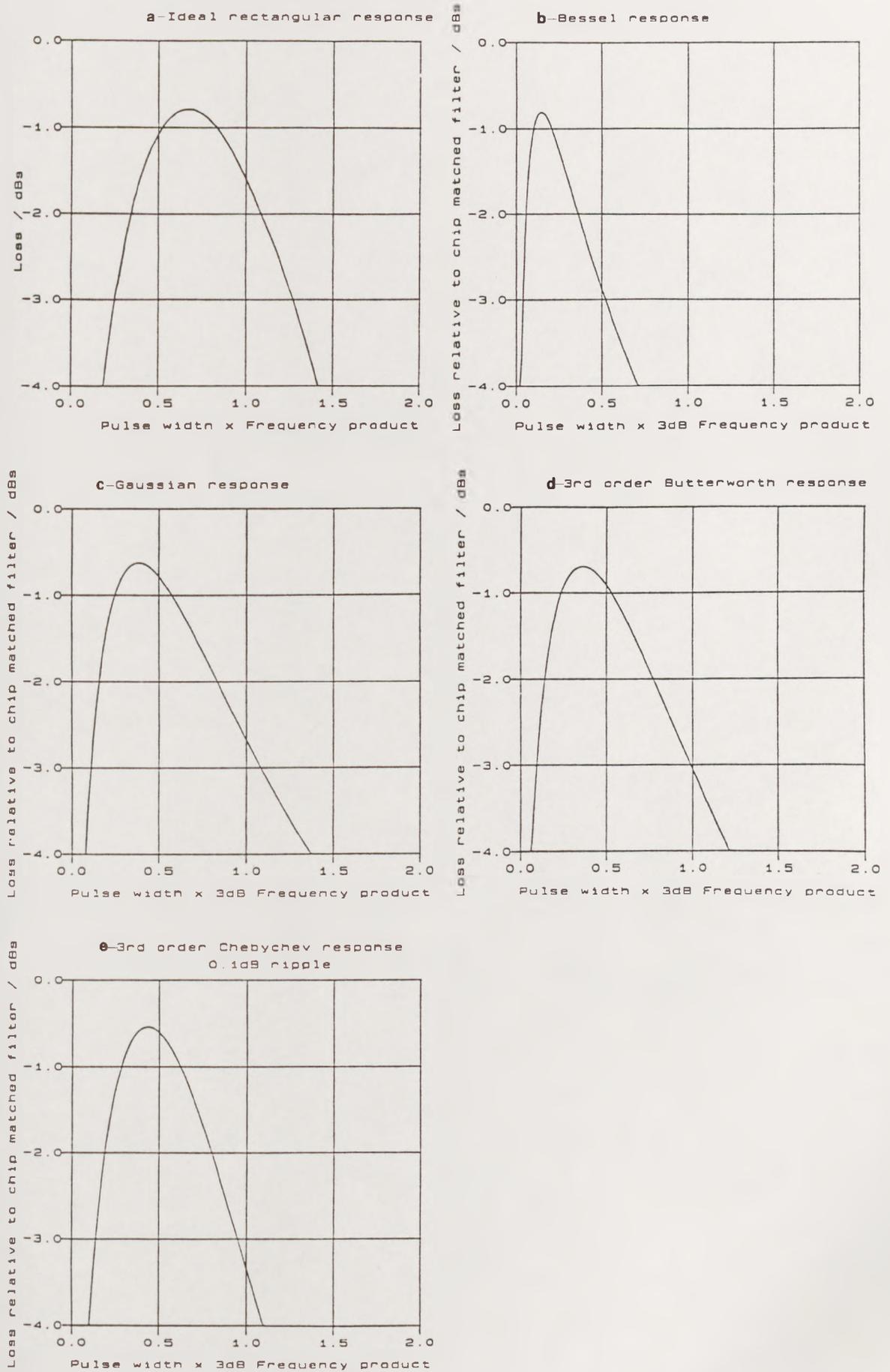


Figure 3-5. Loss relative to ideal chip matched filter for test filters.

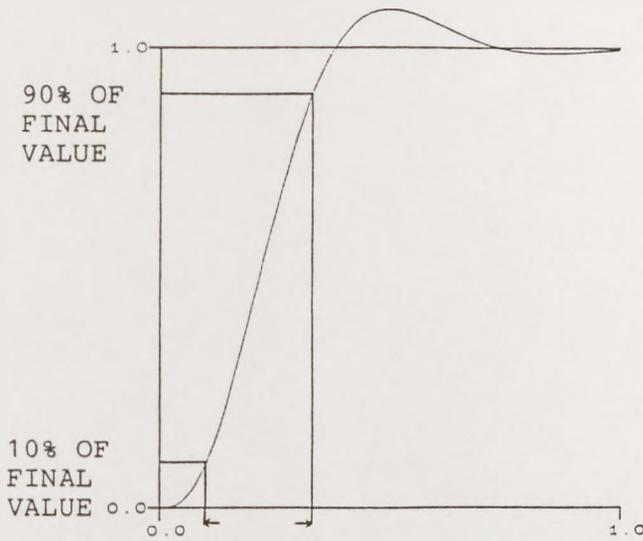


Figure 3-6a. Risetime by 10% - 90% definition.

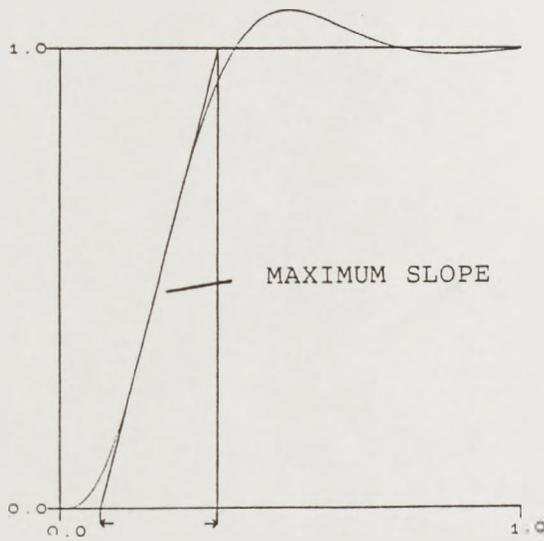


Figure 3-6b. Risetime by maximum slope definition.

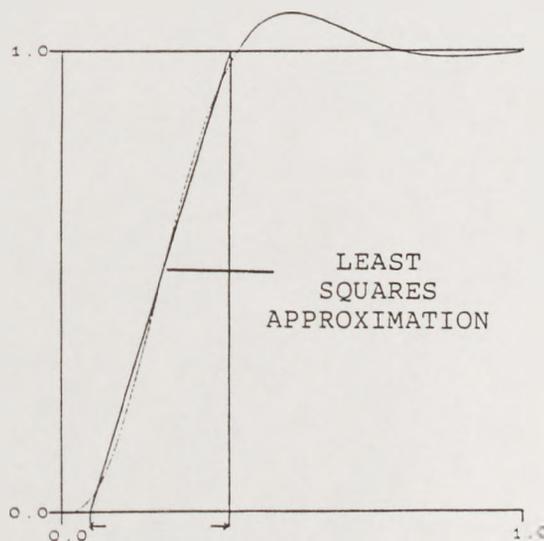


Figure 3-6c. Risetime by Su's definition.

Figure 3-6. Lowpass filter risetime definitions.

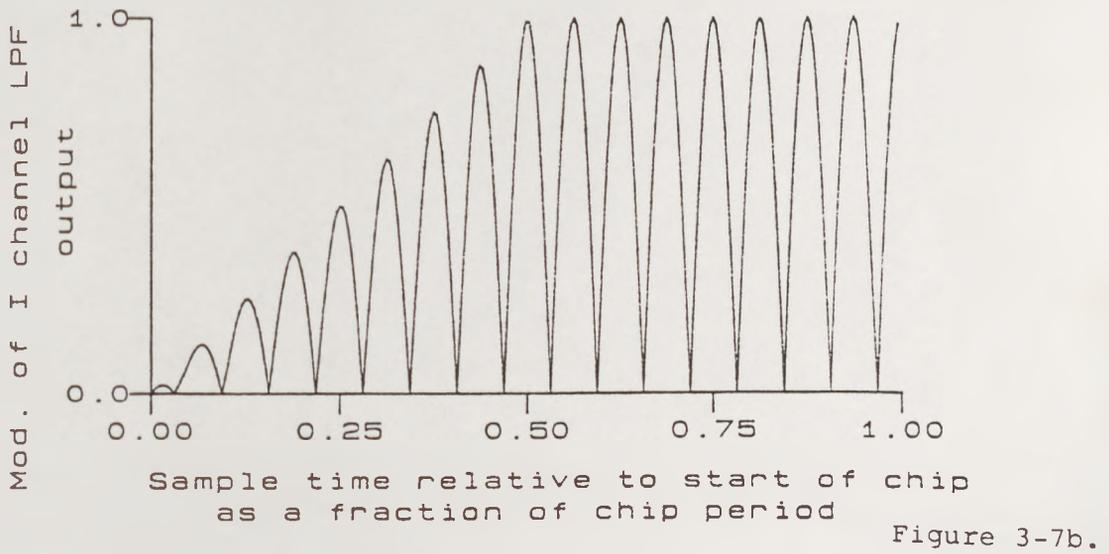
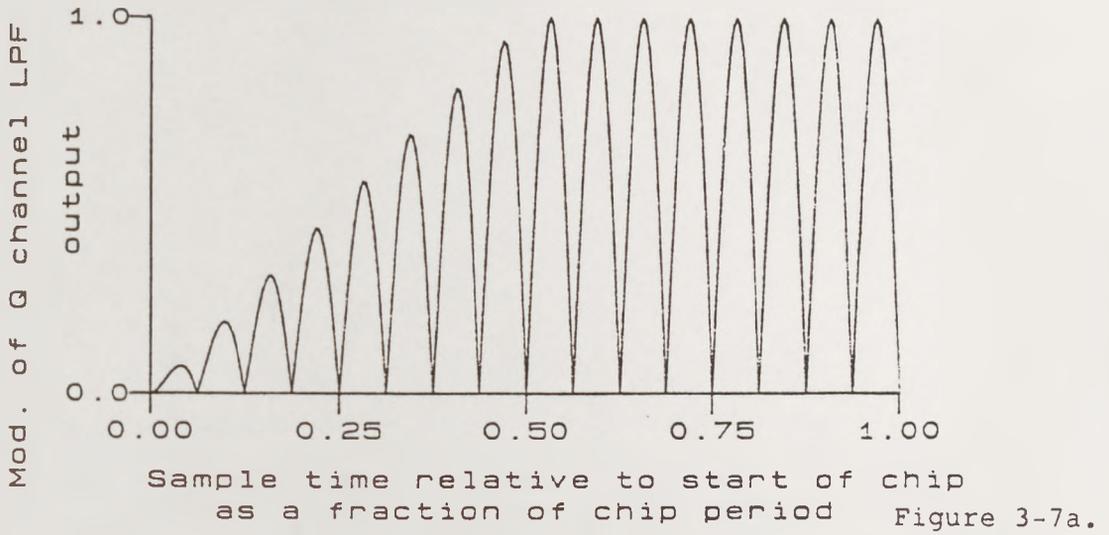


Figure 3-7. Lowpass filter risetime superimposed on mixer output null pattern.

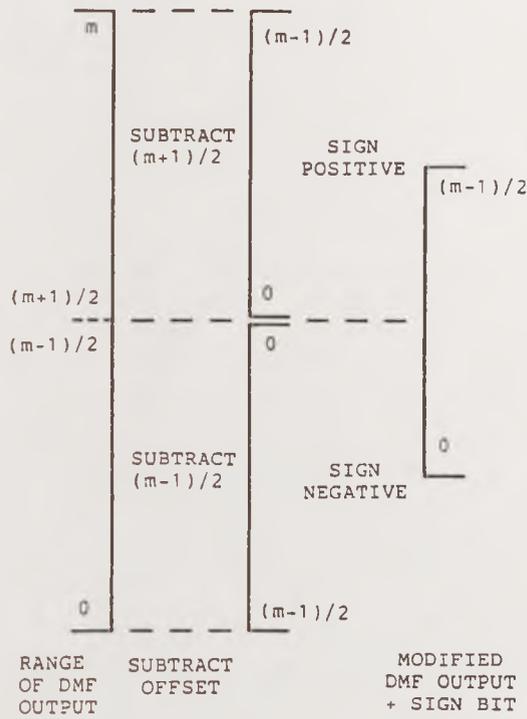


Figure 3-8. Algorithm to modify DMF output.

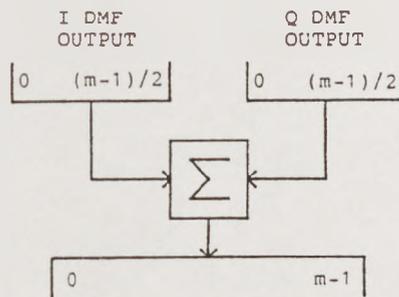


Figure 3-9a. Summing algorithm.

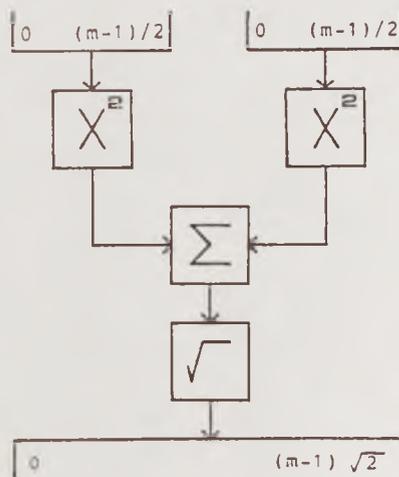


Figure 3.9b. Square root algorithm.

Figure 3-9. DMF combining algorithms.

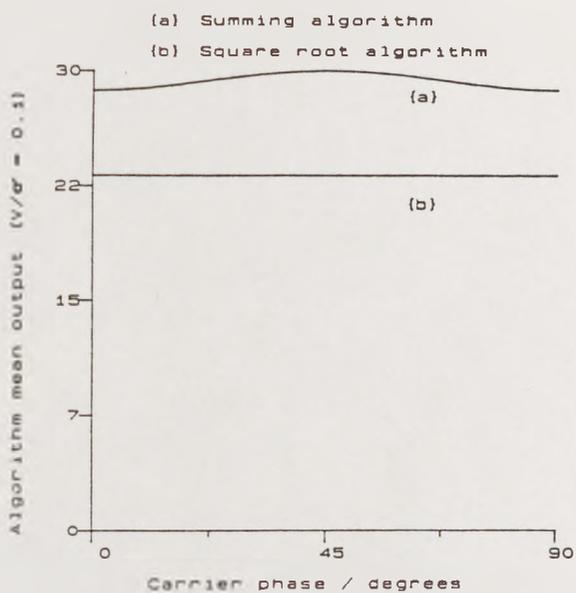


Figure 3-10a.

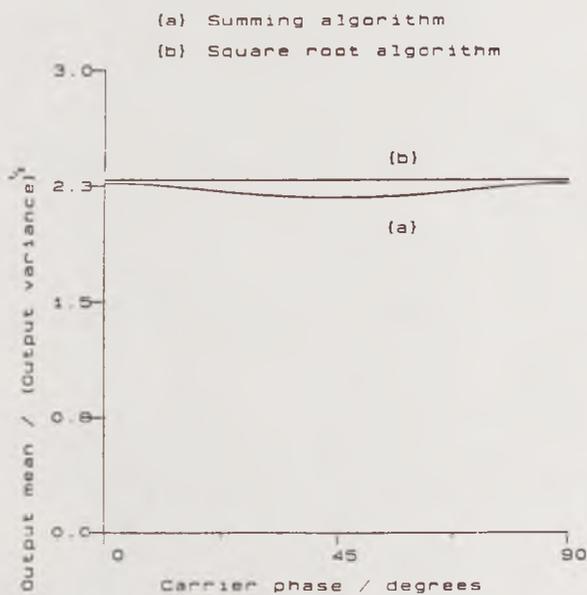


Figure 3-10b.

Figure 3-10. Output statistics for algorithms, constant snr, varying phase.

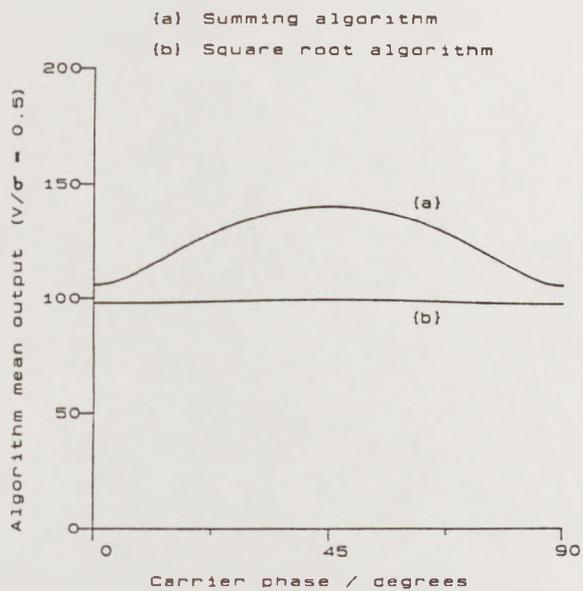


Figure 3-11a.

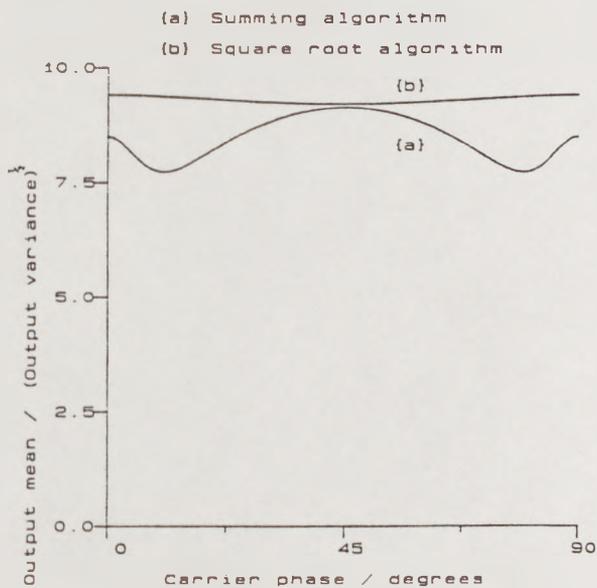


Figure 3-11b.

Figure 3-11. Output statistics for algorithms, $V/\sigma = 0.5$.

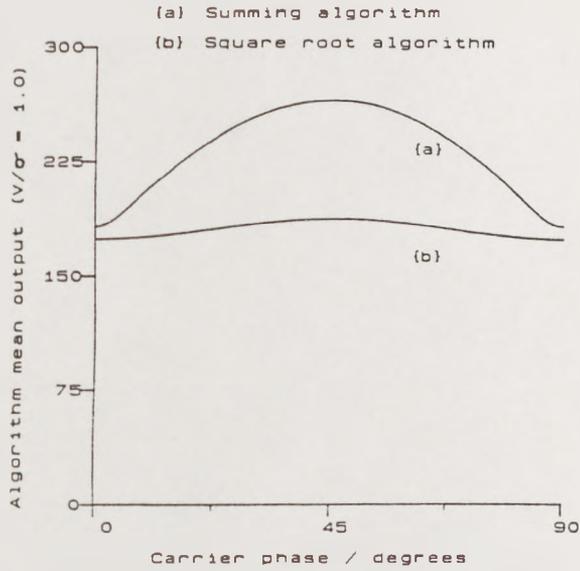


Figure 3-12a.

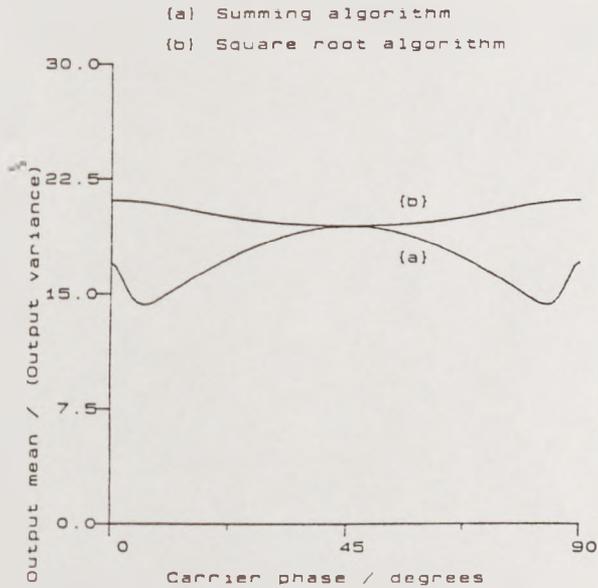


Figure 3-12b.

Figure 3-12. Output statistics for algorithms, $V/\sigma = 1.0$.

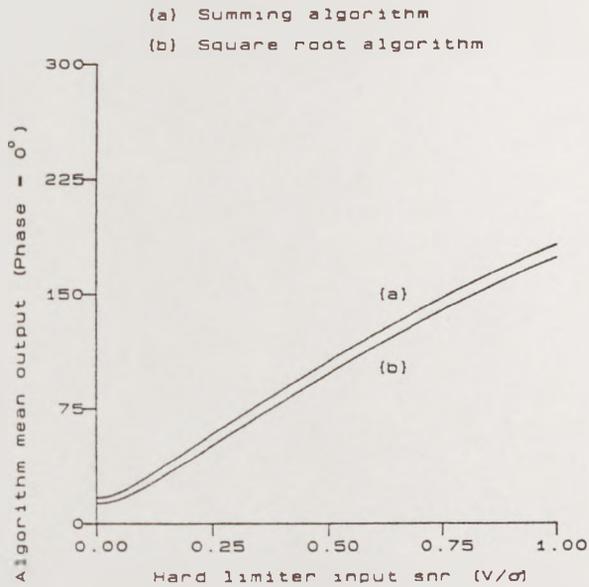


Figure 3-13a.

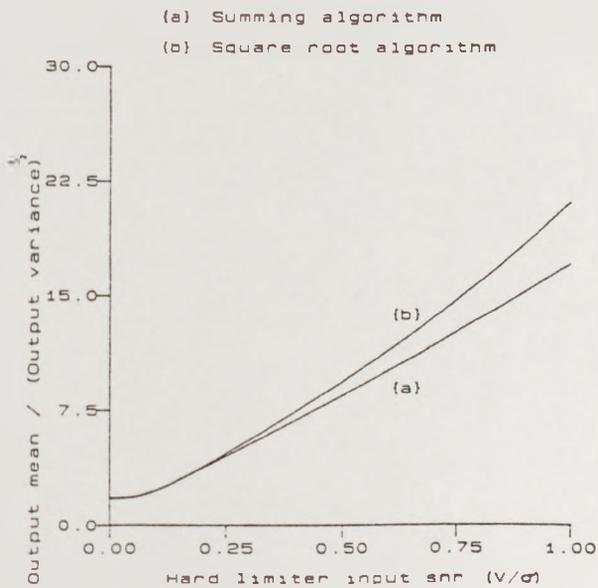


Figure 3-13b.

Figure 3-13. Output statistics for algorithms, increasing snr, constant phase (σ)

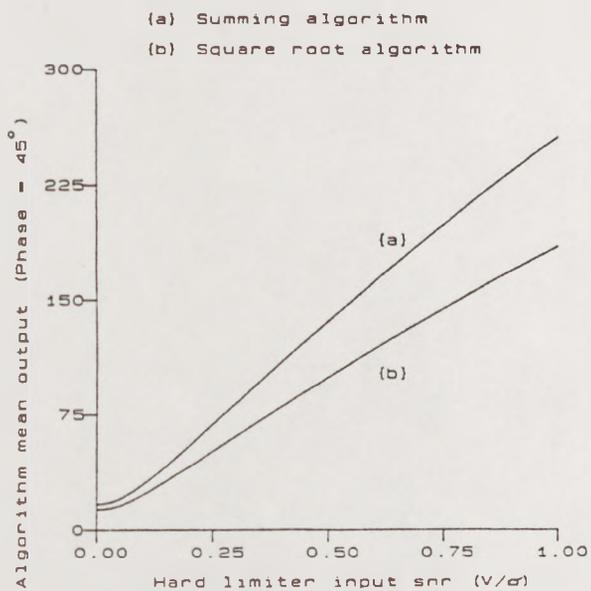


Figure 3-14a.

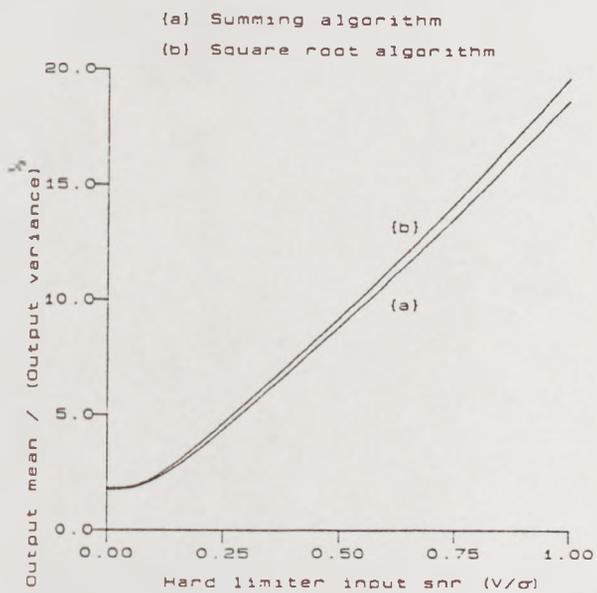


Figure 3-14b.

Figure 3-14. Output statistics for algorithms, increasing snr, constant phase(45°)

CHAPTER 4

SYNCHRONIZATION TECHNIQUES

4.1 INTRODUCTION

It has been shown that the most important operation performed in the DSSS receiver is correlation. The emphasis of the work described in this thesis has been the study of a digital approximation to this process, mainly from a practical viewpoint. The discussion is extended in this section to consider the important aspects of receiver synchronization.

There are three levels of synchronization in the DSSS receiver, code synchronization acquisition, code tracking, and RF recovery. Each is discussed in this chapter, and methods used by other authors are described in section 4.2. Several original techniques have been developed, particularly in the area of code synchronization acquisition and these are presented in section 4.3. These techniques utilise the power of the DMF to advantage in reducing the mean time to acquire code synchronization. In section 4.4 the techniques developed are applied to a study of the operation of a DSSS scheme in the Land Mobile Radio environment. The receiver's ability to acquire lock is investigated and the required receiver input snr is found. It is shown that in a Base/Mobile situation the DSSS system can operate in an overlay mode, and the effects of Doppler shifts on the received carrier can be compensated for at the Mobile receiver by the use of an analogue RF recovery loop. At the Base false carrier lock is prevented by utilising carrier phase information obtained from a pair of DMFs. The resulting scheme has the benefit that receiver complexity, in terms of RF recovery, is concentrated at the Base.

Digital techniques at baseband will be discussed as

these techniques are not presently applicable to bandpass signals due to speed limitations. Furthermore bandpass techniques are well covered in the literature.

4.2 SYNCHRONIZATION AND TRACKING SCHEMES

4.2.1 CODE SYNCHRONIZATION ACQUISITION

The first level to be considered is that of code synchronization acquisition. This may be defined as obtaining knowledge of the time of the received code epoch by the receiver. A detection decision can then be made at the code epoch, as discussed in chapter 2. This is the only phase of the code autocorrelation function at which data can be detected or demodulated, in the presence of noise, for a code with a uniform out of phase autocorrelation function close to zero.

The DSSS code is the key element in all DSSS systems. Knowledge of the code leads to the ability to despread the wanted information and spread unwanted information, thereby providing the desired processing gain. For this reason synchronization acquisition performance is often considered to be one of the most important receiver parameters. Therefore several techniques for obtaining code synchronization have evolved, and the literature is comprehensive, see for example [HOLMES 1982]. However the majority of these techniques are variations or refinements of the basic maximum likelihood detector, e.g. [WARD 1967]. This is shown schematically in figure 4-1.

In the maximum likelihood detector each code phase is examined by performing a separate correlation with a reference aligned to a unique code phase. If the code length is 511 chips the number of correlators required is 511, one for each code phase. Each correlator comprises a multiplier and an integrator. The input data is multiplied by a locally generated replica of the code, and the

integral is formed over the code period. This technique may be used for detection and, if implemented in analogue form with a real multiplier and integrator, is generally termed 'active correlation' as depicted in figure 4-2. A maximum likelihood detector selects the correlator with the maximum output at the end of the period of integration as corresponding to the phase of the code epoch. Clearly this can only be the receiver's estimate of the code phase in the presence of noise. Integration over several code lengths will improve the estimate and thus reduce the probability of false synchronization detection.

In a hard limited digital format, figure 4-3, the active correlator comprises a comparator (hard limiter), an Exclusive-NOR gate, and a resettable counter. At the beginning of each reference code the counter is reset. The local reference is continuously generated and the hard limited data is compared chip by chip with this. At the end of each reference code the sum of agreements, or correlation coefficient, is measured by the counter and this forms one input to the threshold detector. This technique is similar in principle to techniques used at bandpass, [HOLMES 1982].

One difference between the active binary correlator and the DMF is that the former can only produce a correlation output according to one code phase in each reference code period. The DMF produces a correlation output on every clock cycle and, if the reference code is static or fixed within the DMF, each correlation output corresponds to a unique code phase. Consequently for a code of length m an array of m active binary correlators is required to give the same amount of reference code phase information as a single DMF. Also each active binary correlator requires a continually generated reference code with a unique phase, although in practice these could be derived from a single generator with different feedback taps [GOLOMB 1967].

The previous discussion assumes optimum chip sampling. Prior to full synchronization this condition is unlikely to be met. For a baseband binary version of figure 4-1 it may be necessary to add parallel integrators with chip sampling delays to cover other possible chip phases. This is necessary to allow for the possibility of sampling close to chip transitions as discussed in chapter 3. Increased receiver complexity is an unavoidable consequence of this approach. A similar argument applies to the DMF implementation. For the analogue maximum likelihood synchronization scheme of figure 4-1 this problem does not arise since chip sampling does not occur.

For both the active binary correlator and the DMF the time taken to obtain the first estimate of reference code phase is one code period, and this is the minimum synchronization acquisition time for a code with a uniform out of phase autocorrelation function close to zero [SIMON et al 1985].

If these techniques are too complex or expensive the stepping correlator structure may be used. The stepping correlator, figure 4-4, reduces the number of active binary correlators required for an m bit code from m to 1. The single active binary correlator is made to search each possible code phase separately and in sequence. This technique exchanges the complexity and amount of hardware required to implement a full maximum likelihood structure with the increased time taken to achieve synchronization. The threshold detector at the integrator output, figure 4-4, is similar to that of the maximum likelihood synchronization scheme. Extra logic circuitry must be included to step the relative code phase at the end of the reference code if the search is unsuccessful. The process is repeated until synchronization is detected. The term 'stepping correlator' is used here as it seems more appropriate to a digital implementation, the term 'sliding correlator' is more widely found in the literature when applied to an analogue loop. In sliding correlator

synchronization schemes the time difference between the received code and the reference code is varied linearly in time. Such an approach is possible in an analogue implementation but for a digital approach the time difference must be stepped.

For a code of length m the mean time to synchronization is $mT/2$, where T is the code period. This assumes each code phase is searched once only, and that the probability of the search being initiated at any relative code phase is uniformly distributed with probability $1/m$. Also optimum chip sampling is assumed. If this is not the case, as might be expected during a cold start synchronization, it becomes necessary to step the search in fractions of a chip period rather than whole chip periods. For the binary stepping correlator this implies stepping the relative sampling point of the received data as well as the phase of the reference code. For example if four chip phases are to be searched for each code phase then the relative chip sampling point will change four times for each code phase change. If each possible chip phase is searched c times then the mean time to synchronization becomes $mcT/2$. If dwell is added to each phase to improve the false synchronization rate [SIMON et al 1985] the mean time to synchronization will increase by the dwell ratio.

For very long codes this technique leads to long average synchronization times. However since the stepping correlator requires considerably less hardware than either the full active binary correlator or the DMF it may be practical to have an array of c stepping correlators, each sampling the received data at a different chip phase. This would then lead to a reduction in the mean synchronization acquisition time from $mcT/2$ to $mT/2$ for systems not employing dwell techniques.

In the methods discussed in this section integration or averaging the threshold detector or maximum likelihood detector over several code periods would improve the

receiver's code phase estimate in the presence of noise. If noise is not present synchronization can be achieved with certainty in one code period, although this situation might not be regularly expected in a DSSS environment where operation in a multiple user mode would be at interference limited levels, and in covert applications operation would be below the receiver noise floor.

A technique which attempts to reduce both hardware complexity and mean time to synchronization acquisition is rapid acquisition by sequential estimation (RASE). This is a digital technique which attempts to estimate the state of the transmitter's code generator by loading an estimate of the received data chips into a local shift register. This register has identical feedback taps to the transmitter register and is also of the same length. Therefore both registers are of length z , and when the receiver has loaded z consecutive chip estimates into the local register the code estimate can be generated, since the z chips form an initial condition for the local code generator. Code synchronization is then tested and the process repeated until successful. If no errors have been made in the chip estimates the received code and the reference code will be in alignment, and therefore synchronization will have been achieved. This technique relies on the property that the next state in a pseudonoise sequence depends only on the present state of the generator, and that each generator state is unique [GOLOMB 1967].

A refinement of the RASE system is recursion aided rapid acquisition by sequential estimation (RARASE), [WARD and YIU 1977]. In this system the basic RASE technique is supplemented by a partial correlation between a number of received chips and the local reference. If the initial phase estimate was incorrect the cross correlation between these two sequences will be higher than would be expected if they were in phase. Therefore testing at this code phase can be terminated and the basic RASE process repeated until synchronization is achieved. The mean time to

synchronization is lowered by reducing the time spent searching over false code phases.

Both RASE and RARASE techniques are susceptible to noise and interference [SIMON et al 1985]. However, these techniques are most useful when very long codes are used. In this case the time taken for synchronization to be achieved by a sliding or stepping correlator could be extremely high. Also the construction of a very long DMF could prove too expensive, or power consumption could be a limiting factor. An area where these techniques are particularly useful is in the use of composite codes [ORMONDROYD and AL-RAWAS 1986]. Such codes are generally very long, being formed by a logical operation between two or more shorter sequences. The cross correlation properties of these codes are not as good as those of maximal length or Gold codes, and they are therefore not as useful in CDMA applications, [GOLD 1967], [BEALE and TOZER 1979]. However the potential reduction in synchronization acquisition time makes composite codes attractive for ranging and burst transmission schemes. In these schemes RASE and RARASE synchronization acquisition techniques prove useful provided the received snr is not too low [ORMONDROYD and AL-RAWAS 1986].

4.2.2 CODE TRACKING

Once initial code synchronization has been obtained the receiver must adjust the rate at which the local reference is generated to the rate at which the code is being received. This is necessary to maintain synchronization and allow data demodulation to occur. For a digital implementation this also constitutes an adjustment in the chip sampling rate since this is linked to the code generator clock. This process is termed code tracking.

Tracking is generally performed by one of two techniques, although several modifications and refinements of the techniques exist. The basic techniques are introduced here to allow comparisons to be made with the DMF implementation.

The first tracking technique to be discussed is the delay lock loop. The operation of this can be seen from figure 4-5a. The received data is correlated in two correlators, but the reference code is delayed in one correlator. The correlator output around the region of code synchronization is shown in figure 4-5b. If the output from the delayed correlator is subtracted from the non-delayed correlator an error signal is generated, figure 4-5c. This can be used to provide a correction voltage into the VCO, which can then force the code generator to track the incoming data once the loop is closed. For satisfactory operation initial synchronization must be accurate to within the region T-T', figure 4-5c. Outside this range the loop will lose lock and the initial code synchronization procedure will have to be repeated. An alternative implementation has advanced and retarded reference delays, although the characteristics are essentially the same, [ZIEMER and PETERSON 1985].

The second tracking technique uses dither to generate the error signal. Generally termed the Tau-dither loop this scheme has the advantage of not requiring a second correlator. The operation can be explained by examining figures 4-6a and 4-6b. The basic loop is similar to the delay lock loop but the error voltage into the VCO is a function of the correlator output and a dither signal. The dither signal causes the phase of the reference code to be switched between two values at the code rate. The correlator output is measured at both phases and the VCO is adjusted until the error signal is the same for each phase offset. At this point the basic code generator would be tracking the incoming code if dither were not present. This indicates one problem associated with the dither loop.

Since the reference code is switched between symmetrical points of the optimum phase there is always some loss in correlation [SIMON et al 1985], since correlation never occurs at the optimum code phase. This problem can be overcome with an additional correlator, but the resultant implementation will have a similar hardware efficiency as the delay lock loop.

Both techniques are applicable to baseband digital implementations. Further discussion on these techniques can be found in the literature.

4.2.3 RF RECOVERY

The techniques described for code synchronization acquisition and tracking require coherent signal recovery at RF [COOPER and MCGILLEM 1986]. It is therefore necessary to consider techniques for acquiring a coherent RF carrier from the received spread spectrum signal. Several methods are described in the literature, and two popular techniques are briefly discussed here.

For simplicity it was decided that the practical system, which is described in chapter 5, would operate with PSK modulation. The RF recovery loops to be considered should therefore be capable of operating with this form of modulation imposed on the carrier.

The squaring loop, figure 4-7a, forms the square of the modulated input signal and this is then bandpass filtered to remove noise. The phase modulation is also removed by the squaring process. The signal at the square law operator output contains a component at twice the input signal frequency. This is then used as the input signal to a basic phase locked loop, operating at twice the input carrier frequency. A frequency division of two on the VCO output results in recovery of a carrier at the input signal

frequency.

The Costas loop, figure 4-7b, is a quadrature loop technique. The loop produces an error signal which is proportional to twice the sine of the carrier phase error. The VCO characteristic is chosen such that when the loop is closed the phase error between the in phase channel and the input signal carrier approaches zero. This has the advantage that the input signal can also be demodulated to baseband, resulting in a possible hardware saving.

Both loops will exhibit random carrier phase fluctuations when the input signal is corrupted with noise. In both cases the variance of the phase fluctuations due to noise is proportional to the loop bandwidth, and inversely proportional to the input snr [SIMON and LINDSEY 1977].

The two techniques, the squaring loop and the Costas loop, are frequently used for coherent carrier recovery in many phase modulation schemes, such as PSK [HOLMES 1982]. However, the analysis of the performance of these loops is complicated in DSSS systems since several spread spectrum signals may be received in an overlay mode. A further complication may arise if the carrier frequencies of the various received signals differ slightly. This situation could arise due to Doppler shifts generated by the relative movement between transmitters and receivers. Since the loop does not have the benefit of the potential processing gain of the DSSS receiver other techniques may be required for carrier recovery. This is discussed in section 4.3.2.

4.3 DIGITAL SYNCHRONIZATION SYSTEM

This section describes techniques for code synchronization acquisition, tracking, and RF recovery using digital techniques at baseband.

4.3.1 DMF RECEIVER CODE SYNCHRONIZATION ACQUISITION

Unlike the sliding or stepping correlator schemes the DMF produces a new correlation value for each new chip sample accepted. Similarly the combining algorithm must operate at this rate. In the case of the sliding correlator the synchronization test instant is determined locally, since it coincides with the start or end of the locally generated code. Therefore it is independent of the received data stream. At present digital matched filtering devices are expensive compared to the technology required to implement a bandpass sliding correlator structure. Therefore if digital techniques are to be adopted they must provide some positive benefits with respect to current well established techniques. This might then lead to a more widespread adoption of such techniques which would undoubtedly lead to a reduction in costs.

One desirable benefit would be the reduction of average time to synchronization. In this section a new technique for the synchronization of a DMF based DSSS receiver is discussed. This technique makes use of the output available from the DMF combining algorithm on every new chip, and this leads to a lower average synchronization time.

In the baseband digital DSSS receiver it is necessary to determine the correct instant at which to sample the sign of the DMF output and hence form a detection decision with a finite error probability. For the synchronization scheme under discussion the complex baseband receiver as discussed in chapter 3 is used. This allows for carrier phase offsets which may arise due to noise, as discussed in section 4.2.3 The square root algorithm is chosen to combine the DMF outputs, as it was shown that this produces an output which is largely independent of local oscillator phase.

We begin by looking at the probability that the

algorithm output exceeds a particular output threshold K for the code in phase and out of phase conditions. This has been evaluated for three values of snr, -20 dBs, -15 dBs, and -10 dBs, and shown in figures 4-8a to 4-10a. These values of snr correspond to chip error probabilities of approximately 0.44, 0.40, and 0.33 respectively for BPSK modulation and AWGN noise. With an available processing gain of some 27 dBs (less 2 dBs for hard limiting - chapter 2) this gives equivalent post processing gain snrs of 5 dBs, 10 dBs, and 15 dBs. The values were chosen to give a realistic range of input and output snrs such that chip and symbol error probabilities could be measured practically. From these figures it is clear that, for a given probability of threshold crossing, the in phase condition will exceed a higher threshold at constant snr. This might suggest that a form of simple 'threshold detection' be used, whereby the data is demodulated by considering all outputs exceeding a suitably chosen threshold as representing code in phase conditions. Data demodulation would then be performed by looking at the sign of the DMF outputs. Such a technique might prove useful at higher snrs, figure 4-9a, but would lead to very high symbol error probabilities if the out of phase error probabilities are to be kept low, figure 4-8a. Further this does not take into account the number of occurrences of the out of phase condition.

For a code of length m there are $m-1$ out of phase conditions and it is necessary to consider a cumulative out of phase probability of exceeding a particular threshold. For the practical system under development the code length is 511 giving 510 out of phase conditions. If the probability of a threshold crossing for any out of phase condition is assumed to be equal, and denoting this P_K for a threshold value of K , then the probability of not obtaining a threshold crossing is $1-P_K$. The probability of obtaining zero threshold crossings in 510 code phases is ${}^{510}C_{510} (1-P_K)^{510} = (1-P_K)^{510}$. Therefore the probability of obtaining a false threshold crossing is this value

subtracted from unity, $P_f = 1 - (1 - P_K)^{510}$. The values presented in figures 4-8a to 4-10a have been incorporated in figures 4-8b to 4-10b to reflect the number of out of phase conditions. The figures show the probability of the in phase condition occurring once in 511 chips, or one code length, and the probability of any out of phase condition occurring in the same 511 chips.

These figures show that at low snrs the cumulative probability for the out of phase case exceeds that for the in phase case for certain threshold values, and is particularly severe for values of K less than about 50 in figure 4-8b. In this region the probability of correct decision (P_c) does not exceed the probability of incorrect decision (P_f) until P_c is very low. If we use simple threshold detection this condition would lead to false detection signals in the out of phase case and missed detection signals in the in phase case, although the system might be expected to perform better at higher snrs, as shown in figure 4-10b. This arises since all out of phase conditions are considered to be part of a collective alternative to the in phase condition, rather than one of 510 unique alternatives.

From figures 4-8a to 4-10a it is clear that we must develop an approach which treats each out of phase condition as an isolated alternative to the in phase condition. This will then lead to the error probability statistics of figures 4-8a to 4-10a, which are clearly superior to those of figures 4-8b to 4-10b from a threshold detection viewpoint. Three techniques will now be considered.

4.3.1.1 THRESHOLD DETECTION ON AVERAGE ALGORITHM OUTPUT

In this technique the average algorithm output for each of the possible output code phases is calculated. This could be performed by storing the last n outputs for each phase and performing an averaging operation on these. This is equivalent to a lowpass filtering or integration operation over a fixed period of time.

The practical system under investigation would require an array of 511 by n elements, where each element could be a number in the range 0 to 361 . This range covers all possible outputs from the combining algorithm, and is shown schematically in figure 4-11 where, for clarity, the figure shows the system configuration for $n=4$. The past 4 outputs of each code phase are stored in a recirculating memory, or shift register array. The average of these 4 outputs is taken, again for each code phase. If chip samples were taken in AWGN the DMF output statistics will be as determined in chapter 2. Therefore the algorithm output statistics will be as determined in chapter 3. If a long period of integration is chosen, that is n becomes large, then the average output for each code phase will tend to the expected output at that code phase as determined by the code autocorrelation function. Thus the effects of the noise will be reduced by averaging. Under these conditions the in phase case will show a positive bias against all out of phase cases, assuming the code has low or zero out of phase autocorrelation values. This bias may then be detected by a threshold decision on the outputs of the averagers, shown schematically in figure 4-13. This technique is equivalent to the maximum likelihood detector discussed in section 4.2.1.

At present this approach poses problems for a practical implementation but improvements in digital signal processing and VLSI techniques will undoubtedly make such an approach feasible. The problems encountered are more likely to be ones of integration rather than speed, since

each averager only needs to perform the averaging operation at the code or symbol rate, which in this case would be only 1/511th the chip rate. Since this technique averages out the effects of the noise it could also be used to detect the presence of delayed signals due to effects such as multipath propagation. Some practical results along these lines are given in chapter 6.

This technique required an array capable of storing the full algorithm output range, 0 to 361 for the practical system discussed. The result is a wide memory structure, 9 bits being required to cover the full range of the algorithm output, and relatively high complexity adders. Although such a technique could be implemented in VLSI techniques the particular device or devices required are not readily available. If future DSSS receivers adopt a digital approach to synchronization and detection then this technique may be developed further.

4.3.1.2 AVERAGE OF NUMBER OF THRESHOLD CROSSINGS

In the technique examined in this section the algorithm output is compared with a threshold and the result of this comparison is stored as a single bit number in the array, or shift register. The array length must still be 511, corresponding to the number of code phases. However the amount of data stored is reduced from 511 by n by 9 bits to 511 by n by 1 bit. Therefore the past n outputs for each code phase are stored as the result of n threshold comparisons, figure 4-12, where the output from each comparator block is a single binary digit. This reduces the amount of storage required, and also significantly reduces the amount of processing required on the array values. In common with the previous technique described this approach also requires code epoch detection, shown schematically in figure 4-13.

Detection by the average number of threshold crossings

technique has been evaluated and some results are shown in figures 4-8c to 4-10c for $n = 4$. Here the probability that 3 or 4 successive outcomes will be 1, corresponding to 3 or 4 threshold crossings, has been evaluated for each algorithm output probability.

These are found from

$$\text{Prob}(\text{Sum}=1) = \binom{3}{4} p_1^3 (1-p_1) + \binom{4}{4} p_1^4 \quad (4-1)$$

where p_1 is the probability that the algorithm output exceeds the threshold set, as given in figures 4-8a to 4-10a. and $\text{Prob}(\text{Sum}=1)$ is the probability that 3 or 4 of the successive outcomes will be 1.

These statistics can only be expected if the four algorithm output samples are independent. This will be the case provided the chip samples are independent, and this assumption is made throughout the discussion. The result of this is a delay of four symbols before the next sum can be formed, since only then will each sample be independent of any samples taken in the previous sum. In a practical implementation such a decision strategy could be realised with simple AND gates and shift registers, removing the need for complicated adder structures. This corresponds to a maximum likelihood detection, based upon a simple binary integration technique.

The results, shown in figures 4-8c to 4-10c, indicate that the curves are sharpened up. The effect of this can clearly be seen from some sample values shown in table 4-1.

TABLE 4-1
 PROBABILITY THAT ALGORITHM OUTPUT EXCEEDS THRESHOLD 'K'
 SINGLE SAMPLE AND 4-BIT BINARY INTEGRATION

K	Probability algorithm output > threshold		Prob(Sum=1) 4 bit binary integration	
	In phase	Out of phase	In phase	Out of phase
	Chip $P_e = 0.3274$		$(E_c/N_o = -10\text{dBs})$	
49	0.999764	4.50497×10^{-5}	1.00000	3.65697×10^{-13}
51	0.999652	1.95442×10^{-5}	0.999999	2.98612×10^{-14}
53	0.999444	8.10824×10^{-6}	0.999998	2.13225×10^{-15}
	Chip $P_e = 0.4007$		$(E_c/N_o = -15\text{dBs})$	
22	0.996637	0.140551	0.999933	9.93541×10^{-3}
28	0.985208	4.22734×10^{-2}	0.998713	2.92597×10^{-4}
33	0.957732	1.25105×10^{-2}	0.989875	7.75877×10^{-6}
	Chip $P_e = 0.4437$		$(E_c/N_o = -20\text{dBs})$	
13	0.955277	0.508227	0.988703	0.32494
18	0.884202	0.270212	0.950251	9.35009×10^{-2}
23	0.766415	0.120451	0.765656	6.35869×10^{-3}

These figures indicate that 4 bit binary integration significantly reduces the out of phase probabilities, particularly for high snrs. It is clear that simple threshold detection could be successfully used as a method of data demodulation for OOK modulation if the input snr was greater than -15 dBs. For example, with the tabulated values for 4 bit binary integration a threshold of $K = 28$ would give a probability of false detection of about 3×10^{-4} , and probability of missed correct detection of about 10^{-3} with $E_c/N_o = -15$ dBs. Combining these probabilities we find the equivalent output snr from chapter 2 is $E_s/N_o = 6.4$ dBs, where E_s is the equivalent symbol energy for BPSK modulation, and N_o has the usual definition.

For the practical system we would expect an output snr of

$$(\text{snr output})_{\text{dBs}} = (\text{snr input})_{\text{dBs}} + \text{PG} - \text{LOSS}_{\text{hl}} \quad (4-2)$$

where (snr input) is the chip snr, PG is the maximum processing gain in dBs, $10 \log(m)$, and LOSS_{hl} is the loss due to hard limiting, again in dBs. For the practical system this would give $-15 + 27 - 2 = 10$ dBs. Therefore the simple threshold detector has only introduced an effective loss in snr of about 3.6 dBs.

These figures do not allow for the effects of RF phase variation and should be considered optimistic. Further unless RF phase coherence is maintained data recovery is not possible in systems employing SIK modulation since the algorithm output is a measure of the modulus of the code correlation but gives no information of the sign. Thus further processing is necessary, and the data may be recovered from the DMF outputs rather than the algorithm output. However if OOK modulation were used it is highly

likely that a satisfactory system could be operated under the given conditions.

The figures of table 4-1 would give acceptable results in certain systems, and would naturally give better results at higher snrs. It can be seen that system performance depends upon the choice of threshold K . If operation at low input snrs is required, where the output snr post processing may be of the order of a few dBs, then K must be set to a low value to allow detection of the in phase peak. However this results in a significantly reduced error rate performance due to the increased probability of obtaining a false peak at one of the code out of phase positions. The figures suggest that if input snrs are high enough such that the post processing snr is of the order of half the available processing gain then the value of K can be set to a higher level and improved error rate performance can be expected.

Increasing the number of samples taken for binary integration will improve synchronization error rate performance, at the expense of synchronization time. This follows from a reduction in the variance of the binary integrator output by the process of averaging allowing the decision threshold of the maximum likelihood detector to be set to a lower value for a given false synchronization probability, at which point there will be an increase in the probability of correct synchronization detection, and hence improved system synchronization error probability performance.

This technique requires 511 binary integrators and a maximum likelihood detector of the form shown in figure 4-13. It is clearly a much simpler system than the algorithm output averaging technique described in section 4.3.1, requiring less hardware. This would result in fewer practical problems for a VLSI implementation, although the system might be expected to offer a somewhat reduced performance particularly at low input snrs.

4.3.1.3 SINGLE THRESHOLD CROSSING WITH BINARY INTEGRATION

The previous technique of binary integration using threshold estimates on the comparator output gives a considerable saving in hardware over the averaging technique described in section 4.3.1.1. However it is still necessary to construct m binary correlators and maximum likelihood detectors (or 3 or 4 from 4 detectors) for a code of length m . This is feasible, and a static RAM or shift register array could be used in practice. If an array of shift registers were used the complexity of this would approach the complexity of the DMF itself. Therefore a new and much simplified technique, single threshold crossing with binary integration, has been devised and is discussed here.

This technique replaces the m binary integrators of the previous technique with a single binary integrator and decision logic. The operation of the system can be seen from figure 4-14. The system is analysed assuming a 4 bit search, which requires a 3 bit shift register, and an output detector which gives an output of 1 (or true) if the shift register contains 2 or more 1s. Initially the shift register is reset. The system is activated by the first threshold crossing of the algorithm output. This will either be due to the in phase correlation peak, or a false peak due to noise at one of the $m-1$ out of phase cases. At this point the shift register clock is enabled by the control block initiating a synchronization search. The shift register clock is running at a frequency of $1/m$, compared to the chip clock, the SR clock being termed the symbol clock. The next sample into the shift register occurs at the same code phase of the received data as the initial threshold crossing. This process is repeated until the register contains 3 successive samples after the initializing crossing and then the threshold detector output is sampled. For the case discussed the threshold

detector output is set to a logic 1 if there are 2 or 3 logic 1s in the shift register. If this output is true then synchronization is assumed, with a given error probability, and control is passed to the tracking phase by assertion of the synchronization detection flag. If the output is not true then the initial threshold crossing is assumed to have been due to an out of phase correlation peak resulting from noise. The system is reset and the process must then be repeated.

In practice the shift register length need only be 3 bits, since the first threshold crossing need not be stored. The decision logic then only needs to test the 3 bit array for 2 or 3 threshold crossings, figure 4-14. If a false synchronization has occurred it will be necessary for the code tracking section to abort and return control to the acquisition mode.

This may appear a somewhat crude approach, and in some senses may be likened to the RASE approach discussed in section 4.2.1. However it was shown in section 4.3.1.2 that the binary integrator with maximum likelihood decision has improved output statistics in terms of probability of threshold crossing in the out of phase case relative to the statistics at the threshold comparator input. Therefore for a given minimum input snr a suitable threshold may be selected such that the probability of error at the binary integrator output is extremely low, that is correct outputs will be detected with a very high degree of certainty, and false outputs will similarly be rejected. This allows the code synchronization decision to be made with a very high level of confidence, since the benefit of the gain introduced by the binary integration process has been utilised along with the processing gain of the DMFs. The statistics determining the probability of starting the estimation process at the correct code phase are defined by the basic probability of threshold crossing discussed in section 4.3.1.

The average time to synchronization can now be found. Assuming that a synchronization test may begin at any code phase we may find the average number of tests required for each phase. We define p_o to be the probability that the threshold is crossed for any out of phase condition, and assume that this remains constant for all out of phase conditions. Further p_i is defined as the probability that the threshold is crossed for the in phase case. For an arbitrary code phase offset of n chips the probability that the correct phase will be reached without a false threshold crossing in the preceding $n-1$ out of phase cases is $(1-p_o)^{n-1}$. Further the probability that the in phase condition, if reached, will cause a correlation peak is p_i , giving a total probability of success of $p_i(1-p_o)^{n-1}$ at this phase.

Once a potentially successful search has been initiated the probability that the binary integrator output will detect a successful search is p_b , where p_b is defined in (4-1). Therefore the average number of tests required to give success at this phase is given by $1/p_b p_i (1-p_o)^{n-1}$. The average number of tests for all m phases can be found by summing the average for each phase giving, since each phase is assumed equally likely,

$$\text{Mean number of searches} = \frac{1}{m} \sum_{n=1}^m \frac{1}{(p_b p_i (1-p_o)^{n-1})} \quad (4-3)$$

The average synchronization search time can be found by evaluating the average search time for each code phase.

If t_c is the chip period and t_s the code period, where $t_s = mt_c$, the average time to search each phase is

$$t_n = (nt_c + 3mt_c) / (p_b p_i (1-p_o)^{n-1}) \quad (4-4)$$

The term $3mt_c$ arises since binary integration of 4 samples takes 3 further code periods after an initial threshold crossing has initiated a search. The term nt_c is the time taken to reach the in phase code position, which is assumed to be n chips away.

Assuming each code phase is equally likely at the start of each search attempt, that is no memory of rejected code phases is assumed, the average time to synchronization, T_{Av} , may be found by averaging over all code phases giving

$$T_{Av} = 1/m \sum_{n=1}^m (nt_c + 3mt_c) / (p_b p_i (1-p_o)^{n-1}) \quad (4-5)$$

$$= 1/m \sum_{n=1}^m (nt_s/m + 3t_s) / (p_b p_i (1-p_o)^{n-1}) \quad (4-6)$$

chip periods.

This analysis assumes that the binary integrator will always reject false searches initiated by out of phase correlation peaks. This a reasonable assumption since table 4-1 indicates that the out of phase threshold crossing probabilities are heavily suppressed by the action of the

binary correlator for high threshold settings. Indeed this is the primary purpose of using binary integration. For the '3 or more from 4' integrator all probabilities below about 0.7 are reduced, whereas those above about 0.7 are increased, as might be expected in an 'average' sense.

Equation (4-6) has been evaluated for some sample values of p_i and p_o , to highlight the effect of varying these parameters. The results are shown in table 4-2.

TABLE 4-2
 MEAN NUMBER OF SYNCHRONIZATION SEARCHES
 SINGLE THRESHOLD CROSSING WITH BINARY INTEGRATION

Probability		Mean number of searches
In phase	Out of phase	
1.0	0.0	4
1.0	0.001	5
1.0	0.01	125
1.0	0.1	1.69×10^{22}
0.8	0.0	5
0.8	0.001	7
0.8	0.01	190
0.8	0.1	2.58×10^{22}
0.9	0.01	147
0.8	0.01	191
0.7	0.01	274
0.6	0.01	438
0.5	0.01	799
0.4	0.01	1742
0.9	0.0	5
0.8	0.0	6
0.7	0.0	8
0.6	0.0	13
0.5	0.0	23
0.4	0.0	49
0.3	0.0	140
0.2	0.0	644
0.1	0.0	947

The values in table 4-2 are presented in 2 groups. The top group, for in phase probabilities of 1 and 0.8, show the mean number of searches with varying out of phase probabilities. The bottom group shows the mean number of searches against varying in phase probability but for the 2 values of out of phase probability of 0.01 and 0.

The results of table 4-2 suggest that it is more important to maintain a very low probability of false threshold crossing, rather than maintaining a high probability of correct threshold crossing. This would suggest setting the threshold level at a high value, to ensure low probability of threshold crossing for the out of phase condition. If the out of phase threshold crossing approaches 0.1 the mean time to synchronization becomes impractically large. If this figure can be kept to below 0.01 then, for in phase threshold crossings greater than 0.6, the mean time to synchronization is comparable to that expected from of a sliding correlator under the same conditions of input snr [HOLMES 1982]. If conditions are more favourable, corresponding to higher input snrs, the mean time to acquisition will be reduced. For a given probability of synchronization error this is not the case for the sliding correlator.

This simple approach could have benefits in a multiple user environment, where a calling channel might be used at a higher snr when compared to other users of the channel. Under this condition the simple single threshold crossing with binary integration technique could be optimised for acquisition of the calling channel.

The main factor contributing to the excessive mean time to synchronization for higher out of phase threshold crossing probabilities is the length of the sequence. If m is large the probability of a false threshold crossing before the correct phase is reached is high, resulting in a large number of searches on average. To reduce this effect

the number of detectors may be increased. This is shown in figure 4-15, where the number of circuits has been increased from 1 to 4. Each detector searches one quarter of the possible code phases, and all other phases are gated out. A four phase clock is required, to provide the gating signals, and this can be incorporated within the control block. Since the sequence search length, m , is now effectively reduced to $m/4$ the mean time to synchronization is dramatically reduced. This is shown in table 4-3. Two chip error probabilities are considered, $p_e=0.4437$ and $p_e=0.4007$, corresponding to input snrs of -20 dBs and -15 dBs for BPSK modulation. The probability of threshold crossing has been evaluated for the in phase and out of phase conditions and for several threshold levels.

The use of 4 detectors reduces the mean time to synchronization considerably, and for an input snr of -20 dBs the system can out perform RASE, RARASE, and sliding correlator synchronization schemes by a factor of at least 4 [ORMONDROYD and AL-RAWAS 1986], [ORMONDROYD and COMLEY 1984], [WARD and YIU 1977]. In many situations the improvement factor will be considerably higher than this. In common with the RASE and RARASE systems the synchronization acquisition time of the binary integrator search technique depends on the input snr. However the improvement in performance of the new system discussed is due to the processing gain obtained from the DMFs which considerably reduce the effects of noise.

The analysis has assumed a uniform autocorrelation function for the code. In practice the effects of modulation, code cross correlation peaks, and multipath propagation will reduce the performance of the system. However the degradations imposed by these parameters are common to all systems.

TABLE 4-3
 MEAN NUMBER OF SYNCHRONIZATION SEARCHES
 SINGLE THRESHOLD CROSSING WITH ONE AND FOUR
 BINARY INTEGRATORS

Alg. output t'hold	Mean number of searches			
	$p_e = 0.4437$		$p_e = 0.4007$	
	Number of integrators			
	1	4	1	4
24	3.21×10^{22}	3.10×10^5	3.10×10^{21}	1.10×10^5
25	3.93×10^{18}	3.93×10^4	3.37×10^{17}	1.31×10^4
26	6.24×10^{13}	6.15×10^3	7.22×10^{13}	1.88×10^3
27	9.08×10^{11}	1.59×10^3	1.06×10^{11}	437
28	5.89×10^9	600	6.85×10^8	145
29	6.55×10^7	270	7.39×10^6	56
30	2.67×10^6	166	2.77×10^5	29
31	1.63×10^5	116	1.50×10^4	17
32	2.31×10^4	99	1.83×10^3	11
33	6.02×10^3	96	388	9

TABLE 4-3 (continued)
 MEAN NUMBER OF SYNCHRONIZATION SEARCHES
 SINGLE THRESHOLD CROSSING WITH ONE AND FOUR
 BINARY INTEGRATORS

Alg. output t'hold	Mean number of searches			
	$p_e = 0.4437$		$p_e = 0.4007$	
	Number of integrators			
	1	4	1	4
34	2.13×10^3	102	111	7
35	1.05×10^3	119	43	6
36	729	151	22	5
37	647	204	14	5
38	673	284	10	5
39	798	422	8	5
40	1.05×10^3	645	7	5
41	1.38×10^3	1.02×10^3	7	5
42	2.37×10^3	1.75×10^3	7	5

4.3.2 DIGITAL TRACKING AND RF RECOVERY LOOPS

Code tracking may be performed in a DMF based DSSS receiver by either of the techniques discussed in section 4.2.2. The performance of either technique will not be enhanced by the presence of the DMFs, since during the code tracking phase the DMF output is sampled only at the code epoch. Therefore a simple binary integrator of the form of figure 4-3 will provide as much information at the code epoch as obtained from the DMF. A DMF based DSSS receiver using the delay lock loop tracking scheme has been discussed in the literature [OULD and VAN WECHEL 1981], although the code tracking loop requires an increase in the number of samples taken per chip. If each chip is sampled twice it is necessary to double the length of each of the DMFs for each half of the complex baseband receiver, to provide early and late signals. From this it is clear that the power of the DMF is not really exercised during the code tracking phase, since maximum benefit is obtained from this device when an output is required on each chip sample.

RF recovery may be achieved by the squaring or Costas loops described in section 4.2.3. If the signal is received in noise the VCO output will exhibit random phase fluctuations. To overcome this additional control of the VCO is required, using information obtained at the DMF outputs.

The modified DMF outputs for low input snrs, from chapters 2 and 3, are

$$\text{Mean I output} = m / (2\pi)^{1/2} \sqrt{V/\sigma} \cos\phi \quad (4-7)$$

$$\text{Mean Q output} = m / (2\pi)^{1/2} \sqrt{V/\sigma} \sin\phi \quad (4-8)$$

where θ is an arbitrary phase offset between the received signal and the local oscillator.

From this we may obtain

$$\theta = \arctan (\text{Mean } Q / \text{Mean } I) \quad (4-9)$$

Therefore when the input snr is low, and hence approximations (4-7) and (4-8) may be applied, the average DMF outputs at the code epoch can be used as an unbiased estimator of the relative carrier phase. If the input snr is high the approximations are not valid and the value of θ tends to 45 degrees. It then becomes a biased estimator of carrier phase [GOLD 1978]. Correction of the local carrier phase may therefore be achieved by measuring the phase offset as a function of the DMF outputs and applying a suitable correction signal to the VCO. If the input snr is high phase correction may not be required as phase fluctuations will be lower. From chapter 3 it was shown that the algorithm output gives a measure of the input snr and this may be used to decide how much phase control is required.

It was also shown in chapter 3 that a preferred relationship exists between the ratio of the RF carrier frequency and the chip rate. If this ratio is an integer then a repetitive null pattern is obtained at the I and Q channel lowpass filter outputs. This pattern also includes the effects of the lowpass filter risetime on the snr at the A/D inputs. At the receiver the integer ratio removes the need for code tracking. If stable RF recovery is achieved then, from figure 3-7, the relative carrier phase remains constant, and therefore the sampling point relative to the start of each chip must also remain constant. Slight carrier phase fluctuations will result in slight movements in the relative sampling instant, but this will not cause problems provided sampling does not occur close to chip edges.

One practical advantage of using an integer ratio between carrier and chip clock frequencies is the simplification of receiver hardware. Synthesis of the RF carrier from a stable source is simplified if an integer division ratio can be used, avoiding the need for a double loop approach [COOPER 1974].

4.4 SYSTEM STUDY OF CONTROL OF A DMF BASED DSSS RECEIVER

Having investigated the operation of each subsection of the DMF based DSSS receiver we may now consider the receiver as a system. This brings together the synchronization, tracking, and RF recovery operations. This study will consider a DMF based DSSS receiver, and outline a practical implementation of the necessary processes required to successfully operate such a system when the received signal is corrupted in AWGN. Emphasis is placed on a lack of stability between the transmitter and receiver local oscillators, and 'cold start' conditions, therefore assuming a lack of synchronization at all levels.

The proposed system consists of a simple DSSS transmitter, figure 4-16, and receiver, figure 4-17. The transmitter is of a conventional format, with digitization of the message signal at the symbol rate, generation of an m-sequence also at this rate, with m chips transmitted within the symbol period. The RF synthesiser is locked to the chip rate oscillator, or alternatively the chip clock is derived from the RF oscillator by frequency division. Both techniques are intended to result in an integer ratio between the RF and chip frequencies, the diagram indicates the frequency synthesizer approach.

RF modulation is conventional BPSK for simplicity, and data modulation is achieved by inverting the digital m-sequence to represent a logic 0. The output is bandpass filtered and passed to the channel.

The receiver is considerably more complex, as might be expected from discussions in the previous chapters, and is shown schematically in figure 4-17. The first section of the diagram, the Heterodyner, is represented more fully in figures 4-18 and 4-19. The module consists of direct conversion of the DSSS signal to baseband, after bandpass filtering, figure 4-18. This is followed by I and Q phase digital matched filtering as described in chapters 2 and 3. The matched filter outputs are combined using the squaring algorithm and the modulus taken. A further block is added to calculate the phase of the local oscillator relative to the received data stream, as described in section 4.3.2. For clarity the chip clock is not shown in figure 4-19, although this would be required at each block shown.

The oscillator section could consist of a voltage tunable chip rate oscillator from which the RF could be synthesised. Alternatively a voltage tunable RF oscillator could be used in the RF section of figure 4-18, with a frequency divider network providing the appropriate division ratio to the chip clock rate.

Code synchronization detection uses the modulus output from figure 4-18, and the technique to be considered will be 4 single threshold crossing detectors with binary integration, shown in figure 4-15 and discussed in section 4.3.1.3.

RF recovery is shown in figure 4-17 as receiving the RF phase estimate from the baseband section of figure 4-19. This will allow discussion of RF recovery directly from baseband information by the technique discussed in section 4.3.2. Alternatively RF recovery may be achieved directly from the RF signal by the squaring or Costas loops of section 4.2.3, implying use of the frequency divider approach to chip clock generation. For clarity only the prior technique is shown, however the alternative technique will be discussed.

Data demodulation simply consists of sampling the output from the appropriate DMF at the code epoch, as defined by the code synchronization result. The term 'appropriate DMF' is used since there is an implication between the correct DMF to sample and the absolute phase of the local oscillator relative to the received data stream. The ideal situation is to maximize the snr into one DMF only, that is to maintain a relative carrier phase of 0 degrees. This then allows optimum sampling of the received chip stream within this DMF, as discussed in chapter 2, thus allowing the minimum achievable symbol error rate to be achieved at the DMF output for a given input snr. The output from the other DMF is used purely in RF recovery, which in turn leads to code tracking by the integer ratio of code and RF frequencies.

The final block shown in figure 4-17 is Tracking and Control. Little has been said about this important function in this work, since the emphasis has been on the acquisition of code synchronization. In practice this block would be required to perform many functions within the DSSS receiver. Although details of these functions will not be presented attention will be drawn to them where applicable.

Initially the receiver would not be synchronized to the incoming data stream, and the first task to be performed would be code synchronization acquisition. Therefore the code synchronization block would be enabled and this would identify the correct phase of the code epoch. This information could then be used to enable the RF recovery section, permitting optimum sampling of the DMF outputs for the formulation of the RF phase measurement. RF lock would then be achieved and the relative carrier phase adjusted to 0 degrees by the track and control unit.

At this point data demodulation could be enabled, since optimum detection would almost have been achieved.

Optimum detection would not actually be achieved until allowance has been made for the risetime of the lowpass filters, figure 4-18, as discussed in chapter 3.

This point is inevitably overlooked in discussions of digital communication systems, where reliance is placed on the ability to optimally sample the chip according to some embedded synchronization data within the received message, or an adequately high received snr. In the DSSS system under discussion close to optimum detection can be achieved by allowing the receiver to synchronize as described, and making correlation modulus measurements for advanced and retarded code phases, whilst maintaining the RF tuning voltage. This would be performed by the track and control unit. The time period required to perform such 'fine tuning' of the chip sampling instant would not be sufficient for RF lock to be lost. Once the optimum instant has been found data demodulation would be enabled, providing the best output snr for the available input snr.

The complexity of the Tracking and Control functions, and the philosophy by which they would be applied, suggests that a microprocessor approach would be suitable. It would undoubtedly be possible to research this area in it's own right, and this would be a valuable exercise. However it was felt that this would be beyond the scope of this thesis, where the emphasis has clearly been on the performance of the DSSS subsystems during the synchronization acquisition phase. Therefore the scope of the DSSS system study will be largely restricted to this area with some discussion on any implications relevant to tracking which may arise.

4.4.1 CONDITIONS FOR CODE SYNCHRONIZATION

From the discussions of section 4.3.1.3 we know that code synchronization will be achieved, on average, in n code periods for an input snr of E_c/N_o . The value of n depends on the value of E_c/N_o , and the number of binary integrators used on the algorithm output. For this study n will be taken as 4, and this does not impose a serious hardware increase over the $n=1$ case since the actual hardware implementations are very simple. Further since the square root algorithm is used to combine the modified DMF outputs we assume that carrier phase variations will not cause the amplitude of the sampled chip to vary and hence will not degrade the synchronization process, provided the variation does not increase too rapidly. A rapid variation will lead to uncertainty in the optimum chip sampling point, or at worse a change in the phase of the local carrier relative to the carrier of the received chips within one code sampling period. This can be seen from the following example. In n code periods there are $m.n$ chips, and we allow a maximum sampling slip of $1/s$ chips. Therefore

$$(mn)/f_c < (mn+1/s)/(f_c+f_d) \quad (4-10)$$

giving

$$f_d/f_c < 1/mns \quad (4-11)$$

where f_c is the transmitted chip frequency, f_c+f_d is the receiver chip sampling frequency, and m is the number of chips in the code. Expression (4-11) is in the form of a frequency stability, as would be obtained from a manufacturer's data sheets for a typical oscillator. For example with a code length of 511 chips, a maximum allowable slip of 1/8th of a chip, and a search of 50 code

periods an oscillator with basic stability of 5ppm would be required. Under these conditions the results of table 4-3 suggest that the input snr could be as low as -20 dBs.

We may now consider a practical oscillator, with a slightly reduced performance in terms of stability. The device considered [VECTRON 1985] is a voltage tunable 10 MHz oscillator, with temperature stability of +10 to -10 ppm over a temperature range of 0 to 50 degrees C, and a tuning range of +30 to -30 ppm. If we assume that at the start of the synchronization procedure there is a maximum error in frequency at the receiver of 20 ppm relative to the transmitter, due to temperature variation and tuning offset, and again assuming a maximum slip of 1/8 chips, we obtain the result $n < 12$ from table 4.3. Clearly this is not, on average, achievable by a sliding correlator synchronizer under any input snr conditions for the code length assumed. It is possible, however, to satisfy this condition using the one of four binary integrators scheme with an input snr greater than -15 dBs. Further simulation has showed that if the slip is increased to 1/4 of a chip, giving $n < 24$, synchronization may be achieved with sixteen binary integrators at an input snr of -20 dBs.

This highlights a major benefit of synchronization schemes which reduce the number of code periods required to obtain synchronization, and is a major advantage gained from the use of DMF detection in the DSSS receiver.

In general other factors may influence the frequency offset between transmitter and receiver. One candidate for this is Doppler shift introduced by the relative movement of transmitter and receiver. As a further example the potential use of DSSS systems for Land Mobile Radio (LMR) schemes at UHF is considered. Some interest has been shown in the potential of DSSS systems at these frequencies, since the inherent frequency diversity of the spread spectrum signal provides a degree of rejection against multipath distortion [YAMADA et al 1984]. In the LMR

environment Doppler shift arises due to movement of the mobile relative to either the base station, or other mobiles. Assuming a maximum vehicle speed of 35m/s, a maximum relative velocity of 70m/s is achievable between mobiles, and this figure is taken as the worse case.

The Doppler shift is given by

$$f_D = f_U / (1+v/c) \quad (4-12)$$

where v is the relative velocity between transmitter and receiver, f_U is the UHF carrier frequency, f_D is taken as the received centre frequency with Doppler shift (assumed to be below f_U and hence assumes transmitter and receiver are moving apart), and c is the speed of light. Assuming a carrier frequency of 2 GHz we obtain a Doppler shift of -467 Hz at the UHF frequency, or +467 Hz if transmitter and receiver are moving towards each other. Therefore in 1 second we receive $2 \times 10^9 + 467$ carrier cycles, assuming positive Doppler shift for clarity but without loss of generality, which corresponds to $(2 \times 10^9 + 467) / 200$ chips at baseband, assuming a chip rate of 10 MHz. Therefore the effective shift in sampling frequency at baseband is 467/200 Hz, less than 2.5 Hz.

This figure is considerably lower than the worse case offset obtained from instability of the oscillator, which was 10ppm, or 100Hz. This implies that the effect of Doppler shift on the spread spectrum signal at this carrier frequency is negligible when compared to the basic stability of the local oscillator. For example if the local RF carrier is synthesised from the basic oscillator and we assume a stability of 20 ppm there will be a maximum frequency offset at RF of 40 kHz. This is approximately two orders of magnitude greater than the expected Doppler shift, and it has already been shown that synchronization can be achieved given an adequate input snr at this frequency offset.

4.4.2 ABILITY TO ACHIEVE RF SYNCHRONIZATION

Unlike the relative sampling difference frequency the RF frequency offset is converted directly to baseband, giving a beat frequency of 40 kHz at the outputs of the lowpass filters of the complex baseband receiver. This results in a varying phase offset between the local oscillator and the baseband data, which must be corrected by the RF recovery network. This might also be termed RF tracking, since the presence of the integer ratio between RF carrier frequency and chip clock frequency leads to simultaneous achievement of coherent RF recovery and code tracking. In one code period, corresponding to a time of $511/(10 \times 10^6 \text{ Hz}) = 51.1 \text{ uS}$, the RF will vary by $(40 \times 10^3 \text{ Hz}) \times (51.1 \text{ uS}) = 2.044 \text{ cycles}$, or 12.8 radians per second. Therefore the carrier phase variation at baseband is far too great for recovery in a closed loop, since the loop bandwidth, derived from the basic code period, is too small. Under the worse case frequency offset the relative carrier phase will have passed through two complete cycles before the next phase estimate is taken at the DMF outputs.

Two possible approaches to solving this problem will now be considered. Firstly, RF recovery may be possible with the squaring or Costas loop. The stability of either loop will depend upon the input snr, and the bandwidth of the loop filter. For simplicity the squaring loop of figure 4-7a will be considered. Since frequency deviations from the nominal carrier frequency of up to 40 kHz in either direction are possible, the deviation bandwidth is 80 kHz. However after the squaring operation the signal at twice the carrier frequency will have a deviation bandwidth of 160 kHz. Therefore the squarer is followed by a bandpass filter, centred on the nominal carrier frequency, and with bandwidth 160 kHz. After down conversion, achieved by mixing with the local oscillator operating at twice the

input carrier frequency, the signal is lowpass filtered to a bandwidth of 80 kHz. This determines the range of frequencies over which the loop will acquire lock, and also defines the noise bandwidth into the VCO. At this point there is an improvement in snr between the input snr and the snr at the VCO input due to the reduction in noise bandwidth.

The improvement is approximately equal to the ratio of noise bandwidths pre and post filtering, suggesting an improvement of approximately

$$(10 \times 10^6 \text{ Hz}) / (80 \times 10^3 \text{ Hz}) = 125$$

or almost 21 dBs.

However since squaring is a non-linear operation there is interaction between the signal and noise spectral components and a 'squaring loss' is incurred, [ZIEMER and PETERSON 1985]. This represents the factor by which the phase error variance is increased over that of a conventional phase locked loop structure, and a loss of about 4 dBs can be expected at an input snr of -15 dBs, and about 2 dBs at an input snr of -20 dBs. The result is that the loop will have a high phase error variance for very low input snrs. Therefore the equivalent VCO input cnr will be about 2 dBs for an input snr of -15 dBs, and about 1 dB for an input snr of -20 dBs [ZIEMER and PETERSON 1985]. Similar results have been shown for the Costas loop [SIMON and LINDSEY 1977]. Under these input snr conditions the phase error variances will be approximately 94 and 160 radians². Thus for the two cases considered the phase error standard deviations will be approximately 9.5 and 12.7 radians [ZIEMER and PETERSON 1985]. During one code period these values are 4.9×10^{-4} and 6.5×10^{-4} radians respectively.

It may be possible, if necessary, to correct the phase of the VCO with information obtained from the DMFs. However, in general, the loop performance at these input

snrs would be acceptable for tracking purposes [ZIEMER and PETERSON 1985], although information regarding the loop's ability to acquire lock under these conditions was not presented in the above references.

For low input snrs and carrier frequency offsets, which are acceptable for synchronization, it would appear that satisfactory RF tracking is possible with the squaring or Costas loops, but not with the information available at the output of the DMFs alone. This is due to the bandlimiting effect of sampling the DMF outputs at the code rate.

This analysis highlights a fundamental difference between the performance of the squaring or Costas RF recovery loop, and the DMF output recovery loop. In the former the loop performance is limited by the snr, resulting in an expected variance in the resulting carrier phase estimate. In the latter the noise is reduced by the processing gain available from each DMF, and the system performance, when compared to squaring and Costas loops under identical input snr conditions, is limited by the rate at which the DMF outputs are updated compared to the maximum expected frequency deviation. Thus, although the DMF loop performance is ultimately limited by noise, a more fundamental limitation is the loop bandwidth.

A problem arises with squaring and Costas loops if the system has multiple users in an overlay or CDMA mode. Here the RF recovery loop may lock onto the strongest carrier, since no knowledge of the desired code is assumed by the RF recovery network. If each signal has a slightly different centre frequency, due to Doppler shifts or local oscillator instability at the transmitters, RF recovery of the correct carrier may not be possible. Therefore it is essential for the information available at the output of the DMF to be used, since once code synchronization has been accomplished the receiver has the benefit of the available processing gain.

This leads to the second technique for RF recovery, which is not based on the squaring or Costas loop structures, but on the output from the DMFs. If RF tracking is to be controlled by the carrier phase estimate obtained from the DMF outputs it is necessary to consider the maximum rate at which the carrier phase may change, such that a satisfactory control system can be designed.

As a simple approximation to the problem, the DMF outputs are considered as sampled and digitized versions of a phase comparator from which the absolute phase can be determined.

The sampling rate at the output of the phase detector is equal to the symbol rate, or

$$\begin{aligned} \text{Phase sampling rate} &= 1/mt_c && (4-13) \\ &= f_c/m \end{aligned}$$

where $f_c = 10$ MHz and $m = 511$ for the system under consideration giving

$$\text{Phase sampling rate} = 19.57 \text{ kHz}$$

Therefore the maximum frequency which can be reproduced at this sampling rate is $1/2mt_c$, as dictated by the sampling theorem, or approximately 9.78 kHz.

Thus it is possible, in principle, to design a phase-locked loop RF recovery system capable of locking to the received carrier provided the initial local oscillator frequency estimate is within about 9 kHz of the received signal carrier. For the system under discussion this would amount to a frequency stability of 9kHz/2GHz or 4.5ppm,

achievable in practice but possibly an expensive solution.

This approach is rather optimistic and does not consider the effects of noise at the DMF outputs. In terms of the average DMF output the above argument may be acceptable, however in practical terms the distribution of values about the mean DMF output will lead to a noisy estimate of carrier phase. This will be particularly evident as the snr is reduced. To combat this and produce an output more closely corresponding to the mean DMF outputs for the I and Q channels a number of successive outputs can be applied to a filtering algorithm. In it's simplest form this could be an averager, forming the sum of the successive outputs.

The averaging algorithm would lead to an improvement in the ratio of mean DMF output to standard deviation by a factor of $n^{1/2}$, n being the number of successive samples summed. In decibel notation this would equate to an 'integration gain' of $10\log(n)$ dBs, an equivalent expression to that used to define the processing gain of the DMF. In spectral terms the averager acts as a lowpass filter with a $\sin(x)/x$ frequency response, and is equivalent to an n tap finite impulse response filter with equal tap coefficients of $1/n$. The first transmission zero for this filter will occur at a frequency of f_c/mn , or f_s/n , and this may be defined to be the passband of the simple filter, with f_s as the symbol sampling rate at the DMF output.

Thus an improvement in output snr is obtained by filtering the DMF output but this is at the expense of loop bandwidth, which is reduced by a factor of at least $1/n$.

4.4.3 SUMMARY OF PROPERTIES OF PROPOSED RECEIVER

The proposed receiver utilises 4 single threshold crossing detectors with 4 bit binary integration to achieve an excellent average code synchronization period, clearly outperforming the sliding correlator approach.

RF recovery at the receiver can be by an analogue recovery loop, such as the squaring or Costas loops, and this will be achieved at approximately the same input snr as that required to achieve code synchronization. However this form of RF recovery will only work in an overlay mode between base and mobiles, assuming all overlaid signals transmitted from the base are sent at the same carrier frequency. Failure to adhere to this condition could lead to the receiver locking onto the strongest carrier received, which may be the incorrect signal.

Alternatively RF recovery may be achieved by utilising the phase information which can be calculated from the DMF outputs once code synchronization has been achieved. This technique is only applicable to very low frequency differences between the transmitter and receiver oscillators, since the RF recovery loop bandwidth is restricted. However this technique will work in an overlay mode even if all received signals have slightly different centre frequencies since the processing gain of the DMFs can be used to reject unwanted signals in favour of the required signal.

A further benefit of applying the squaring algorithm at the receiver is the provision of information of the received signal power level. This can be used to provide power control, helping to avoid the so called 'near-far' problem associated with CDMA schemes.

Thus if a Land Mobile Radio system consisting of a

Base and several surrounding Mobiles is considered it is clear that all signals transmitted from the Base can be sent on the same carrier and with the same power, allowing the analogue recovery loop to be used at the Mobile receivers. Each Mobile will receive all signals on a unique carrier frequency, due to the Doppler shift introduced by their own movement. The RF recovery loop will lock onto this carrier, and will not be prone to false lock since all signals will still be centred on the same carrier. If this same frequency is used at the Mobile transmitter then, assuming reciprocity across the transmit/receive path, the Base receivers will each receive unwanted signals offset by an unknown shift, and the wanted signal offset in frequency by twice the Doppler frequency experienced at the Mobiles. Since this offset will be a small percentage of the transmitted frequency, as shown, RF recovery can be achieved at the Base receivers by utilising the information obtained from the receiver DMFs, which will reject the unwanted signals. Furthermore each Mobile receiver can make an estimate of the Base signal received power from the algorithm output as described in chapter 3.

The result is a feasible approach to operating a DSSS system in an overlay mode within a Land Mobile Radio environment. Such schemes have been proposed and analysed by many authors, but the important questions of code and RF synchronization inevitably overlooked in favour of usage, regulation control, and number of users capable of occupying the channel simultaneously.

These arguments are doubtless of prime importance when planning such a system. Opponents to the DSSS approach have frequently hooked on the complexity aspect of DSSS receivers in proposing their criticisms. The study of the DSSS receiver presented in chapters 2, 3 and 4 of this work support this argument, although the techniques described would be readily applicable to integration using VLSI techniques. The most obvious barrier to this goal is the high cost of initiating such technology, which could only

be surmounted by the economies of scale obtained with volume sales.

4.5 CONCLUSIONS

This chapter extended the analysis of the DSSS receiver to cover the important areas of receiver synchronization, in the sense of code synchronization acquisition, tracking, and RF recovery.

For code synchronization established techniques were reviewed, and compared to simple digital implementations. In terms of correlation power, it was shown that a single DMF of length m chips can perform the same task as an array of m binary integrators. Each scheme can achieve code synchronization in a minimum of 1 code period in the absence of noise. However the DMF is a more efficient integrated hardware solution.

The sliding correlator was reviewed, this being the minimum hardware solution at the expense of acquisition time. Other estimation techniques, RASE and RARASE, were also outlined briefly for completeness, although there is no direct comparison between these techniques and techniques employing the DMF. These techniques have been found useful when composite codes are used for signalling, a technique beyond the scope of this work.

Code tracking was reviewed, and the operation of the delay lock and Tau dither loops outlined. Each technique is applicable to a baseband implementation, but would require additional DMFs to form the references.

Two analogue RF recovery schemes were discussed, the squaring and Costas loops. These are widely covered in the literature and only brief attention was given to their relative performances. It was noted that in a DSSS system operated in an overlay mode such schemes could result in

false carrier recovery if received signals had slightly different centre frequencies, the loops tending to lock onto the strongest signal. This problem arises since the loops do not have the benefit of the processing gain of the DSSS DMF.

Code synchronization acquisition using digital techniques was examined, and it was shown that this must be initiated by a threshold detection scheme. The probabilities of threshold crossing were examined for in phase and out of phase cases, and it was found that the false alarm rate would be high if each out of phase case could not be discriminated from the collection of out of phase cases. Two techniques for achieving this were considered, the most hardware efficient being binary integration of a number of threshold crossings. Tabulated theoretical results were presented which showed that the ratio of in phase to out of phase threshold crossing probability is dramatically increased by this scheme when compared to the case of a single sample, at the expense of response time.

The analysis was extended to considering the case of a single threshold detector in place of the m detectors required for a code of length m . It was shown that the search philosophy changed from selecting one of m outputs and assuming that synchronization would be achieved within a single search phase, to one of assuming that a search is initiated by a single threshold crossing, and that synchronization is assumed correct if the binary integrator output exceeds the preset synchronization level at the end of the search. For the latter scheme the time taken to achieve synchronization is the most important parameter since a high false alarm rate is possible. Values of mean search periods were presented for several input error probabilities, and these indicated that the out of phase threshold crossing probability must be kept extremely low if false synchronization is to be avoided. This feature was attributed to the large number of out of phase positions

within a code period and the resulting likelihood of a search initiation at a false synchronization position.

It was shown that this problem could be reduced considerably by operating a number of detectors independently, each searching a subset of the code phases. This reduced the number of out of phase locations included in the search and tabulated results indicated the enormous benefit of this simple enhancement, particularly at low input snrs. During synchronization this scheme utilised the processing gain of the DMFs, and an example was presented in which a loss in snr of only 3.6 dBs was incurred relative to the coherent DMF.

RF recovery was considered and it was shown that the outputs from the DMFs could be used to formulate a measurement of the relative carrier phase, although a restriction on the rate of phase change exists due to the bandwidth of the data at the DMF outputs.

A DMF receiver was proposed and analysed, utilising the new techniques of code synchronization acquisition and RF recovery described. It was shown that such a scheme could achieve code synchronization and RF recovery in a land mobile radio environment given a suitable snr at the receiver input.

Code synchronization could be achieved in an average time period considerably lower than would be achieved by a sliding correlator system, allowing the receiver to be used in a burst transmission mode. In such a scheme Doppler effects and local oscillator stability play important roles.

Local oscillator instability at the receiver leads to the requirement for an analogue RF recovery scheme, which in turn implies that the Base must transmit all signals on a single centre frequency. At the Mobiles the signals are received on the same carrier frequency and can be separated

by using the processing gain available within the DSSS receiver. The movement of each Mobile relative to the base results in the carrier being shifted slightly due to the Doppler effect, and this may be different for each Mobile. The Mobile transmits back to the Base at the received carrier frequency, which is the Base transmitted carrier frequency shifted by the unknown Doppler frequency. This is in turn received by the Base, shifted by twice the unknown Doppler frequency. Thus the Base only has to separate the received signals using phase information obtained from the appropriate receiver DMF outputs, since it was shown that the Doppler shifts are small for the example analysed.

The system described included most of the advantageous features analysed in this chapter. Undoubtedly further enhancements could be made, particularly in the areas of channel equalization, utilizing information obtained from the DMFs.

Whilst DSSS systems are not currently used in land mobile radio applications they are currently employed in satellite navigation systems. The techniques of rapid code synchronization are equally applicable, and could find widespread use in satellite navigation receivers if the cost of the DMF can be brought down to a competitive level. Further as more bandwidth in the microwave region is released for LMR applications it may ultimately become essential to use broadband techniques such as DSSS, at which point it is highly likely that the DMF will find widespread use.

REFERENCES

- ASH C.P. (1984) 'Digital implementation of a narrowband satellite receiver', IEE Proceedings, Volume 131, Part F, Number 6, October 1984, pp 570 - 576.
- BEALE M. and TOZER T.C. (1979) 'A class of Composite sequences for spread-spectrum communications', IEE Journal on Computers and Digital Techniques, Volume 2 Number 2, April 1979, pp 87 - 92.
- COOPER H.W. (1974) 'Why Complicate Frequency Synthesis?', Electronic Design, Volume 22, Number 15, July 1974, pp 80 - 84.
- COOPER R.C. and MCGILLEM C.D. (1986) 'Modern communications and spread spectrum', McGraw-Hill.
- GOLD R. (1967) 'Optimal Binary Sequences for Spread Spectrum Multiplexing', IEEE Transactions on Information Technology, Volume IT-13, pp 619 - 621.
- GOLD R. (1978) 'Carrier Phase Estimates Using I and Q Channel Digital Correlators', Interstate Electronics Corporation, IOM 3300-281.
- GOLOMB S.W. (1967) 'Shift register sequences', Holden-Day.
- HOLMES J.K. (1982) 'Coherent spread spectrum systems', John Wiley and Sons.
- ORMONDROYD R.F. and AL-RAWAS L.A. (1986) 'The use of composite sequences for rapid acquisition in spread-spectrum receivers', Proceedings of the Fourth International Conference on Radio Receivers and Associated Systems, July 1986.

- ORMONDROYD R.F. and COMLEY V.E. (1984) 'Limits on the search rate of a pseudonoise sliding correlator synchroniser due to self-noise and decorrelation', IEE Proc-F, 131, 1985, pp 742 - 750.
- OULD P.C. and VAN WECHEL R.J. (1981) 'All-digital GPS receiver mechanization', Presented to The Institute of Navigation's National Aerospace meeting, Pennsylvania, 1981.
- SIMON M.K. and LINDSEY C.L. (1977) 'Optimum Performance of Suppressed Carrier Receivers with Costas Loop tracking', IEEE Transactions on Communications Volume COM-25, February 1977.
- SIMON M.K., OMURA J.K., SCHOLTZ R.A. and LEVITT B.K. (1985) 'Spread Spectrum Communications', Volume 3, Computer Science Press.
- WARD R.B. and YIU K.P. (1977) 'Acquisition of pseudonoise signals by sequential estimation', IEEE Transactions on Communications Volume COM-23, August 1977, pp 784 - 794.
- VECTRON (1985) 'Hybrid DIP TTL VCXO', Vectron Product Bulletin, Vectron Laboratories Inc., 166 Glover Avenue, Norwalk, Connecticut, 06850, U.S.A.
- YAMADA K., DAIKOKU K., and USUI H. (1984) 'Performance of Portable Radio Telephone Using Spread Spectrum' IEEE Transactions on Communications, Volume COM-32, No. 7, July 1984.
- ZIEMER R.E. and PETERSON R.L. (1985) 'Digital communications and spread spectrum systems', Macmillan Publishing Company.

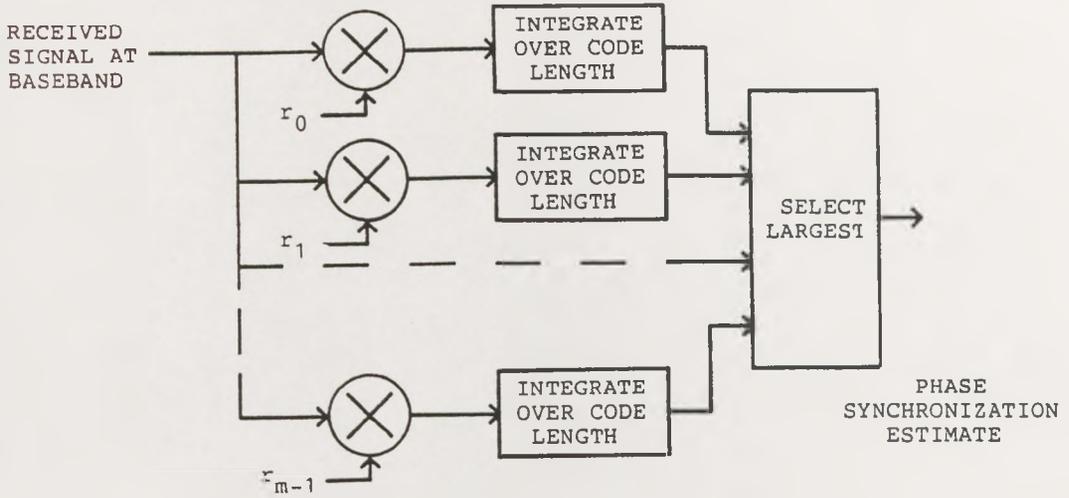


Figure 4-1. Maximum likelihood detector.

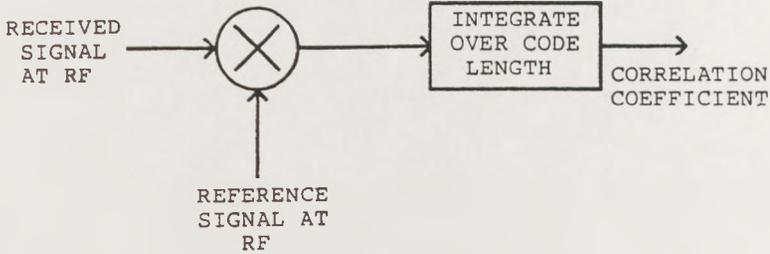


Figure 4-2. Active correlator.

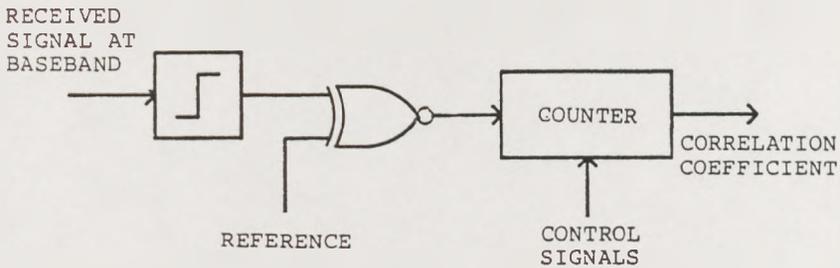


Figure 4-3. Digital correlator.

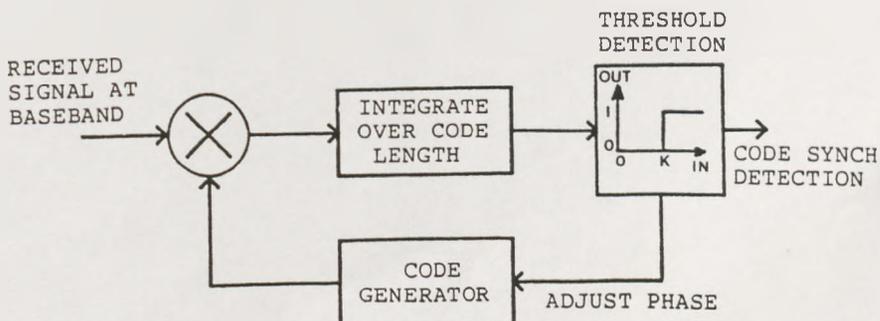


Figure 4-4. Stepping correlator.

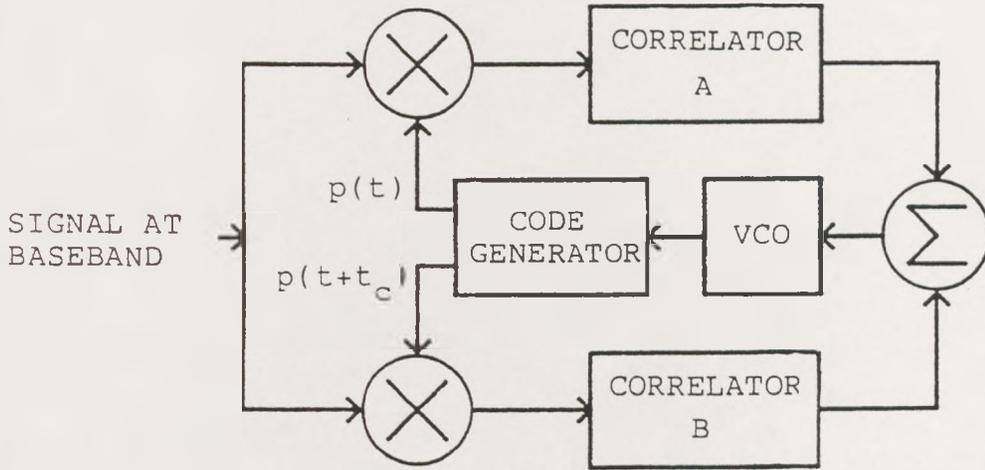


Figure 4-5a.

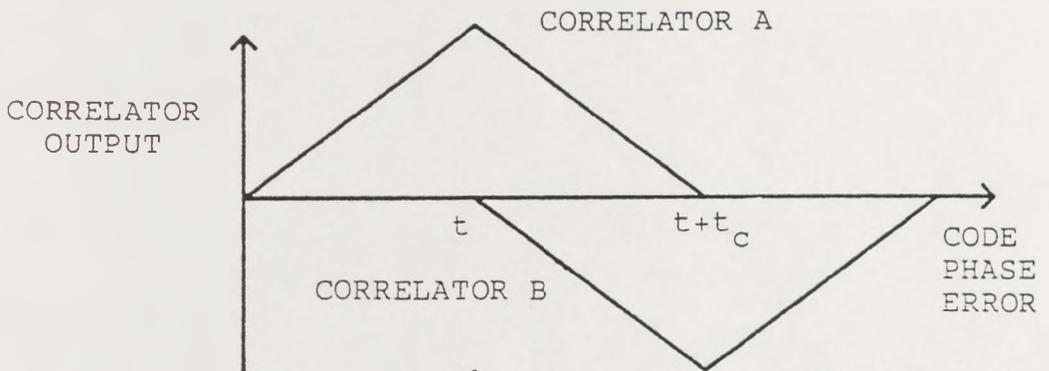


Figure 4-5b.

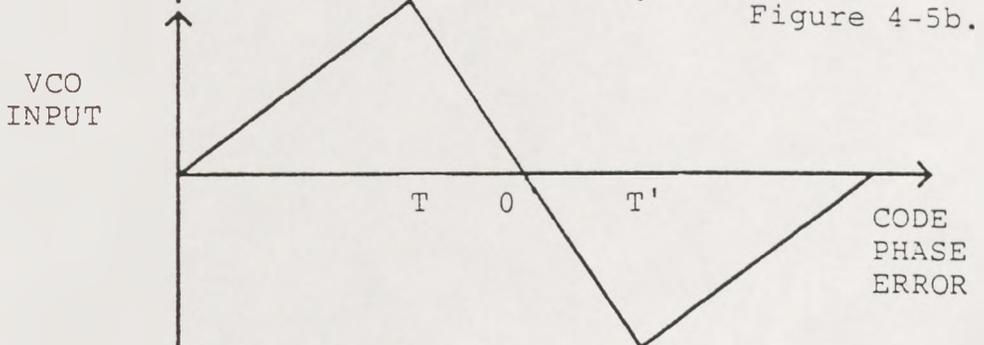


Figure 4-5a. Delay lock loop.
 Figure 4-5b. Integrator outputs.
 Figure 4-5c. Error signal.

Figure 4-5. Delay lock loop

Figure 4-6a.

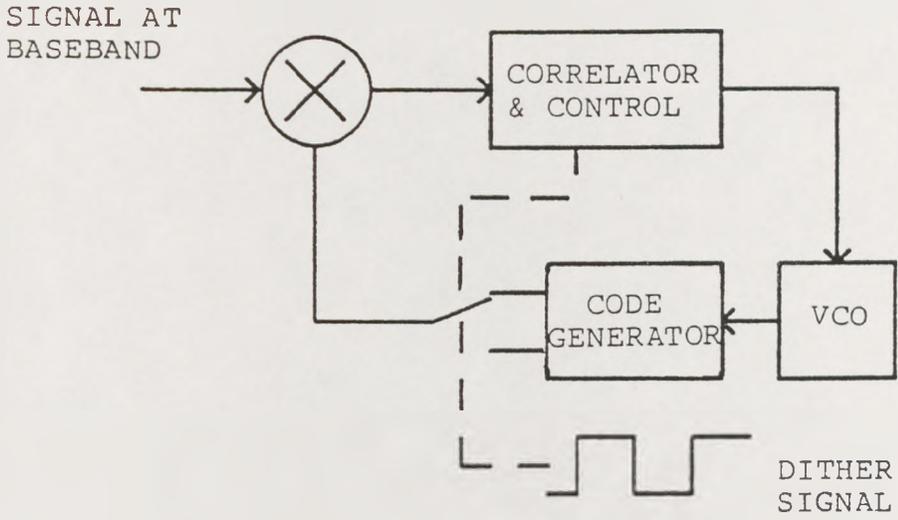


Figure 4-6b.

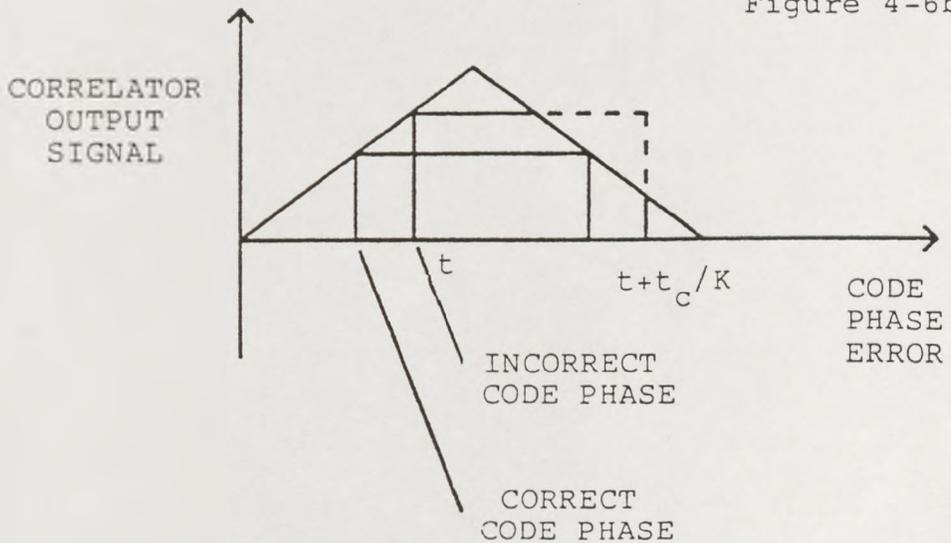


Figure 4-6a. Dither loop.
Figure 4-6b. Error signal.

Figure 4-6. Dither loop

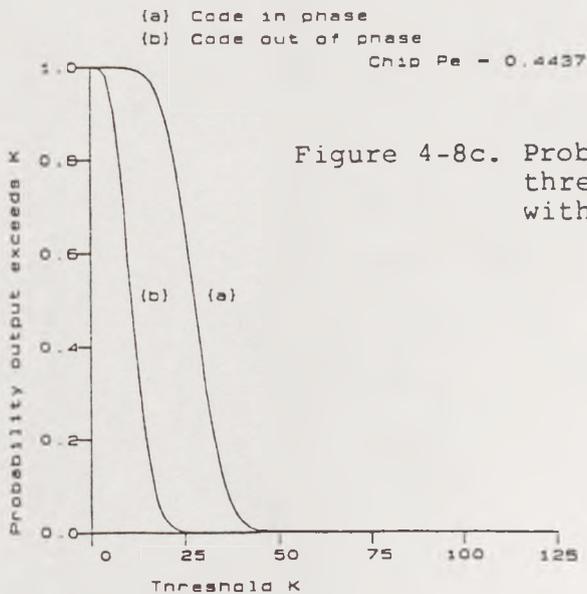
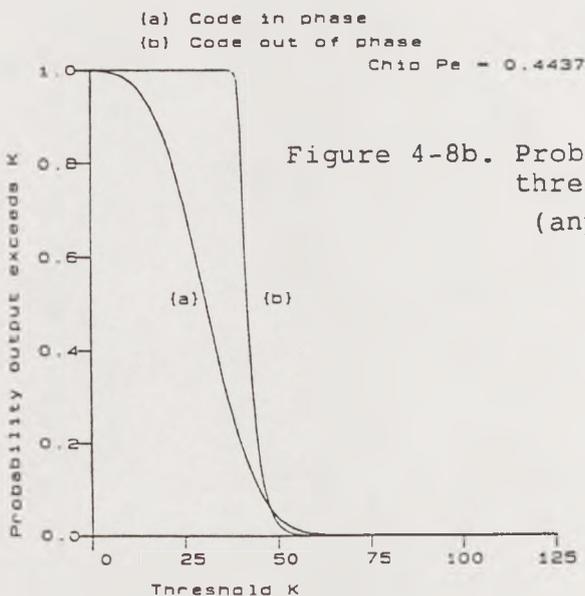
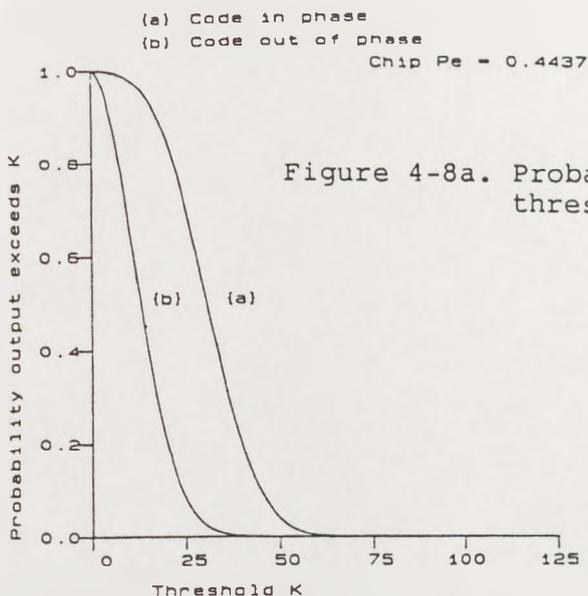


Figure 4-8. Algorithm output probabilities, $p_e = 0.4437$

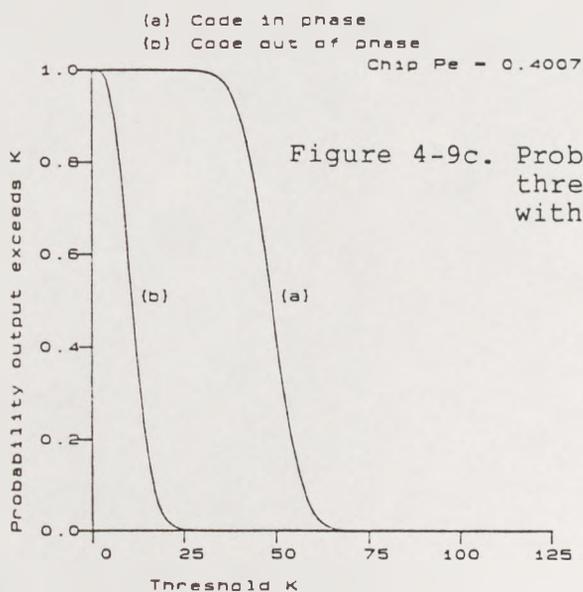
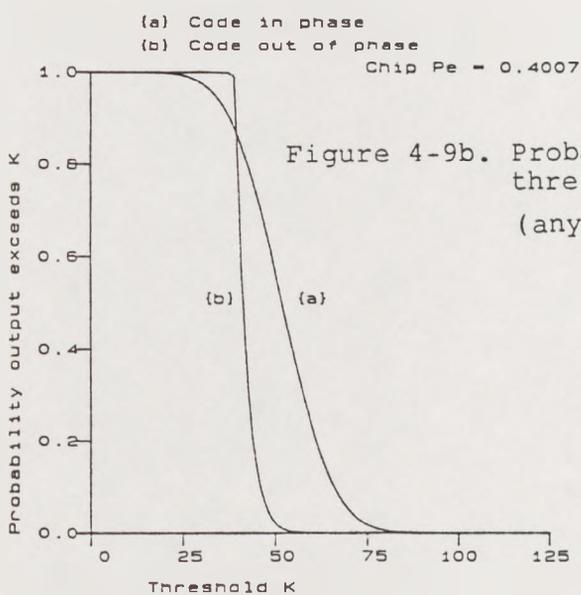
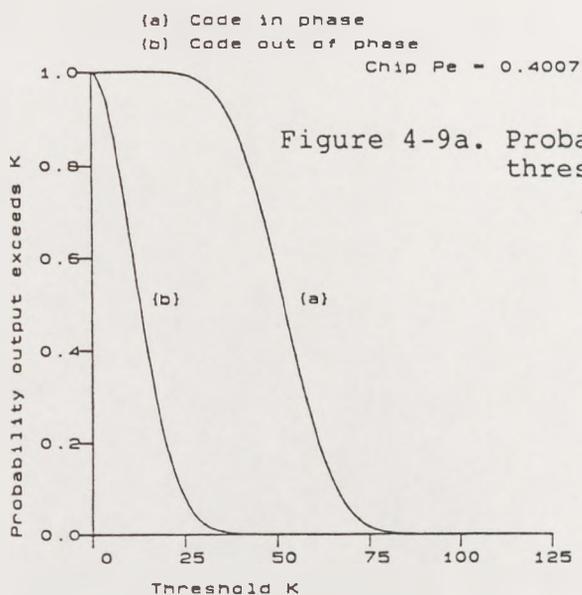


Figure 4-9. Algorithm output probabilities, $p_e = 0.4007$

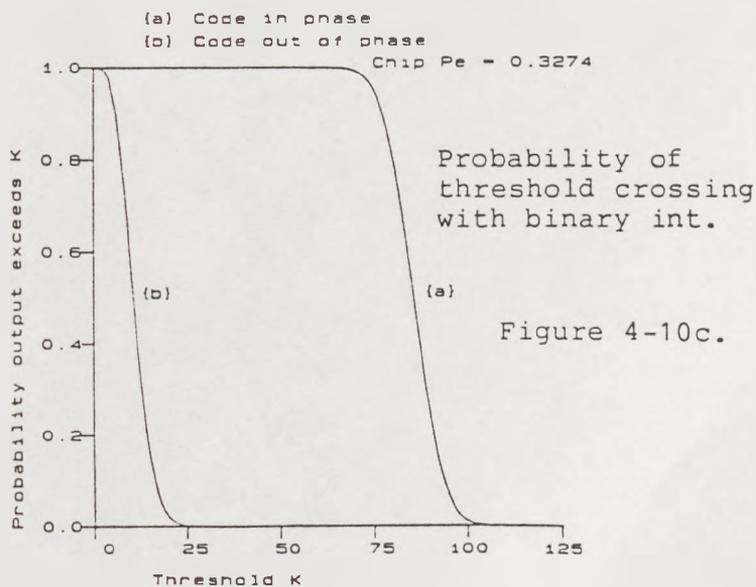
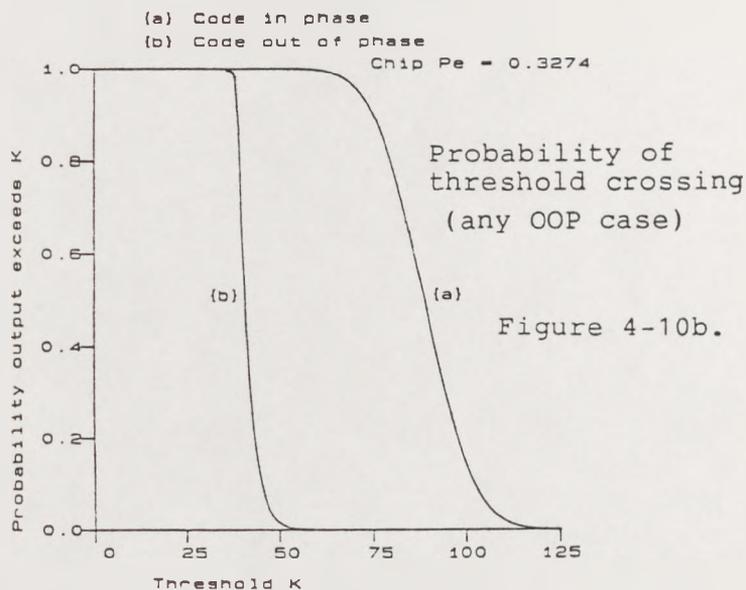
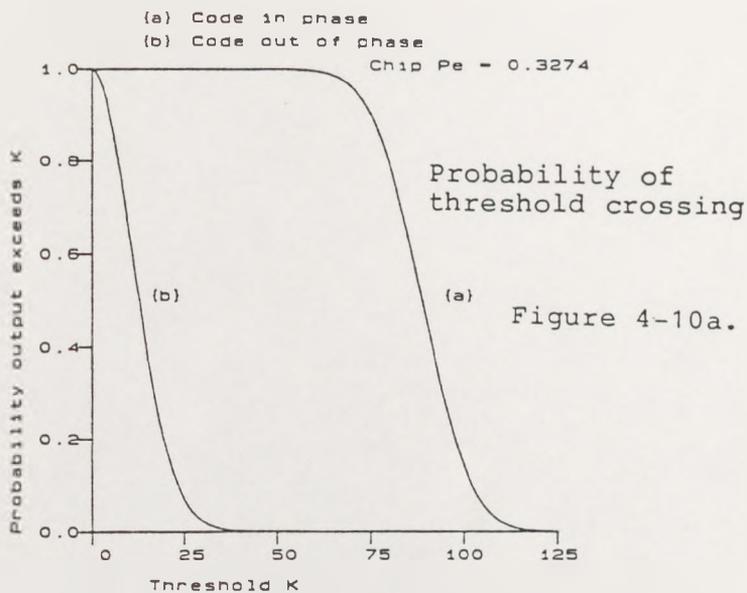


Figure 4-10 Algorithm output probabilities, $p_e = 0.3274$

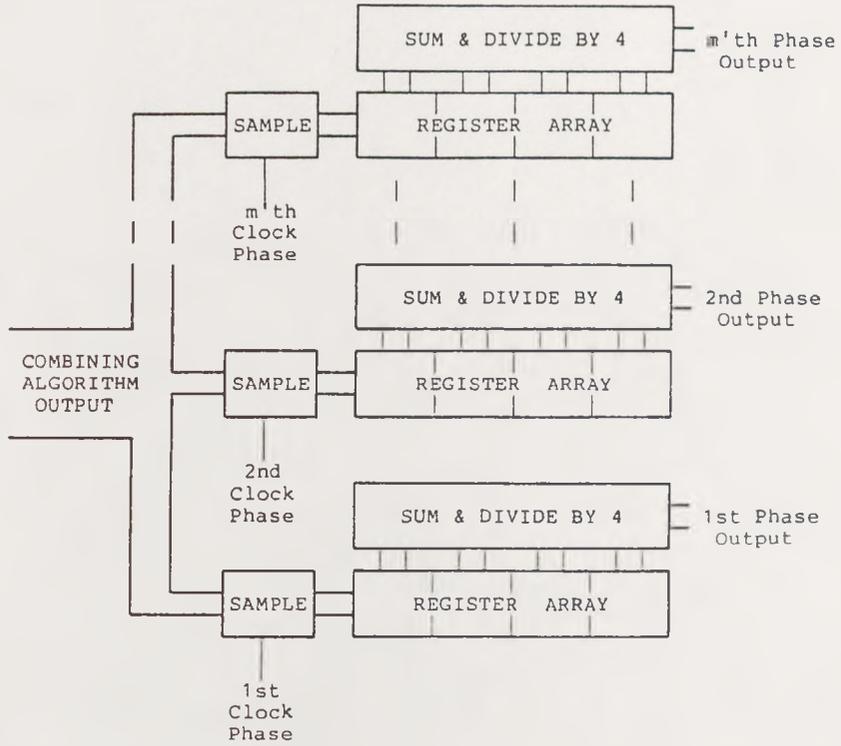


Figure 4-11. Production of Average Algorithm Output.

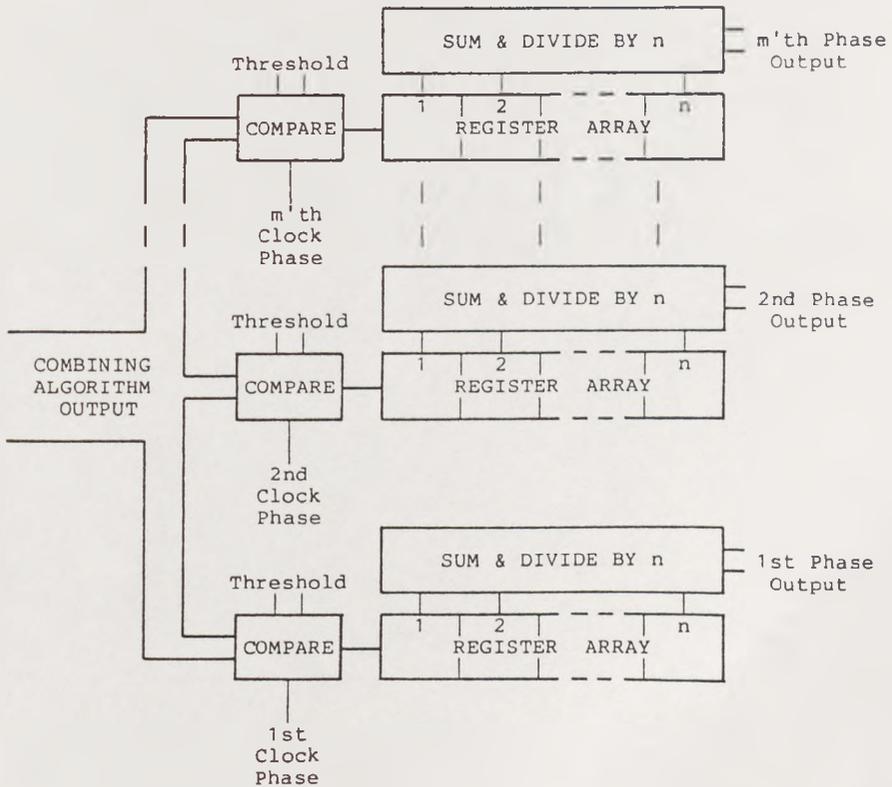


Figure 4-12. Production of Sum of Threshold Crossings.

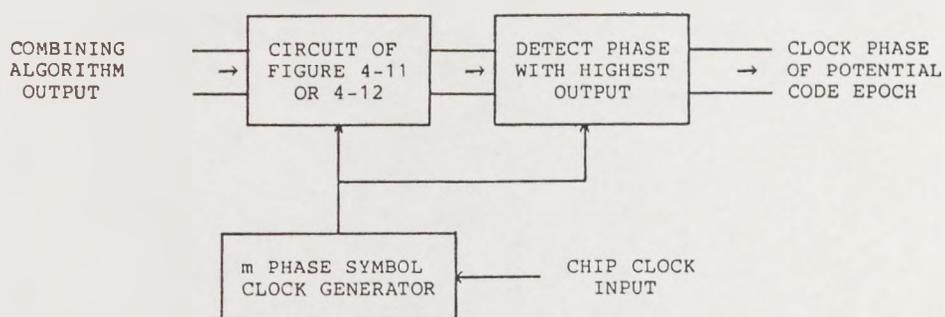


Figure 4-13. Code Epoch Detection.

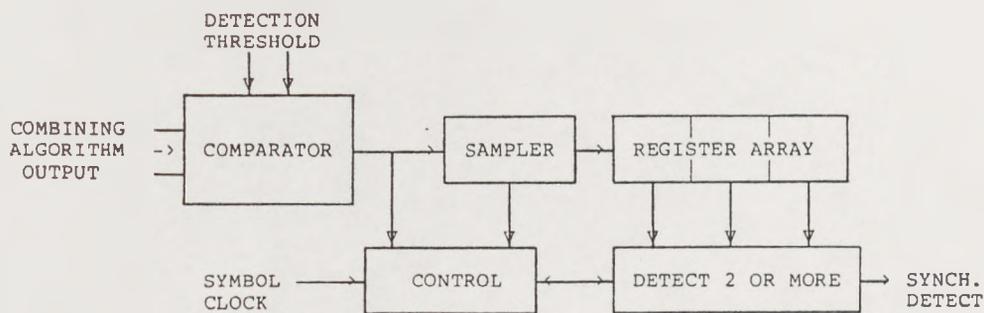


Figure 4-14. Code Synchronization Detection by Single Threshold Crossing with Binary Integration.

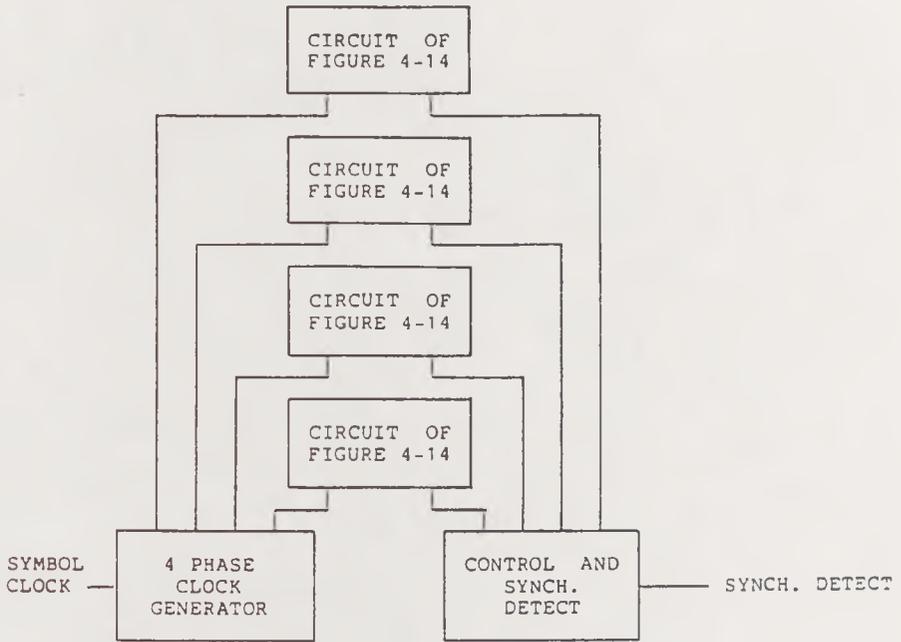


Figure 4-15. Code Synchronization Detection by 4 Single Threshold Crossing Detectors with Binary Integration.

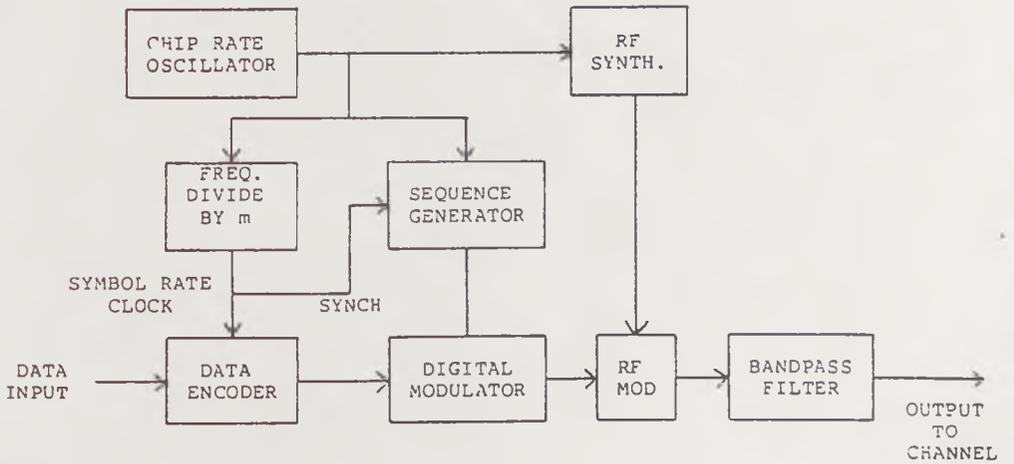


Figure 4-16. DSSS Transmitter block diagram

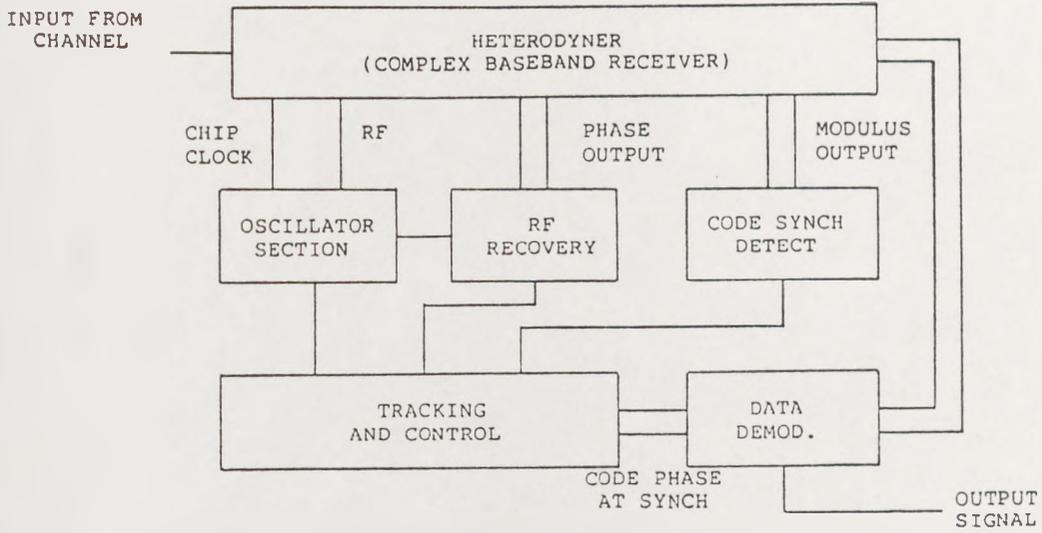


Figure 4-17. DSSS Receiver block diagram.

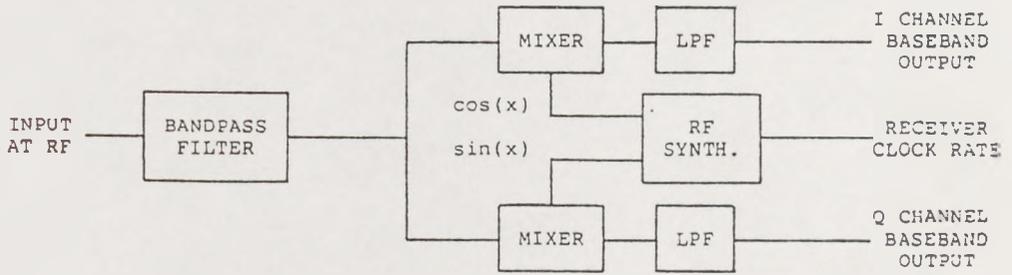


Figure 4-18. DSSS Receiver RF down conversion section.

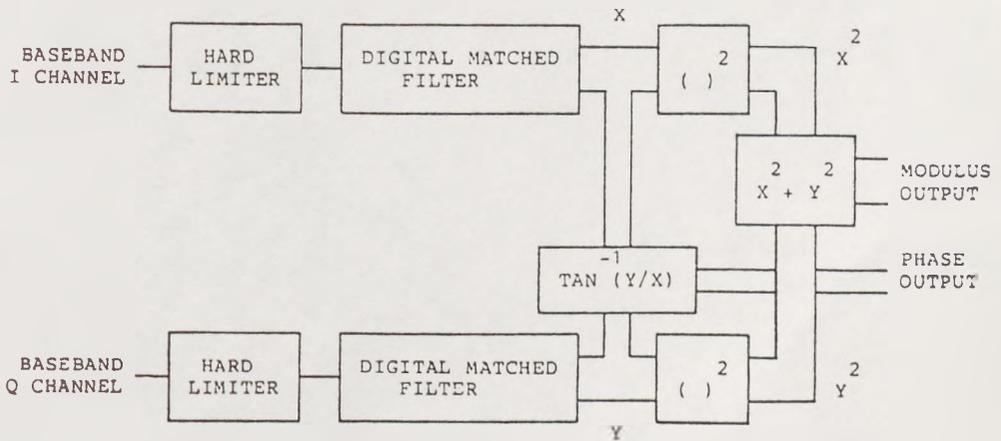


Figure 4-19. DSSS Receiver baseband section.

CHAPTER 5

PROTOTYPE DIRECT SEQUENCE SYSTEM.

5.1 INTRODUCTION

The importance of obtaining results from practical systems is often overlooked or ignored, particularly by those who favour computer simulation techniques. Although the power of these techniques cannot be ignored their advocates frequently describe practical experimentation as 'inflexible', 'inaccurate', or even 'cumbersome'. This somewhat narrow-minded view is exposed when attempts are made to simulate DSSS systems. The potentially large number of computations required in such a simulation, resulting from the large bandwidth expansion ratios, require considerable processing time. The use of a powerful and expensive mainframe computer is necessary to perform the simulation in a reasonable time period [COOPER & NETTLETON 1977].

Simulations of digital systems rarely produce results of chip error rates below 10^{-4} without the aid of predictive techniques, such as the Monte Carlo method [MIDDLETON 1965]. However a practical system operating with a chip rate of 10 MHz can produce an error rate measurement of 1 part in 10^7 in 1 second. Further, if a bandwidth expansion ratio of 511 is used, symbol error rates can be measured to the same precision in approximately 10 minutes. The processing of such an enormous amount of data by a simulation technique would require many hours of dedicated CPU time [SKAUG 1985].

Simulation rarely, if ever, exposes problems encountered in practical systems arising from component or sub-system imperfections. Practical measurements must always be made prior to system design and commissioning. Therefore simulation techniques may be seen as a useful aid

in verifying system specification, although it is unlikely that these techniques will ever totally replace practical experimentation in system evaluation.

This chapter describes the experimental system constructed and used to obtain practical results on the DSSS techniques described, and in particular the DMF and combining algorithm. The transmitter and receiver are treated as separate subsystems, although for the experimental procedures these were linked by coaxial cables for the transmission of signals at the system IF frequency of 70 MHz. The use of a UHF frequency was considered as a means of establishing the performance of the system in a land mobile radio environment. However it was felt that this approach would undoubtedly lead to the introduction of problems at RF, and detract from the main impetus of the research, namely the investigation of the performance of the DMF and synchronization schemes.

The hardware was constructed in a modular fashion to allow various configurations to be evaluated. In general each module occupied a separate card. These were incorporated within a rack system.

5.2 PROTOTYPE DIRECT SEQUENCE TRANSMITTER.

The transmitter block diagram is shown in figure 5-1. A carrier waveform, of nominally 70 MHz frequency, is phase reverse keyed by a pseudorandom binary sequence. This process generates the broad bandwidth signal. The input modulating signal may be either an audio signal or a digital signal generated at the symbol rate. If the former is chosen the audio signal is digitized by a delta modulator which is then used to modulate the broad bandwidth signal, either by on-off keying or sequence inversion keying of the RF carrier. If a digital signal is required to act as the modulating signal then this may be used directly in place of the delta modulator. The

following sections describe the operation of each module of the transmitter in greater detail.

5.2.1 70 MHz OSCILLATOR AND SYSTEM CLOCK GENERATOR.

The oscillator is of the impedance inverting Colpitts type and is shown in figure 5-2, [IQD 1983]. A fifth overtone 70 MHz crystal forms the frequency selective element of the circuit. The oscillator output is buffered by a transistor in common collector mode. This provides isolation for the oscillator from various load conditions and gives a low output impedance suitable for 50 ohm loads. This also has the advantage of giving a high output level of 2 dBm since the oscillator section drives a high input impedance. The buffered output is then split two ways by a Minicircuits power splitter, type PSC-2. One output from this splitter forms the I.F. signal for the mixer section. The other is divided in frequency and converted to TTL logic levels to form the clock for the logic circuits. The division ratio can be set to seven or fourteen giving clock frequencies of 5 or 10 MHz.

The divider circuit, figure 5-3, comprises three cascaded D type flip-flops and a three input NOR gate. The flip-flops are reset to zero by the NOR gate when the outputs are all at logic 1. This corresponds to the seventh count in the sequence and hence gives a division ratio of seven. This can be further divided by two to give an overall division ratio of fourteen. Due to the high clock rate of 70 MHz it is necessary to use a very high speed logic family, Emitter Coupled Logic (ECL). In general this logic family requires a dual rail power supply. However due to the simplicity of the circuit and the use of AC coupling into a biased gate at the counter clock input it is possible to use only a single rail supply.

The oscillator output is AC coupled into the first flip-flop which is biased to half the supply rail,

approximately 2.5V. The counter output is taken from the third flip-flop, this being the most significant bit (MSB). The mark space ratio is 3:4 due to the division ratio of seven. This output is then AC coupled into a TTL buffer to give the system clock. A TTL flip-flop is included which allows the output clock frequency to be switched from 10 to 5 MHz.

5.2.2 CODE GENERATORS.

Many codes are possible in DSSS applications and any system under development for research purposes should have the potential for the introduction of new codes. Those currently used are generated by feedback shift register networks. The feedback function is generally formed by exclusive-OR gates which perform a modulo-2 addition of their inputs. However for very complex feedback functions, requiring more than six inputs, propagation delays through the exclusive OR gates become a problem with the currently available 'LS' series of TTL logic gates operating at 10MHz. This can be overcome by the use of more expensive higher speed logic families such as FAST (Fairchild Advanced Schottky TTL). The solution adopted is the use of a parity generator/checker. The device chosen can produce a modulo-2 addition of up to nine inputs with a propagation delay approximately equal to two cascaded exclusive-OR gates. The use of this device (74F280) thus allows greater flexibility in the generation of codes at high clock rates.

The code generator is shown in figure 5-4. A nine stage shift register is used to generate repetitive codes with a maximum length of 511 bits. In the generation of pseudonoise sequences all registers are not allowed to be zero at the same time, as this condition is usually self perpetuating [GOLOMB 1967]. This accounts for the sequence length being odd. Two nine bit shift registers are available for the generation of m-sequences and their

outputs can be combined for the generation of Gold codes, [GOLD 1967], [PETERSON and WELDON 1972]. DIP switches (Sc, Sd of figure 5-4) are included in the register outputs to give a greater flexibility in the selection of feedback taps.

The code generator operates at a frequency of 10 MHz, available from the backplane signal CLK. This gives a code of period 51.1 μ S with chip period of 100nS. DIP switches (Sa, Sb of figure 5-4) are also included on the preset inputs to each shift register. This allows preset conditions to be entered into the registers at the beginning of a code. Such conditions, or offsets, in the code generator are required for the generation of the complete set of Gold codes. The use of 74LS96 shift registers introduces a problem since the register must be cleared before being preset, as it is not possible to preset a zero into the register. This is achieved by a delay in the delta modulator clock generator and synchronization circuit discussed in section 5.2.4. Four internal sequences, two maximal length and two Gold, are switch selectable with the provision for two further external sequences, EXT 1, EXT 2.

5.2.3 AUDIO ENCODER.

Whilst speech is perhaps not the most appropriate form of test signal for validating digital communication networks it is useful in the experimental phase to assist in establishing a level of performance in a system. In the case of the DSSS system the availability of recorded speech became a useful indicator to the immediate level of performance available from particular system arrangements. Thus a system of speech encoding was added to the DSSS transmitter. Many schemes are available for the digital encoding of speech [JAYANT 1974] but one of the simplest of these is Delta Modulation.

Figure 5-5a is a block diagram of a basic linear delta modulation scheme. At the encoder the magnitude of the input signal is compared with the integral of all the previously transmitted bits. If the magnitude is greater then a '1' is transmitted, else a '0' is transmitted. The transmitted bit is then included in the integral for the next comparison. The main advantage of such a scheme is that only one bit of information is sent. This avoids many of the problems of frame synchronization associated with schemes such as PCM. At the receiver demodulation is performed by a simple integration of the received bit stream, as in the comparison arm of the transmitter. By using an integrator the comparison arm of the modulator closely tracks the input waveform.

A refinement on the basic delta modulator is the Continuously Variable Slope Delta modulator (CVSD) [MOTOROLA 1976] as shown in figure 5-5b. The CVSD circuitry provides increased dynamic range and hence a reduction in the likelihood of slope overload by adjusting the gain of the integrator. External to the basic deltamodulator is a shift register which contains the previous few outputs from the delta modulator. Should the register contain all ones or all zeros then this 'coincidence' condition indicates that the gain of the integrator is too small and slope overload will occur. The gain of the integrator is then increased and in this way it is possible to track steeper slopes with lower average quantization noise. This is repeated at the receiver. Since this algorithm operates on the past serial data it changes the nature of the bit stream without changing the channel bit rate.

The device chosen for the encoder is the Motorola MC3417, and the circuit used is shown in figure 5-6. A 3-bit shift register is used in the algorithm of the MC 3417. This is optimized for general communications systems at low transmission rates. The analogue input is compared with the analogue feedback and the sign of this comparison results in a 1 or 0 output. This is then latched into the

first stage of the shift register on the next clock pulse and is also the input to the slope polarity switch. This controls the sign of the current entering the integrator allowing a rising or falling ramp output voltage from the integrator. If a run of three 1's or 0's is present in the shift register the 'coincidence' output goes low. This then also produces a run of 1's or 0's which is filtered by the syllabic filter and controls the magnitude of the current entering the integrator. The integrator output forms the analogue feedback to the input comparator. The audio input signal is derived from a portable cassette recorder and bandlimited by an external bandpass filter.

5.2.4 DELTA MODULATOR CLOCK AND SYNCHRONIZATION GENERATOR.

This circuit performs two functions in the transmitter. Firstly it divides the 10 MHz clock input by 511 to provide the 19.6 KHz symbol clock for the delta modulator. Secondly it generates reset and preset pulses for the shift registers of the code generator. This is necessary to align the start of the code with the start of the data output from the delta modulator. The circuit is shown in figure 5-7. Three synchronous 4-bit binary counters are used. These are 74LS191 counters, chosen because they have asynchronous load inputs. Three devices are configured to form a 12 bit counter, of which 10 bits are used (Q0-Q9). Since the count of 512 corresponds to Q9 = 1 (Q0 to Q8 = 0) this condition is detected and loads the counter with the number 1. Thus the counter covers the range 1 to 511. Between the count range of 256 and 511 output Q8 is a '1' and between the range 1 and 255 it is a '0'. This then forms the clock for the delta modulator. Output Q8 is then buffered with a 74LS14 inverter. The narrow pulse output of about 30nS from Q9 is inverted and is the 'clear' input for the shift registers of the code generator circuit. A delayed version of Q9 is used as the 'preset enable' inputs of the shift registers. This is required since the shift register used does not allow the

register to be preset with the condition '0'. Hence they must be cleared and then preset where required. The timing delays are calculated such that this can occur within one clock cycle of 100nS to avoid missing a clock pulse in the code generator.

5.2.5 70 MHz LIMITER AND MIXER MODULE.

The function of the mixers is to generate the BPSK signal by phase reversing the carrier at 70 MHz with the code. The data bit stream can then be used to either invert the signal sequence (SIK modulation) or to on-off key the signal at RF (OOK modulation). An initial design gave poor results which were not generally repeatable when the 70 MHz local oscillator source was changed from a transistor oscillator design to a commercially available high quality synthesizer. Several oscillators were tried and similar variations in results were found. This suggested that amplitude variations and pick-up from cables was a problem. A self-contained mixer module was then designed, figure 5-8. The advantage of this design is that all input signals are fixed in level within the module before entering the mixers. This leads to isolation of the mixers from stray pickup.

The input at 70 MHz is AC coupled into an Emitter Coupled Logic (ECL) line receiver, type MC10115. This has a large signal gain of about 5 and limits the signal to 1V pk-pk. The output of this device is then AC coupled into an integrated amplifier, type SL560. This has a gain of 5 also, and a natural roll-off above 70 MHz leading to a reduction in the level of harmonics generated above 70 MHz. Intermodulation products do not exist since only one input is present.

Initially a lowpass filter was included between the output of the amplifier and the first mixer. This reduced all harmonics to below 60 dB relative to the 70 MHz

carrier. However due to non-linearity within the mixer the harmonics reappeared at only 30 dB below the carrier. With the filter removed an increase in harmonic level of only 2 dB was observed. This led to the conclusion that the filter would be better placed at the end of the mixing chain. This filter was therefore removed and placed at the output of the second mixer. The configuration chosen was a fifth order Bessel lowpass filter with cutoff frequency of 100 MHz, figure 5-9. This provided rejection of terms at multiples of the IF frequency of 70 MHz whilst giving a flat response with linear phase at the IF plus a broad margin to accommodate the bandwidth expansion generated by the DSSS signal.

The limited 70 MHz carrier signal enters the local oscillator (LO) input of the first mixer after being attenuated to +2 dBm. The sequence to be transmitted (SEQ) is limited by a transistor and diode network and AC coupled into the RF input of the first mixer stage. A back to back diode pair limit the amplitude of this signal to 0.6 V pk-pk to prevent overloading the mixer input. This also removes any transient spikes from the input data which may occur due to ringing and hence produces squarer pulses. The double-sided spectrum of the sequence input is thus centred on 70 MHz due to the first mixer, hence forming the basic BPSK signal. This broadband signal forms the LO input to the second mixer. If OOK modulation is required the digitized message data is limited by a transistor diode network and DC coupled into the RF input of the second mixer. This then on-off keys the broadband signal.

If SIK modulation is required the sequence is inverted at baseband by the data. This requires the use of an exclusive-OR gate prior to mixing. Modulation by the second mixer section is then bypassed.

The construction of this module resulted in a considerable improvement in the repeatability of results obtained from the system. In particular the system's

sensitivity to various local oscillator sources was reduced. The RF output level from this module remained within 0.5 dB of the nominal level when the input carrier level was varied between -30 dBm and +10 dBm. This was significant in the operation of an experimental system, although such problems would be alleviated in a commercial design by the use of careful pcb design.

5.3 PROTOTYPE DSSS RECEIVER.

The receiver is shown in block diagram form in figure 5-10. The broad spectrum signal is filtered by a linear phase low pass filter to remove image signals and limit the noise input to the receiver mixers. The input filter and mixers are shown in figure 5-11. The signal is then amplified by a single stage integrated amplifier and divided into two paths by a power splitter. One path is the input to the in-phase (I) mixer, the other to the quadrature-phase (Q) mixer. The local oscillator for these mixers is derived in the same way as in the transmitter using the design of figure 5.2.

A 70 MHz crystal oscillator drives a power splitter, one output of which is divided in frequency to form the system clock, and the other is amplified and limited to form the local oscillator for the IF mixers. This signal is low-pass filtered to prevent the mixing down of any remaining higher order products at the input. Two oscillator signals are required and hence the filtered local oscillator drives another power splitter. This power splitter generates two outputs with a relative phase of 90 degrees. Thus the local oscillator inputs to the two IF mixers have a relative phase difference of 90 degrees. This is a complex baseband receiver structure and is described in chapter 3. The outputs from these mixers are the baseband in-phase and quadrature phase channels.

The baseband signals are low-pass filtered to define

the noise bandwidth into the signal processing section of the receiver. An optional amplifier can be inserted after the filters if required. The signals are then sampled by high speed comparators. Baseband sampling occurs at the locally generated chip rate, nominally 10 MHz but switchable to 5 MHz. The outputs from the comparators are equivalent to the sign of the filtered baseband signal at the time of sampling and are held for one clock cycle (chip). These outputs are the receiver's estimate of the transmitted data and become the data inputs for the digital matched filters.

Two digital matched filters are required, one each for the in-phase and quadrature channels. These are 512 bit digital correlators into which a reference sequence can be loaded. The extra bit in the filter length is a result of using correlators of length 64 to form the 512 bit DMF. It was felt that the error introduced by the extra bit would not be significant. The sampled data stream is clocked past the reference sequence and a 9 bit parallel data output indicates the number of bits in agreement between the two sequences. The matched filter outputs are scaled and squared. The summation of these two numbers then forms the combining algorithm output representing the correlation between the input data and the reference. It is then necessary to compare the result with a threshold. If the threshold is crossed then the decision is that a reference sequence has been received. If the threshold is not crossed then the decision is that a reference sequence has not been received. The reference threshold is derived from three thumbwheel switches. The output from these switches is squared in a similar way to the matched filter outputs. This is necessary since the hardware implementation of a 'squaring' function is considerably simpler at high speed than a 'square root' function. The switch threshold is then compared with the combined matched filter outputs by a 16 bit comparator. A single bit output from this comparator indicates sequence detected or not detected. This output pulse is then stretched to 511 clock cycles and is the data

input to the delta demodulator.

The delta demodulator clock is derived by dividing the system clock by 511 as in the transmitter. The audio output from the delta demodulator is low-pass filtered to remove quantization noise.

The following sections describe the receiver modules in greater detail.

5.3.1 IF LOWPASS FILTER AND AMPLIFIER.

The IF low-pass filter is a fifth order maximally flat delay filter with a 3 dB cut off frequency of 100 MHz. The design is from standard filter tables [ZVEREV 1967]. The circuit is shown in figures 5-9 and 5-11. The filter is followed by a single stage integrated amplifier type Plessey SL560 also shown in figure 5-11. The gain of this stage is approximately 14 dB with a 3 dB cut off frequency of about 200 MHz.

5.3.2 70 MHz OSCILLATOR AND SYSTEM CLOCK GENERATOR.

The 70 MHz oscillator is similar to that of the transmitter, section 5.2.1 and has two outputs. One is divided in frequency to form the system clock, as in section 5.2.1. The other is the local oscillator for the IF mixers.

5.3.3 POWER DIVIDERS AND IF MIXERS.

Two power splitters and two mixers are required at IF. The mixer module incorporating the IF filter, amplifier, and local oscillator limiter is shown in figure 5-11. The filtered IF local oscillator signal is divided into two paths by a Mini Circuits PSCQ2-90 power splitter. This also

generates a relative phase shift of 90 degrees between the two outputs. The power splitter outputs form the local oscillator inputs for two Mini Circuits SRA-1 mixers. The output from the IF amplifier (section 5.3.1) is also split by a Mini Circuits power splitter, type PSC2-1. The relative phase of the two outputs from this splitter is 0 degrees. These signals are the RF inputs to the mixers. The mixer outputs are thus the baseband in-phase and quadrature-phase channels.

5.3.4 BASEBAND LOWPASS FILTERS AND VIDEO AMPLIFIERS.

The baseband low-pass filters are required to define the noise bandwidth into the digital section of the receiver. The filters are fifth order Gaussian to 12 dB sections with a cut off frequency of 7.2 MHz, figure 5-12. The selection of cut off frequency was based on optimum filters for pulse detection as discussed in chapter 3.

To increase the signal level into the mixer circuits video amplifiers were constructed, each comprising a cascaded pair of wideband operational amplifiers, type SL541, as shown in figure 5.13. The cascade gain is approximately 70, or 37 dBs, assuming a 50 ohm load.

5.3.5 HIGH SPEED COMPARATORS.

The function of the high speed comparators is to sample and hold the baseband signal and to produce a logic output representing the sign of the input signal. This binary level quantization thus represents the sign of the input signal at the sampling instant. The sampling rate is 10 MHz.

The device chosen is the Plessey SP 9687 [PLESSEY 1981]. This is a dual high speed comparator which is logic compatible with ECL. The analogue inputs are also

compatible with 50 ohm transmission lines. Two comparators are required, one each for the in-phase and quadrature-phase channels.

With reference to figure 5-14 the circuit operation is as follows. The clock signal at 10 MHz enters the clk A and clk B inputs. Since the SP 9687 comparator output is latched when the Latch Enable (LE) input is low and tracks the sign of the input when LE is high it is necessary to generate a narrow clock pulse on which to sample. This prevents the comparator output from trying to track noise on the input signal. If this occurs noise will be introduced onto the power supply rails. The narrow clock pulses are generated by two 74F00 Quad Nand gates. These detect positive going edges on the clock pulse and generate a negative going pulse of width 10 nS. These pulses are at TTL logic levels and are converted to ECL levels for the comparators by Motorola MC 10124 TTL to ECL translators. These also invert the clock to give the required positive going pulses. The comparator outputs are thus latched on the negative edges of LE. The comparator outputs at ECL logic levels are translated back to TTL logic levels by Motorola MC 10125 ECL to TTL translators. These outputs become the inputs to the digital signal processing section of the receiver.

5.3.6 DIGITAL MATCHED FILTERS AND ARITHMETIC FUNCTIONS.

The function of the Digital Matched Filters (DMF), figure 5-15 is to measure the correlation between the past 511 outputs from the high speed comparators with the locally generated 511 bit reference sequence. The filters are based on the TRW TDC 1023 digital correlator. This device contains two 64 bit shift registers, summing circuits, and a comparator. Eight devices are needed to perform a 511 bit correlation, which gives a total register length of 512 bits. This resulted in an uncertainty of 1 bit in the correlation result but is unavoidable as a 511

bit digital correlator device was not commercially available.

The 64 bit correlators are taken in pairs and their 7 bit outputs combined by two parallel adders, type 74LS283. These give 4 outputs, each in the range 0 to 128. Previous circuits had shown that conversion and propagation delays in the correlators and adders could restrict the maximum usable frequency of circuits of this complexity. Thus it was necessary to latch the adder outputs. This delays the inputs to the next stage of summing by one clock cycle but allows the operation of each stage to be well defined in terms of its processing delay in system clock periods. In practice the introduction of a fixed delay caused by latching is not significant since, being a known constant, it can be accounted for. The 4 latch outputs, each 8 bits wide, are again taken in pairs and combined by parallel adders. This results in a pair of 9 bit outputs, each in the range 0 to 256.

To remove the need for a further level of addition the MSB of each 9 bit output is used as a control signal to invert the lower 8 bits. This is performed by exclusive-Or gates. If the 9 bit result is 256, representing only the MSB = 1, then the lower 8 bits, which will all be 0, are inverted. Thus the lower 8 bits now represent the number 255. Hence an 8 bit result is formed with a maximum error of 1. Finally the 8 bit numbers are latched and connected onto the backplane. The overall uncertainty in the correlation is only 2 counts corresponding to the worst case of all 512 bits in agreement with the summing circuit output of 510.

Due to the number of clocked devices on each DMF board all logic and clock signals entering the board from the backplane are buffered.

Two DMF boards were constructed. The outputs are combined by the arithmetic functions board, figure 5-16.

The two 8 bit outputs from a DMF board (AR1 to AR8 and BR1 to BR8) are combined by an 8 bit parallel adder. This produces a 9 bit number in the range 0 to 510. For an m-sequence of length 511 the in-phase autocorrelation value is 511. In the absence of noise all out of phase auto correlation values are 255, plus or minus 1. Since the sequence can be received inverted, as discussed in chapter 3, it is necessary to interpret a correlation value of 0 as a correlation of 510. This is performed by exclusive-Or gates after the adders. If the result is in the range 256 to 510 then by removing the MSB, which has a value of 256, the corresponding range is then 0 to 254. Similarly if the result is in the range 0 to 255 then by inverting the lowest 8 bits the result is in the range 255 to 0. Therefore if the MSB is 0 the lower 8 bits are inverted. If the MSB is 1 the lower 8 bits are not inverted. The result is an 8 bit number equivalent to the following functions :-

$$Y = \text{ABS} (X - 255) \quad ; X = 0 \text{ to } 255$$

$$Y = \text{ABS} (X - 256) \quad ; X = 256 \text{ to } 510$$

There is a loss of continuity about the point $X = 255$ to 256 since the inversion does not produce a true 2's complement result. However this only introduces an error of 1 in the output. This then gives the modified DMF output discussed in chapter 3. After the exclusive-Or gates the remaining 8 bit number is again latched. It is then necessary to combine the two DMF outputs by a suitable algorithm.

The algorithm chosen was to combine the square of the DMF outputs. Two methods of calculating the square of an 8 bit number were considered. The first involved the use of a high speed 8 x 8 multiplier chip. Devices which are capable of operation at 10 MHz are available but expensive. The second option was the use of a look-up table. The advantage of this method is that it is possible to change the

algorithm by changing the look-up table. The device used was a 74S287, a 256 x 4 bit fusible link PROM. Since the square of an 8 bit number can be 16 bits wide it is necessary to cascade the outputs of four such devices. Each device has eight address inputs, representing the numbers 0 to 255, and four data outputs. This can be expressed :-

$$\begin{aligned} \text{SQR}(y) = & (\text{PROM4}(y) \times 4096) + (\text{PROM3}(y) \times 256) \\ & + (\text{PROM2}(y) \times 16) + \text{PROM1} \end{aligned}$$

where PROM1 is the least significant PROM. Therefore if X is the address :-

$$\text{PROM1} = (X^2) \text{ MOD } 16$$

$$\text{PROM2} = (X^2)/16 \text{ MOD } 16$$

$$\text{PROM3} = (X^2)/256 \text{ MOD } 16$$

$$\text{PROM4} = (X^2)/4096 \text{ MOD } 16$$

where MOD means 'to modulo arithmetic'.

The two 16 bit outputs from the look-up tables are then latched. These are then combined in a 16 bit parallel adder giving a 17 bit result. Due to delays in these adders it is necessary to latch the adder outputs again. This output is then available on the backplane (Q0 to Q15 plus CORR). It is also connected to one port of a 16 bit comparator described in section 5.3.8. Since this output represents the sum of two squares it is necessary to take the square root of this value to calculate the result according to the analysis of chapter 3.

The square root algorithm is not easy to implement digitally. If a look-up table were used it would require over 1.3×10^5 addresses, an impractically large table for the purpose at hand. Further rounding of the square root values or a transfer to floating point arithmetic would be

required to evaluate the square root. For this reason the threshold comparison is made between the sum of squares and the threshold value squared. This does not change the statistics, but merely requires comparison of a wider range of numbers.

5.3.6 THUMBWHEEL SWITCH ENCODER.

Three thumbwheel switches are used to set a threshold value in the range 0 to 255, figure 5-17. Each thumbwheel switch has a 4 bit binary output. Hence the total switch output is a 3 digit binary coded decimal (BCD) number. This must be converted to a 16 bit binary number representing the square of the threshold value. The simplest method is again to use look-up tables. Three tables are required since the switch outputs are BCD. The first table, T1, covers the range 0 to 99, the second, T2, the range 100 to 199, and the third, T3, the range 200 to 255. The most significant switch selects which range and hence which look-up table is required. To simplify the remaining decoding the 2 lower switch outputs are considered to be an 8 bit number. It is then necessary to program the appropriate data at the appropriate addresses.

Four PROMs are required in each range giving a total of 12 for the encoder. The 4 PROMs give a 16 bit output which is the binary representation of the square of the BCD input. The programming algorithms are given below :-

For range 1 (0 to 99)

$$\text{DATA} = (10 \times \text{Switch2}) + \text{Switch1}$$

For range 2 (100 to 199)

$$\text{DATA} = (10 \times \text{Switch2}) + \text{Switch1} + 100$$

For range 3 (200 to 255)

$$\text{DATA} = (10 \times \text{Switch2}) + \text{Switch1} + 200$$

For each range

$$\text{PROM4} = (\text{DATA}^2) / 4096 \text{ MOD } 16$$

$$\text{PROM3} = (\text{DATA}^2 - \text{PROM4}) / 256 \text{ MOD } 16$$

$$\text{PROM2} = (\text{DATA}^2 - \text{PROM4} - \text{PROM3}) / 16 \text{ MOD } 16$$

$$\text{PROM1} = (\text{DATA}^2 - \text{PROM4} - \text{PROM3} - \text{PROM2}) \text{ MOD } 16$$

The addresses are calculated accordingly :-

For X = 0 to 99

(This being the switch range on the 2 lowest switches)

$$\text{AddUp} = (X/10) \text{ MOD } 16$$

$$\text{AddLo} = X - 10 \times \text{AddUp}$$

$$\text{Address} = 16 \times \text{AddUp} + \text{AddLo}$$

These are not contiguous addresses as the inputs are BCD. The 16 bit output from the thumbwheel switch encoder forms the other input to the 16 bit comparator.

5.3.8 DIGITAL COMPARATOR.

The digital comparator, also shown on figure 5-16, comprises 4 cascaded 74S85 comparators, each capable of comparing a 4 bit number. Due to the high clock rate of 10 MHz and the requirement for a comparison on every clock cycle it was necessary to use a very high speed logic family, 74S, at the expense of power consumption. This is a penalty resulting from not taking the square root of the algorithm output. However, it was felt that the use of a

high speed 16 bit comparator was preferable to the use of a 131072x16 bit look-up table.

The comparator array has two 16 bit inputs. One input, (A), is the output from the DMF combiner. The other, (B), is the output from the thumbwheel switch encoder. The most significant comparator has an output which goes high if A is greater than B. This is then latched on the next clock cycle and becomes the 'correlation detected' signal, CORR.

5.3.9 CORRELATION PROCESSOR AND DELTA DEMODULATOR CLOCK GENERATOR.

The function of the correlation processor, figure 5-18, is to stretch the correlation pulse (CORR) to 511 clock cycles, this being the length of the transmitted sequence. Since a sequence is transmitted if the delta modulator output at the transmitter is at logic 1 then the correct detection of a sequence should represent 1 data bit input at logic 1 to the delta demodulator at the receiver.

The circuit comprises 3 74LS93 counters. When the counters are reset the clock input is enabled. The counters then begin to count upwards until a count value of 511 is reached. This is detected and disables the clock input. The clock generator for the delta demodulator, also figure 5-18, is similar to the correlation processor with the exception that the clock input is not gated. The counters count upwards from 0 until the count 511 is reached. This is detected and resets the counters to 0. The delta demodulator clock is taken from bit 9 of these counters giving a mark space ratio of 255 : 256. Since there is an inversion between the clock inputs to the 2 sets of counters it is not possible for a change in the delta demodulator data to occur on a delta demodulator clock transition. The inversion arises from the NAND gate in the correlation processor clock input. If an alternative digital source was applied at the transmitter then the

delta demodulator data output becomes the received bit stream data.

5.3.10 DELTA DEMODULATOR AND LOWPASS FILTER.

The delta demodulator, figure 5-19, converts the data from the correlation processor into an audio waveform. The integrator and syllabic filter values are equal to those of the delta modulator. The audio output is lowpass filtered by an elliptic filter [MOTOROLA 1976], figure 5-20.

5.3.11 RECEIVER SEQUENCE GENERATOR.

The sequence generator, also shown in figure 5-18, operates on the same principle as that used at the transmitter and as such generates the same sequences. Since the matched filters are loaded with a sequence once only, at switch on, it is possible to use a low clock frequency for loading. This then allows the use of exclusive-OR gates in the feedback circuit without propagation delay problems and avoids the need for the parity generator device. The sequence generator circuitry also generates 511 clock pulses to clock the reference sequence into the matched filters.

5.4 CONCLUSIONS.

A practical DSSS transmitter/receiver pair have been constructed on a modular basis to experimentally verify some of the theoretical ideas presented earlier in the thesis. Several problems were encountered during construction, and redesigns were necessary in the area of the DMF. In particular cascading 64 bit devices to produce a 511 bit DMF required some overhead in hardware to allow operation at the desired 10 MHz chip rate. This additional hardware was in the form of an array of parallel adders and

latches. It produced a delay of the correlation output from the DMF and was due to propagation delays within the adder circuits. These reduced the number of additions that could be performed within one chip period resulting in the need to latch intermediate results.

Another problem encountered at the transmitter was that of producing sequences at high bit rates. The problem was caused by the use of exclusive-OR gates in the shift register feedback network. The use of parity checkers eliminated this problem and sequences could then be generated at clock rates in excess of 10 MHz for shift register lengths up to and including 9 bits. Clearly both the above problems could be eliminated by suitable improvements in technology which result in the availability of appropriately high speed logic devices.

The performance of the RF sections of both the transmitter and receiver was, to some extent, dependent on the configuration and equipment used for any particular experiment. The problem was traced to the mixer sections where the addition of a limiting and filtering network on the local oscillator input resulted in a massive improvement in the repeatability of results. The cause was undoubtedly signal breakthrough and non-linearity within the original mixer modules. In a commercially manufactured item the problems of pick-up could be reduced by careful PCB layout and since the local oscillator would be a permanent fixture the problems associated with amplitude variations would be eliminated. For research purposes the system required flexibility and therefore it was necessary to isolate the mixers from amplitude fluctuations as far as possible. However in the long term the availability of high clock rate devices may lead to the possibility of generating the IF signal using digital techniques and thus remove many of the problems encountered in analogue design and circuit layout.

The processes of designing and constructing the DSSS

system led to the identification of several critical areas which could lead to the degradation in performance of such a system. Despite extensive time and effort the construction was deemed valuable in the opinion of the author as the majority of the problems uncovered would have been missed by simulation alone.

At the end of the project the system was available in a working state. It is hoped that the hardware will be used to continue the investigation into DSSS systems, and in particular to obtain results in a multipath environment.

REFERENCES

- COOPER G.R. and NETTLETON R.W. (1977) 'Efficient Spectrum Utilization with Wideband Signals', Report No. TR-EE 77-12, School of Electrical Engineering, Purdue University, West Lafayette, Indiana, USA
- GOLD R. (1967) 'Optimal Binary Sequences for Spread Spectrum Multiplexing', IEEE Transactions on Information Technology, Volume IT-13, pp 619 - 621.
- GOLOMB S.W. (1967) 'Shift Register Sequences', Holden-Day.
- IQD (1983) 'Crystal Catalogue', IQD Ltd, North Street, Crewkerne, Somerset, TA18 7AR, England.
- JAYANT N.S. (1974) 'Digital Coding of Speech Waveforms: PCM DPCM and DM Quantizers', Proceedings IEEE No 62 pp 611 - 632.
- MIDDLETON D. (1965) 'Topics in Communication Theory', McGraw Hill.
- MOTOROLA (1976) 'Linear Applications Notes', Motorola Ltd, Motorola House, 69 Buckingham Street, Aylesbury Buckingham, HP20 2NF, England.
- PETERSON W. and WELDON E. (1972) 'Error Correcting Codes', MIT Press, 2nd edition.
- PLESSEY (1981) 'ECL and Instrumentation Integrated Circuit Handbook', Plessey Semiconductors Ltd, Kembrey Park, Swindon, SN2 6BA, England.
- SKAUG R. and HJELMSTAD J.F. (1985) 'Spread Spectrum in Communication', Peter Peregrinus.

ZVEREV A.I. (1967) 'Handbook of Filter Synthesis', John Wiley and Sons.

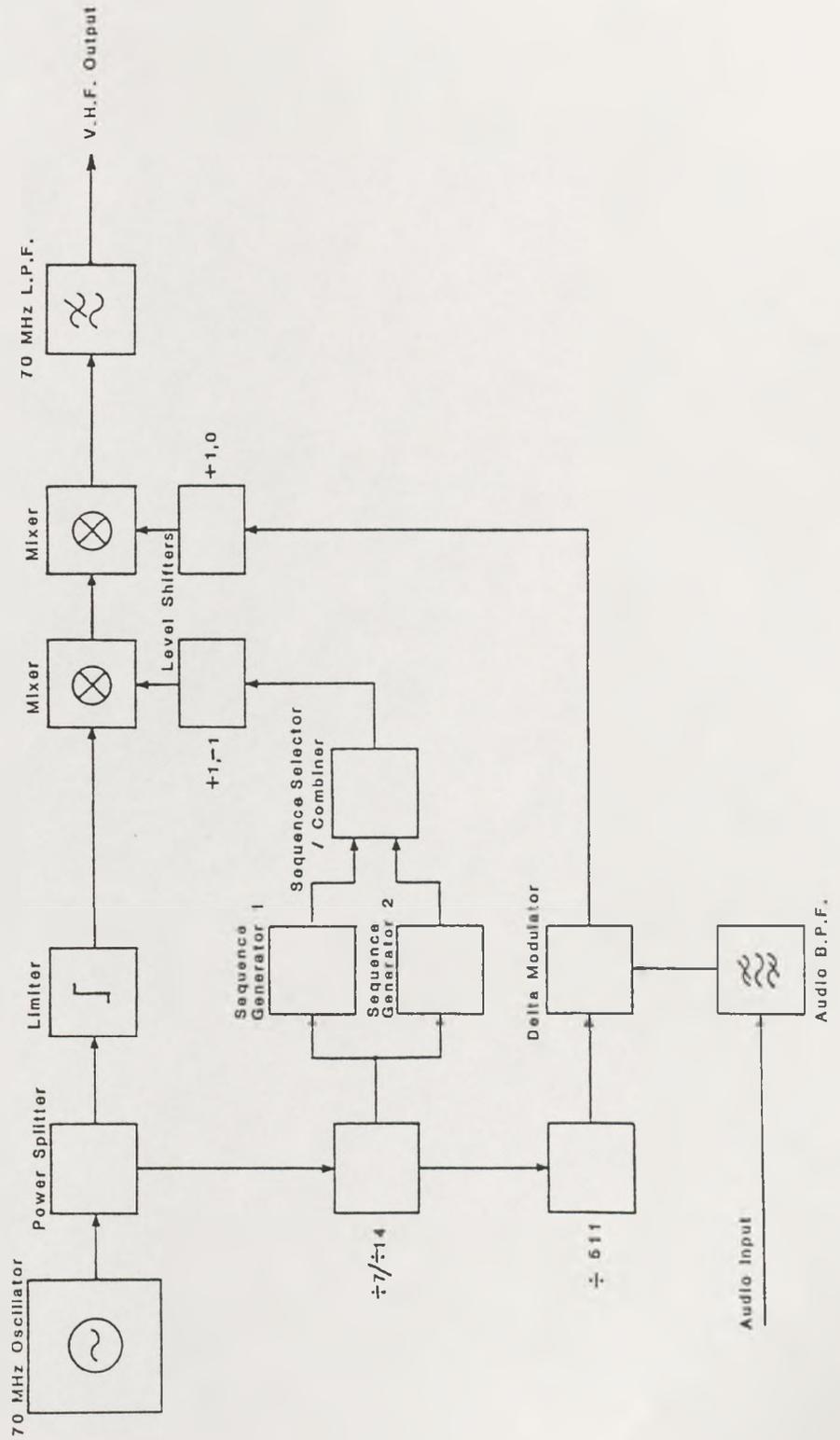


Figure 5-1. Practical DSSS transmitter block diagram

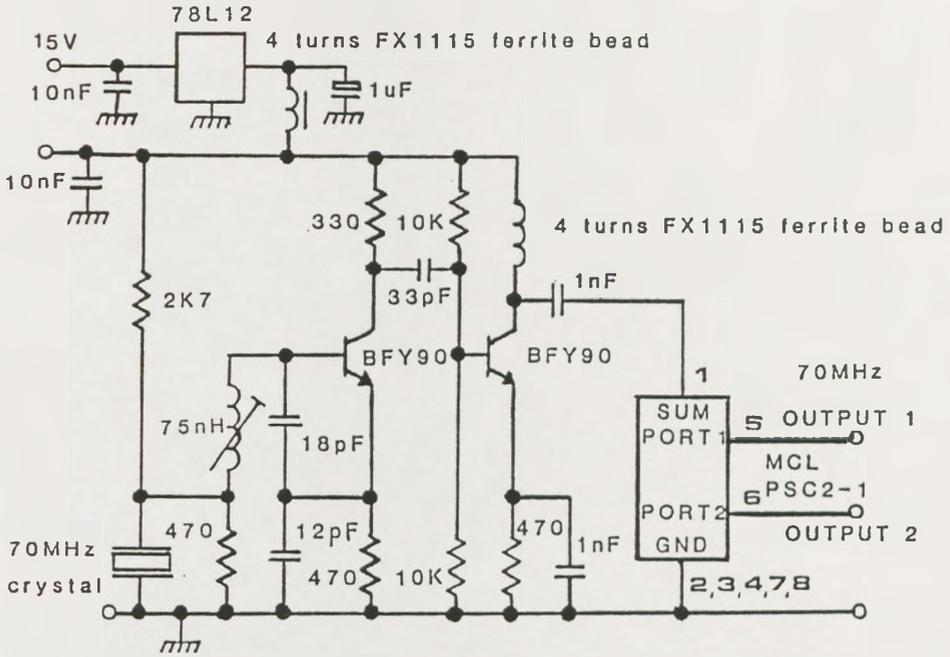


Figure 5-2. 70 MHz oscillator.

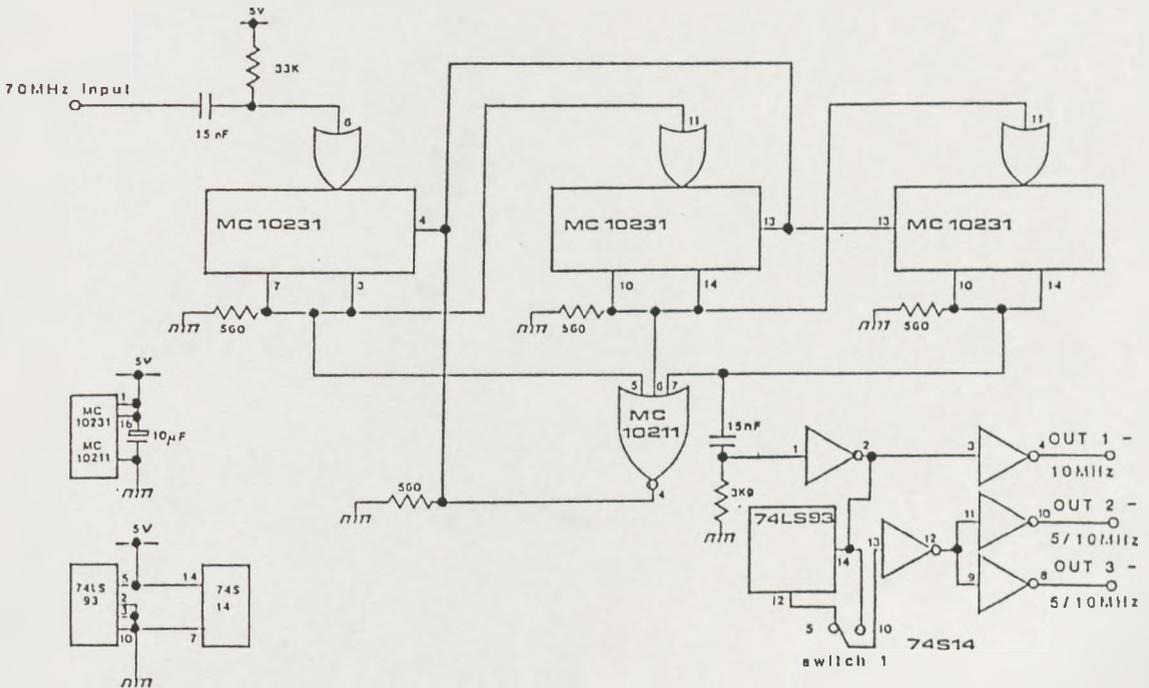


Figure 5-3. 70 MHz divider and system clock generator.

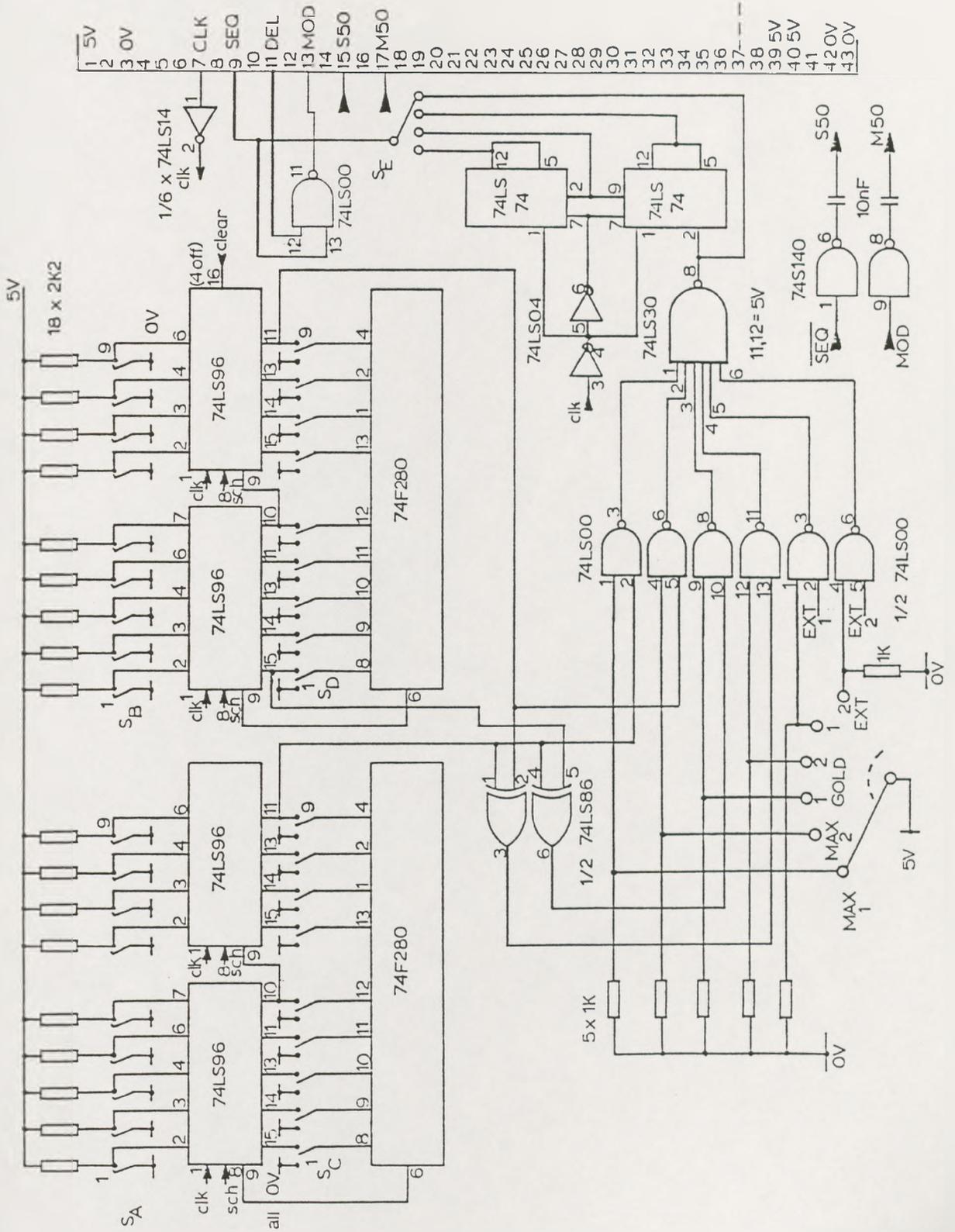


Figure 5-4. Transmitter code generator.

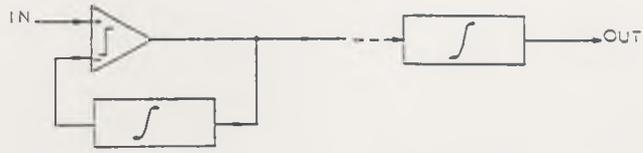


Figure 5-5a. Basic linear delta modulator/demodulator block diagram.

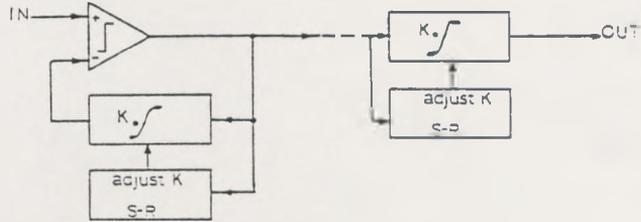


Figure 5-5b. Continuously variable slope delta modulator/demodulator block diagram.

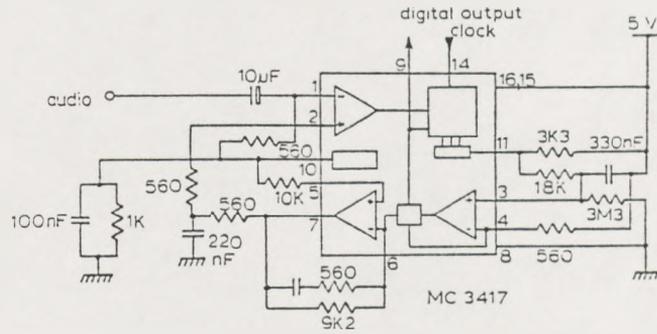


Figure 5-6. CVSD modulator.

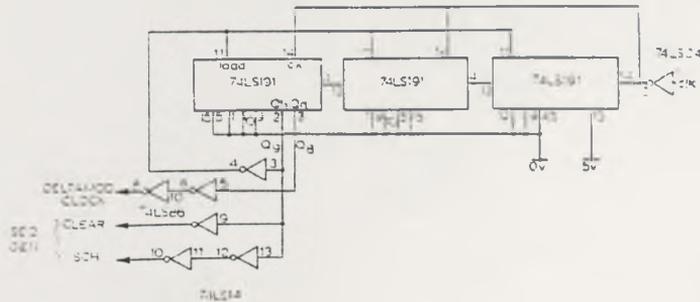


Figure 5-7. CVSD clock and synchronization generator.

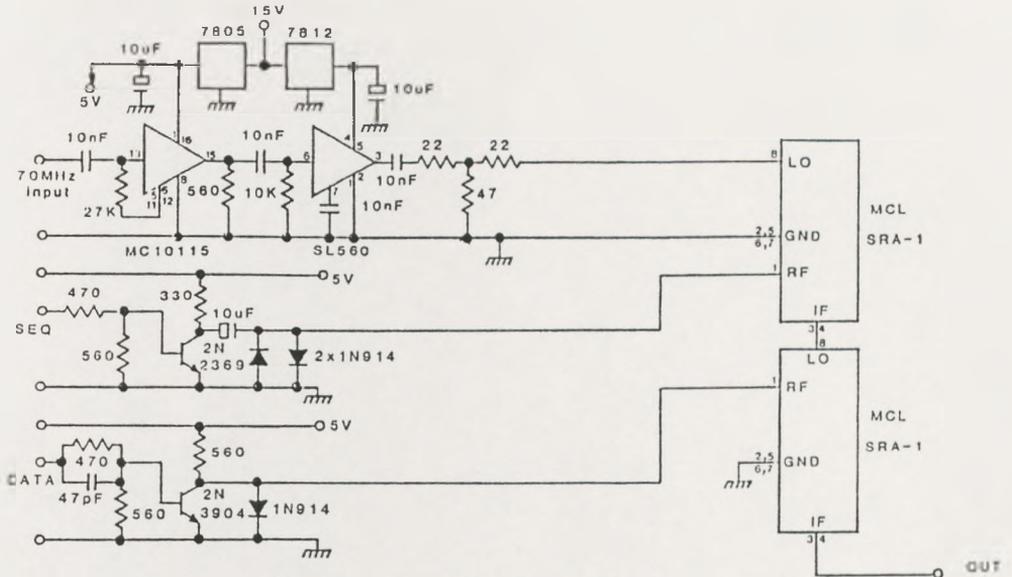
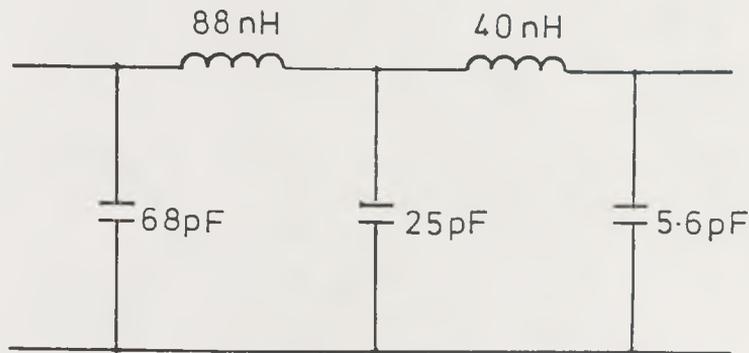


Figure 5-8. Transmitter IF modulator section.



V.H.F. Linear Phase L.P.F. $f_0 = 100\text{MHz}$

Figure 5-9. IF lowpass filter.

Gaussian to 12 dB
linear phase

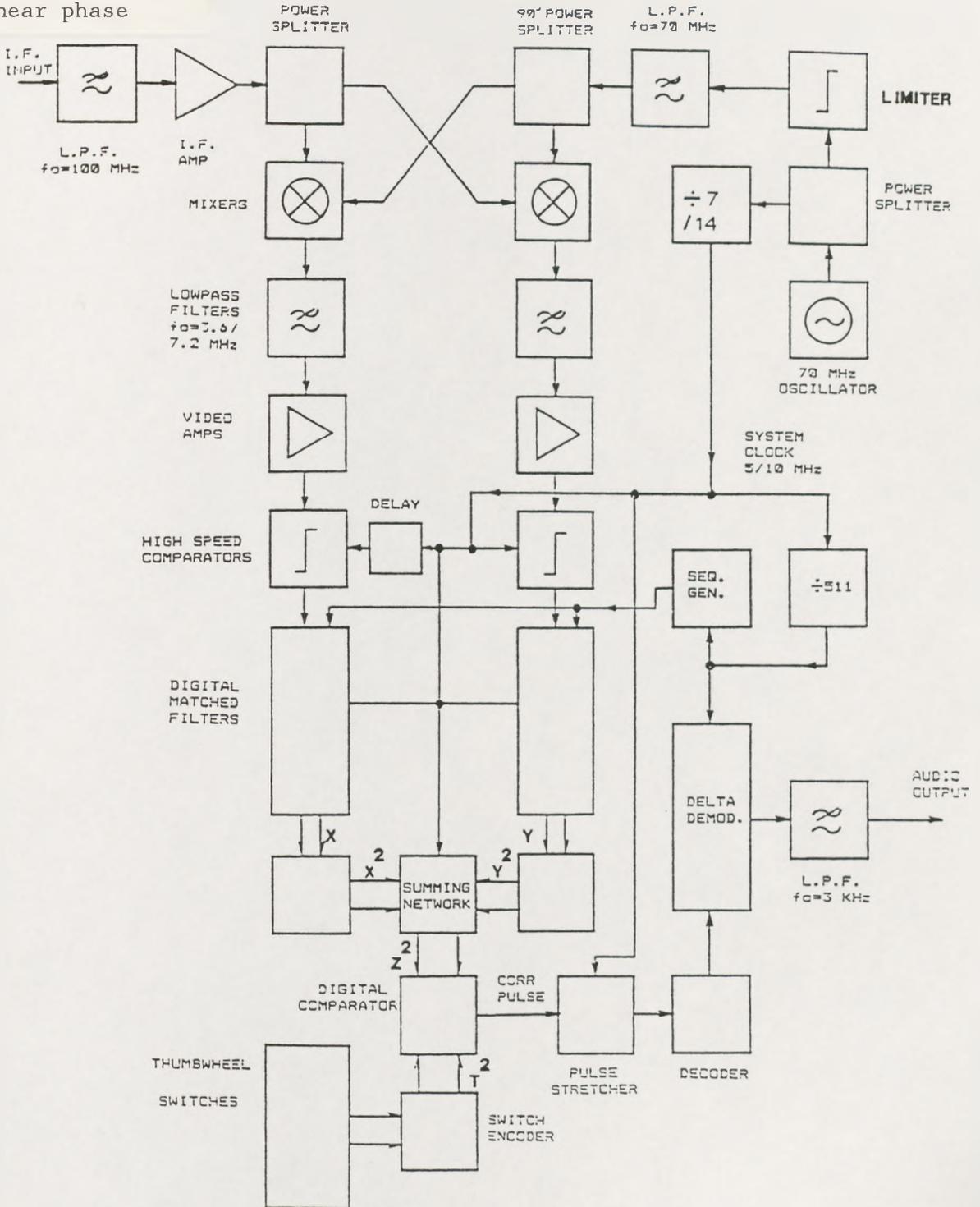


Figure 5-10. Practical DSSS receiver block diagram

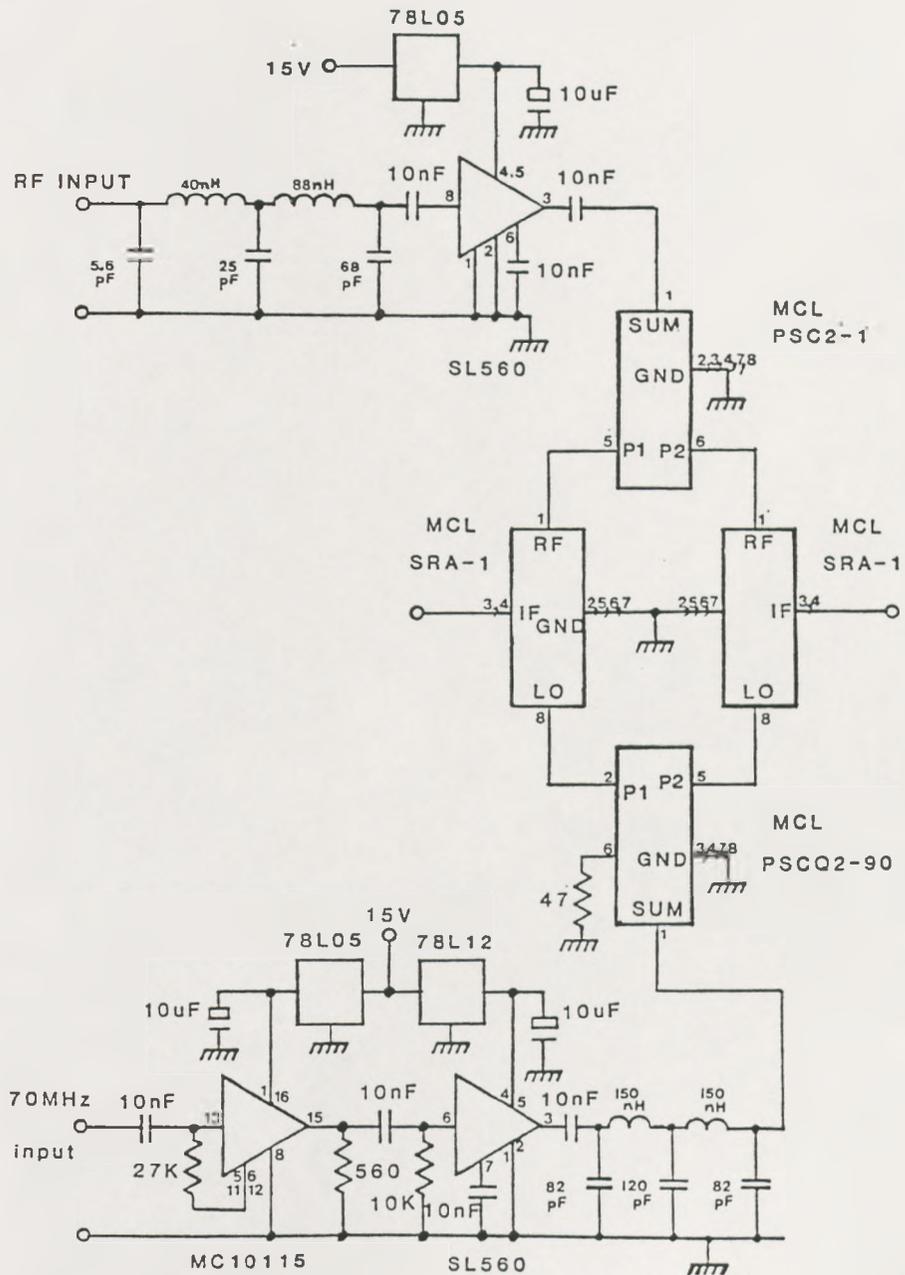


Figure 5-11. Receiver input filter and complex baseband converter.

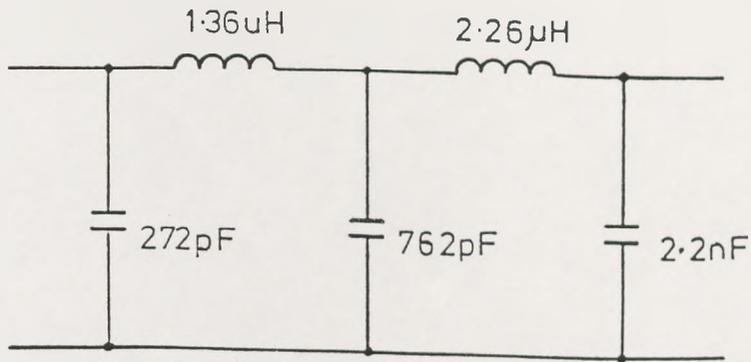


Figure 5-12. Baseband lowpass filter.

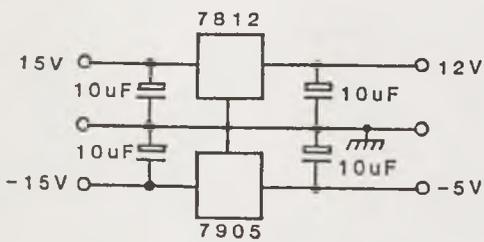
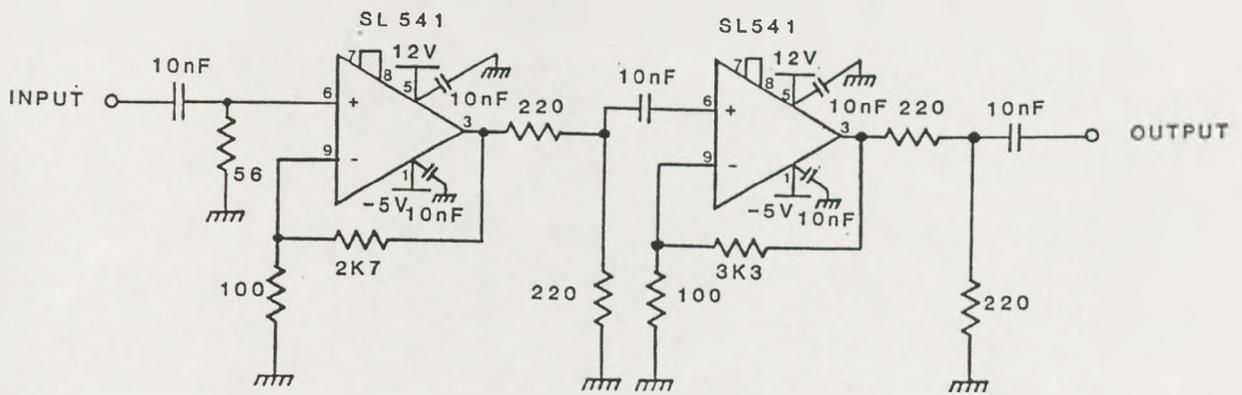


Figure 5-13. Video amplifier.

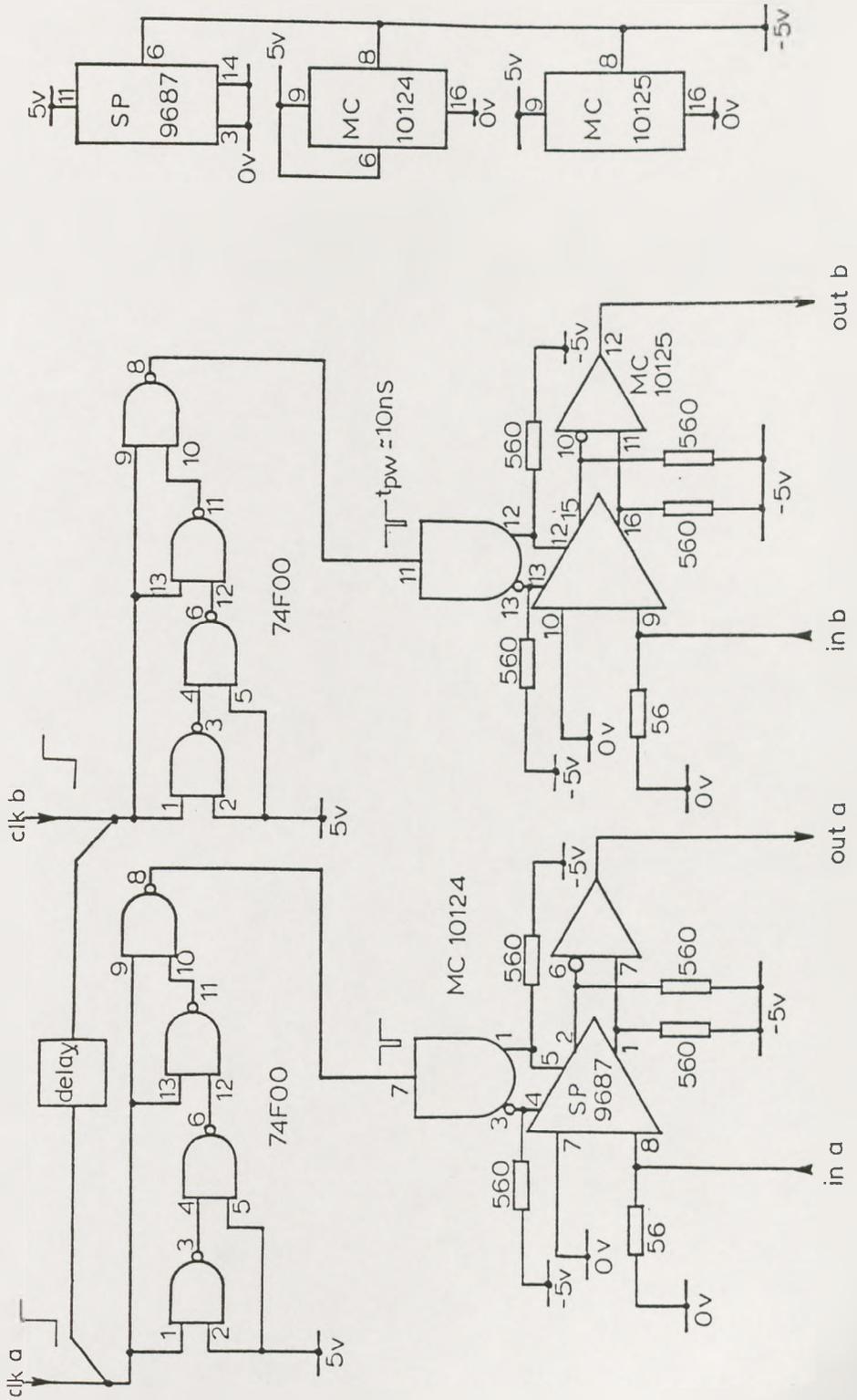


Figure 5-14. High speed comparators.

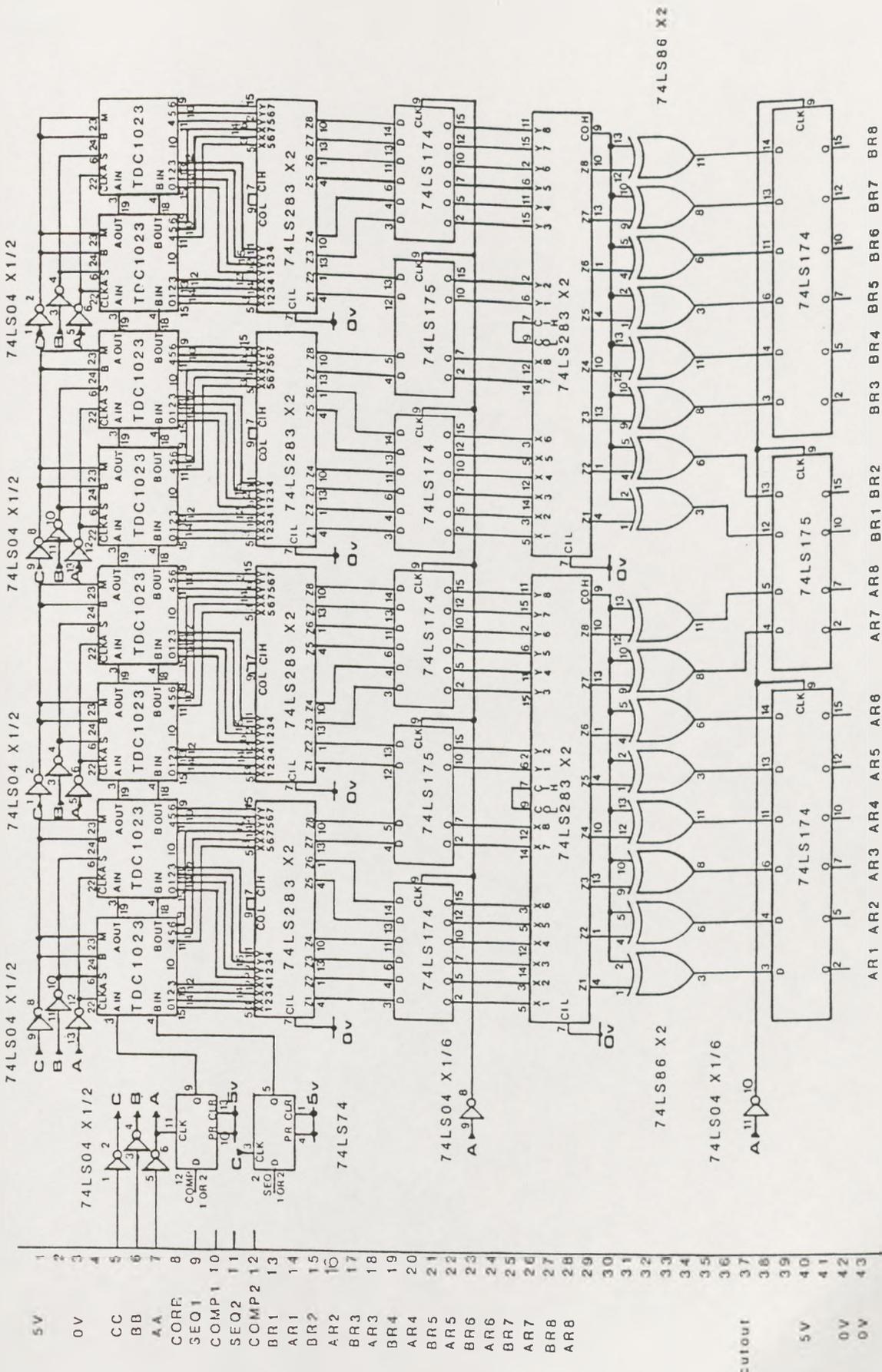


Figure 5-15. Digital matched filter.

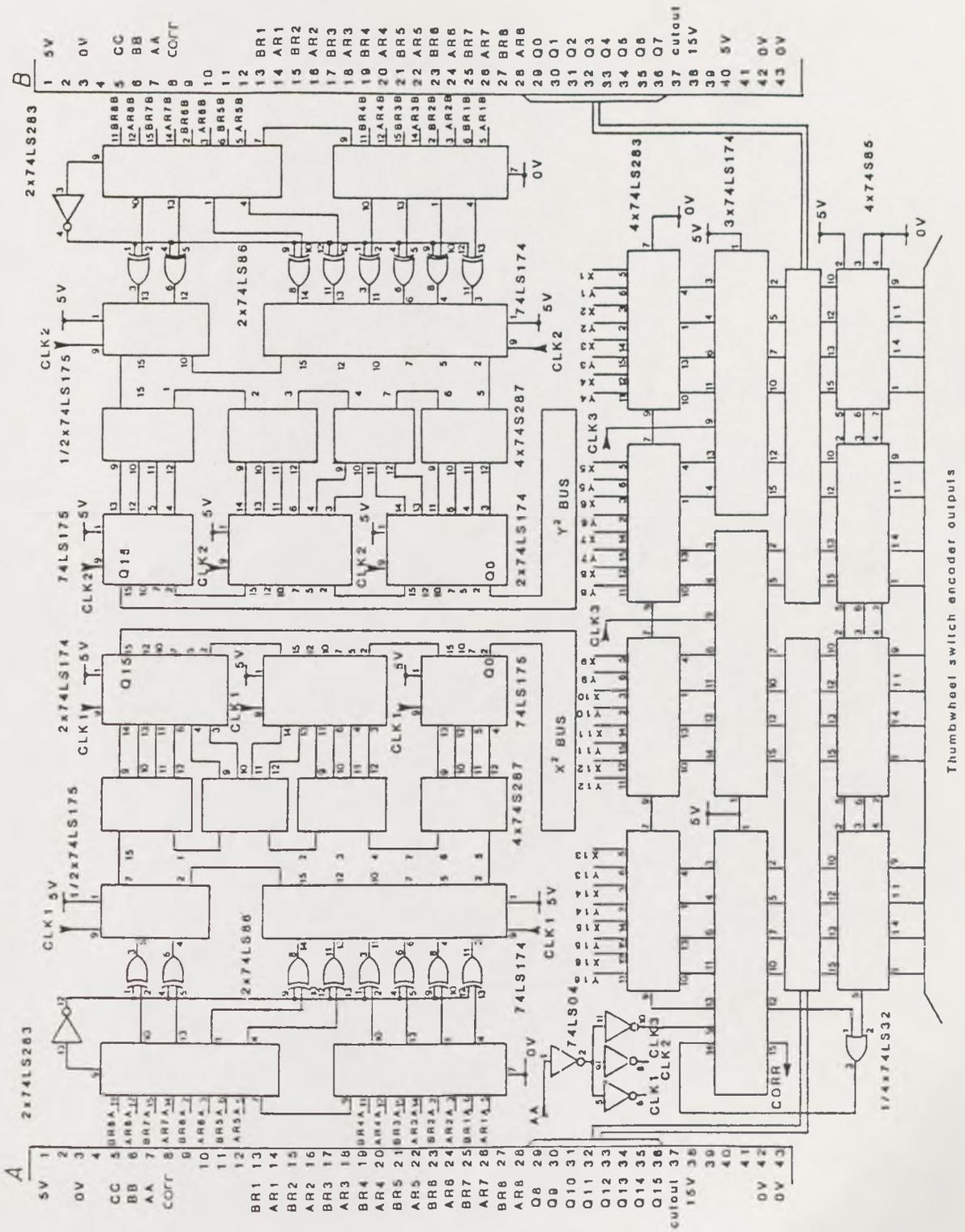


Figure 5-16. Matched filter combiner, arithmetic functions, and digital comparator.

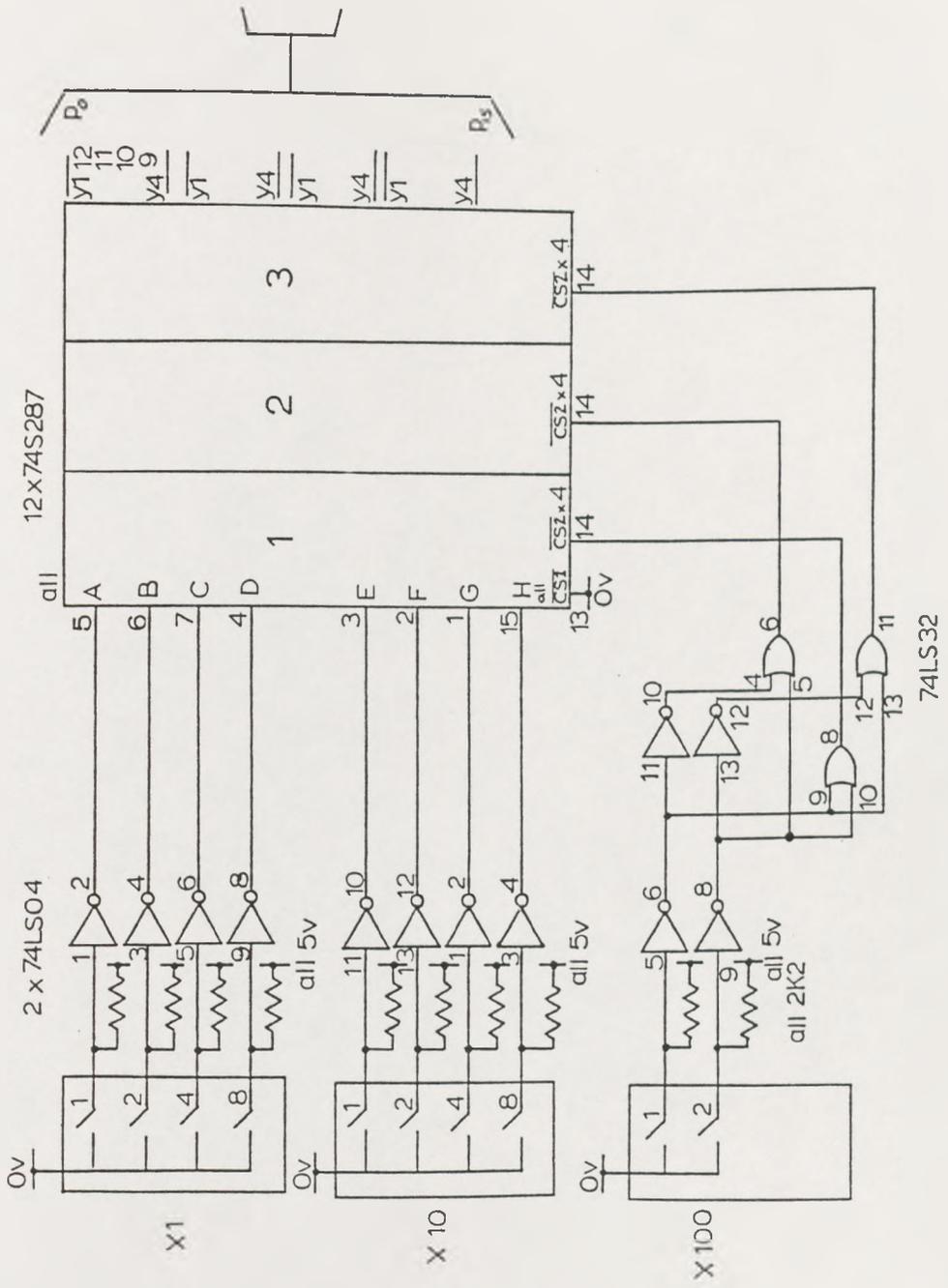


Figure 5-17. Thumbwheel switch threshold encoder.

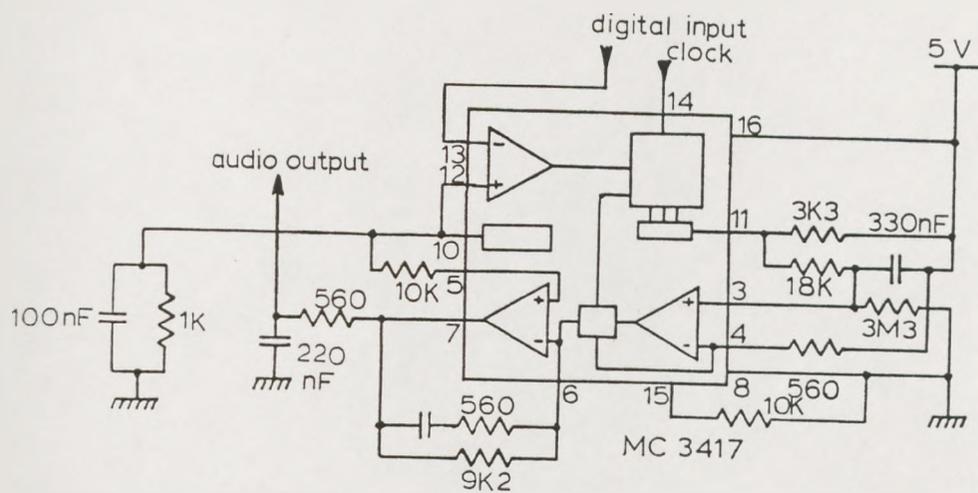


Figure 5-19. CVSD demodulator.

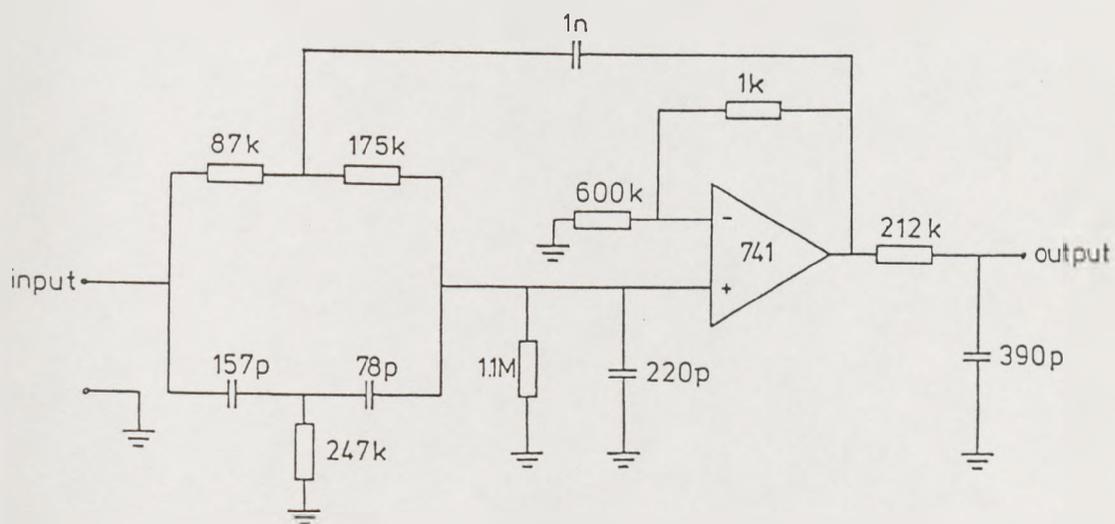


Figure 5-20. CVSD demodulator audio lowpass filter.

CHAPTER 6

PRACTICAL RESULTS

6.1 INTRODUCTION

This chapter presents results obtained from the practical system described in chapter 5. Measurements have been made at baseband and RF, and the results support the theory presented in chapters 2 and 3. In general collection of results for a particular experiment took between 10 seconds and 2 minutes. This compares favourably when compared to the likely execution times of simulation techniques.

In many cases results have been taken assuming bipolar PSK modulation for which the expressions describing the relationships between energy per chip and noise power spectral density (E_c/N_o), signal voltage and noise rms voltage, and chip error rate are taken as those presented in chapters 1 and 2.

6.2 BROADBAND NOISE GENERATOR.

Since an accurate wideband noise generator was not available a suitable generator was constructed from a cascade of proprietary amplifiers (5xPlessey SL560) with a screened resistive input. This was found to give a steady output rms noise level at stable room temperature. To enable results to be compared against well established theories it was felt necessary to characterize the noise source, and 2 techniques were applied with a view to achieving this.

As a first measure the noise power spectral density was measured over a broad bandwidth, 50 to 90 MHz, with an averaging spectrum analyser. This range was chosen since it

was centred at the receiver's IF, and also had a sufficiently wide double sideband bandwidth to exceed the bandwidth of the lowpass filter once downconverted to baseband. The result was a spectrum flat to within 1 dB over this range when averaging was not applied. With the average taken over 16 spectra the result was flat to within 0.25 dB. This represented the approximate precision to which the instrument display could be read, suggesting that the noise generator provided reasonably 'white' noise in spectral terms.

The second characterization technique, a statistical approach, was a direct measurement of the noise amplitude probability distribution. The noise signal, centred at 70 MHz, was bandpass filtered and downconverted to baseband by the receiver front end, and the lowpass filtered signal from the I channel was taken to a sampling oscilloscope. A fixed sampling rate of 100 KHz was selected to ensure that the baseband noise bandwidth at the input of the sampler was in excess of the sampling rate by a factor of approximately 100. Under this condition the noise samples were assumed independent, allowing a representative measure of the amplitude distribution to be made. From this the amplitude probability distribution was found and results are presented in figures 6-1a and 6-1b. In 6-1a 25500 samples were taken and the mean and variance calculated. These were then used to calculate the expected curve for a Gaussian function with these values. This is shown as the continuous line. Against this plot the number of occurrences within each quantization level of the sampling oscilloscope is shown, and this is normalized to the number of samples to give the probability of occupation. These values were not plotted as a continuous curve since we can only assume that the quantization error was linearly distributed. Figure 6-1a suggests reasonable agreement between practical and theoretical results.

To increase the averaging of errors the sample set was increased by a factor of 20 to 510000 samples. This is

shown in figure 6-1b, again plotted against the theoretical curve based on the experimental results. Here it can be seen that there is good agreement between theoretical and experimental results and also that a considerable smoothing of the errors has been achieved by increasing the size of the sample set.

Over much of the theoretical curve the sampled data is a very close fit, although there is some discrepancy for low deviations. Three possible effects which would account for this are now considered, although there are many potential sources of error in practical systems.

Firstly the sampling oscilloscope may have been non-linear over this region. This explanation is unlikely, however, since non-linearity for a typical 8 bit ADC of the type found in the sampling oscilloscope is of the order of 1 LSB. The curves suggest an error somewhat in excess of this magnitude.

The second, and more likely explanation, is that the signal presented to the ADC did not occupy the lower quantization levels in proportion to the expected statistics. This may have been due to a variety of reasons but one possible explanation is that the noise generator produced values which did not quite have the expected statistics. Alternatively the noise signal with linear addition of a bipolar signal, resulting from low level carrier breakthrough or periodic interference from power rails, would have statistics with reduced values about the mean. For example if a low amplitude square wave was added to the noise signal the resulting statistics would be the sum of 'n' noise distributions, each having a mean value corresponding to the occupancy of one of 'n' quantization levels for all possible square wave amplitudes. The result would be a 'smearing' of the statistics, particularly noticeable in the region close to the mean value of the noise statistics. In the practical system low level noise voltages were evident although it was not generally

possible to correlate these with expected signals.

The third explanation is that the theoretical curve based on the experimental mean and variances was not the 'best-fit' Gaussian curve for the data. Indeed the amplitude probability distribution may, in practice, have been closer than shown in the low amplitude regions at the expense of accuracy in the larger amplitude regions. The total area under the graph could have been made constant but the amplitude at the mean reduced and the variance increased.

These arguments suggest that it is not possible to state categorically that the noise generator produced voltages with a Gaussian amplitude probability distribution. However the results were encouraging enough to suggest that the noise generator could be used for experimental work and could further be assumed to produce noise with close to Gaussian amplitude statistics.

6.3 CHIP ERRORS

In order to assess the accuracy bounds of measurements made at the DMF output the input chip error rates were measured against theory. This result was also felt to be of value in further assessing the performance of the noise generator.

Chip error rate measurements were made by adding the output of the sequence generator to the baseband noise signal. The signal was therefore uncurbed rectangular pulses in the presence of assumed Gaussian noise. Prior to addition each source was passed through a variable attenuator to allow the absolute signal and noise levels to be varied. The two signals were then added in a resistive summer and amplified by the baseband amplifier. This amplification was necessary to bring the amplitude of the noise source to a level comparable with that of the signal

source and thus provide the limiter with sufficient drive level. The amplified signal was then limited and sampled by a hard limiter/sampler network as described in chapter 5, and the output compared with the unattenuated, noise free input signal source. An exclusive-OR gate was used to indicate a detection error and the output of this gate was sampled at the opposite phase of the sampling clock to prevent false outputs. A logic counter was used to count the errors over a specified number of input samples.

The transmission and counting process was controlled by a NorthStar Horizon microcomputer, allowing many samples to be taken and processed with operator intervention only necessary when a change of input snr, and hence attenuator settings, was necessary. Due to the high bandwidth of the system, and a requirement to average errors as far as possible, the number of transmitted chips was set to 10^9 . At a chip rate of 10 Mbits/second each result took approximately 110 seconds to obtain, including the time overhead introduced by filing data.

The range over which useful measurements should be made was considered very carefully. Several factors contributed to the choice of input snr values.

It was desirable to include input values which would allow the processing gain of the DMF to be evaluated whilst at the extremes not be subject to unavoidable input noise or distortion at the amplifier or hard limiter inputs. The problems encountered were due to the practical limitations of the circuits used.

Firstly there was a minimum amplifier input noise level which was acceptable for measurements since input levels below this would be swamped by amplifier self noise, local breakthrough of signals from the power supplies, and signals induced by switching transients in other areas of the system. These unwanted noise signals prevented the correct calculation of input noise level, and therefore it

was necessary to ensure that the input noise level was considerably in excess of the unwanted signals. This led to the definition of a maximum input snr, since the minimum input noise level was fixed, and the signal amplitude was limited by the power supply rails.

At the other end of the scale the minimum input signal snr was again restricted by the local input noise to the receiving amplifier but the externally generated noise amplitude had to be restricted to prevent limiting of the input amplifier. In practice the rms level of the input noise signal was always kept below 1/6th of the amplifier power rails to prevent clipping, and hence unknown or indeterminate modification of the noise statistics.

A final consideration was the presentation of results. This can best be envisaged with reference to the actual results, figure 6-2. The parameter subject to experimental error is the input snr, since there are inevitable error tolerances on the measured signals, and on the attenuator values. The latter were subject to absolute errors of less than 0.5 dB each, giving a total maximum snr error of plus or minus 1 dB. Measurements at very low input snrs appear very close to the theoretical curve, and this would be the case for quite large errors in input snr since the derivative of the curve is low in this region. Thus measurements made at extremely low input snrs would have to be shown on a separate graph with suitable scaling. Similarly at high snrs small systematic errors in measuring the input snr could lead to a large discrepancy between practical and theoretical results.

Such errors could disguise the value of the results and it was felt that the most interesting and valid comparison could be made over the 'knee' of the curve, where the general shape of the results could be evaluated. Figure 6-2 is the result of the above experiment performed for 10^9 chip samples. Overall the results are within 1 dB of the theoretical values, as indicated by the horizontal

error bars, and follow the general shape at the curve 'knee'. This level of agreement is within the expected bound of experimental error, and supports the theory presented and the experimental evidence of the noise generator.

The ability to achieve this result allowed further experiments to be performed with a known level of confidence in the input snr measurement. From this result a bound could be placed on the expected chip error rate at the hard limiter output.

6.4 SYMBOL ERRORS

Having measured the input chip error rates for a given range of input snrs it was then possible to extend the practical results to consider the effect of the DMF. For this the output from the hard limiter was fed to the input of the DMF, and code inversion keying was imposed upon the transmitted sequence. This was produced by exclusive-ORing the sequence generator output with a pseudorandom sequence generated at the symbol rate, giving one symbol for every 511 chips transmitted. The DMF output was sampled at the code epoch and the sign of this compared with the sign of the modulating signal, in a manner identical to the chip error rate experiment. Again the error signals were counted by a digital counter, with the entire process being under microcomputer control, except for changes of the signal attenuators.

Figure 6-3 shows the results obtained for 10^6 transmitted symbols. The bold line is the expected output symbol error rate for the given input chip error rate, assuming the full processing gain of $10\log(511)=27$ dBs was available. The dotted line represents the expected symbol error rate assuming the ideal processing gain less 2 dBs for hard limiting, as predicted in chapter 2. The crosses represent the results obtained from the practical system,

and these are in good agreement with the theoretical curve with the expected 2 dBs loss in processing gain.

The range of measurements was restricted at the low input snr end by internally, or locally, generated noise as described in section 6.3. At the high input snr end the measurement was limited to the number of symbols that could be transmitted in a reasonable period of time. The transmission rate for symbols was 10 Mbits/511 seconds, or approximately 19.5Kbits/second. Therefore the time taken to transmit 10^6 symbols was a little under 1 minute. If a measured symbol error rate of 10^{-7} was required it would have been necessary to transmit 10^8 symbols, requiring approximately 1 hour and 25 minutes per reading. This would have been an impractically long period of time due to the possibility of long term drift in the noise generator. Thus fewer symbols could be transmitted for this experiment when compared to chip error rate measurements, due to the reduced bandwidth at the DMF output.

The noise generator output was measured by the statistical technique of section 6.2 prior to each symbol error rate measurement to minimize any errors introduced by long term drift in the amplitude of the generator output. This gave an immediate check on any possible variations in the noise statistics, but was not used as a measure of input noise level. The experiment of section 6.3 had indicated the bound of error on input snr measurement to be of the order of 1 dB, attributed to the errors arising from the attenuators. To verify a processing gain loss of 2 dBs would obviously require a more accurate measure of input snr or input error probability. Therefore the average input error probability was measured prior to each symbol error rate measurement. Since the average input error probability could be measured to approximately 1 part in 10^5 this was clearly a more accurate means of evaluating the loss in DMF processing gain due to hard limiting.

The symbol error rate results of figure 6-3 show

excellent agreement with the predicted results and clearly indicate that the expected loss of 2 dBs, due to hard limiting, is realised in practice.

6.5 DMF OUTPUT PROBABILITY DISTRIBUTION

Section 2.3 of chapter 2 described the expected output probability distribution for the DMF, evaluated from the Binomial theorem, for code in phase, out of phase, and noise only conditions. Experimental results were taken from the practical system to support this theory, and these are described in this section.

To generate the DMF output pdf it was necessary to sample the output of the DMF at the appropriate code phase. To achieve the desired speed a high speed latch was constructed, allowing the DMF output to be sampled and held. The latch was enabled by a synchronizing pulse from the code generator, with a suitable delay generated by a shift register network allowing the sampling phase to be modified. Once the DMF output was sampled the microcomputer was enabled to read the latch and then re-enable the latch itself. The sampled value was then stored and the process repeated until 10^6 samples were obtained. The output probability was then evaluated as the number of occurrences of a particular output divided by the number of samples taken.

The time taken by the microcomputer to read and save a value was slightly longer than one code period, and as a consequence the stored values did not represent consecutive outputs from the DMF at a particular code phase, but DMF outputs at the same code phase spaced several symbols apart. This was not considered to be a serious drawback as there should not be any 'memory' between consecutive DMF outputs. The code generator was not modulated by any other sequence, as was the case in the measurement of chip errors. Instead the code was transmitted either true or

inverted, representing the effect of modulation on the DMF output at the appropriate code phase.

Figure 6-4 shows the theoretical and practical results for an input chip error rate of 0.365, corresponding to an input snr of -12.5 dBs for bipolar PSK modulation. The dotted line represents the results expected from theory, whilst the bold line indicates the experimental results. To allow a comparison between results the theoretical values corresponding to a particular DMF output are plotted slightly to the right of the value to which they correspond, and the experimental results are plotted slightly to the left. Close examination of figure 6-4a in the region where the DMF output equals 352 will verify this. From theory the mean DMF output should be $m(1-p_e)$, or approximately 324. The diagram shows that the theoretical and practical curves have excellent agreement in the region given, and the slight shift in mean value between theoretical and practical results is certainly within the expected experimental error bounds described in the previous sections. The figure is purposely shown over a restricted region to highlight the closeness of the fit between theoretical and practical results. Displaying the data over the entire range of possible DMF outputs would not have produced such a detailed result. The tails of the curve have been omitted in the figure but experimental results showed equally good agreement with theory in these regions, and it was again felt that more information could be conveyed by examining the fit in the most active region of the curve. This was also supported by the evidence from the symbol error rate measurement, since this represents the total probability of any DMF output being below $m/2$, or 255.

The experiment was repeated with the transmitted sequence inverted, simulating a sequence or code inversion induced by external modulation. The result is shown in figure 6-4b, where the expected output of mp_e , or approximately 186, is again achieved in practice. Also the

curve fit is very good, the mean value again shifted by an amount consistent with the bound of error on the measurement of input snr, and the consequent bound of error on the input chip error rate.

A limiter input snr of -12.5 dBs would give an effective DMF output snr of 12.5 dBs, assuming a loss of 2 dBs for hard limiting, which might be acceptable in some applications. To test the theory towards the lower end of the input snr region the experiment was repeated with a reduction in input snr of about 6 dBs and the resulting distributions are shown in figure 6-5. The chip error rate was 0.434, corresponding to an input snr of about -18.6 dBs for bipolar PSK modulation. Comparison with the previous figure shows that the distribution mean has shifted in line with the predicted value, to approximately 289 for the code true case and 222 for the code inverted case, and that the curve fit is still excellent for code true and inverted conditions.

Having established that the DMF output pdf showed agreement with theory for code in phase conditions the experiment was extended to include code out of phase and noise only conditions. For out of phase measurements the phase of the synchronizing pulse into the sampling latch was modified by introducing a delay of 6 chip periods. Results taken for verification purposes suggested that a delay of a single chip period would have been adequate, as might have been expected since the sequence used for the experiment had a uniform out of phase autocorrelation function. Figure 6-6a shows the output pdf for the out of phase condition corresponding to an input chip error probability of 0.434, as before. Again there is excellent agreement over the range presented, and similarly for the noise only case, figure 6-6b, where the phase of measurement was obviously unimportant since the signal was absent.

Results were also taken at higher input snr values,

and these also showed excellent agreement with theory. The graphs were not included for brevity, since the distributions were more 'peaky' and therefore did not highlight the errors due to input snr uncertainty. Further the most interesting area of use for the DMF is at low input snrs and thus, where possible, results have been presented reflecting this.

The practical measurement of the DMF output pdfs required the construction of high speed data acquisition and interfacing circuits, but the results obtained, and their obvious support of the theory presented in chapter 2, far outweighed the effort involved. The distributions obtained by experiment show excellent agreement with theory, to within the bounds of input snr measurement.

6.6 MULTIPATH PROPAGATION MEASUREMENTS

With the system set up to measure the DMF output probability distribution it was possible to investigate the effect of a multipath signal on the DMF output. To do this the output probability distribution was found for various relative code phases, this being adjustable by advancing or delaying the synchronizing pulse received from the transmitter. The mean output was evaluated from the output distribution, obtained in the same manner as for the previous experiment. The multipath signal was generated by adding the sequence generator output to a delayed version of itself, the 3 chip period delay being introduced by a shift register network. The output from the noise generator was added to this via an attenuator to simulate reception of a dual path signal in the presence of noise. The snr of both versions of the sequence could be varied by an attenuator placed between the sequence generator outputs and the summing block. Results were taken for 6 values of input error probability, or input snr, and for two attenuator settings between the sequence outputs.

The results are presented in figures 6-7 to 6-9. Figure 6-7a shows the DMF mean output for an input error probability of 0, that is no noise present, relative to the non-delayed sequence. The delayed sequence has an amplitude 6 dBs below that of the non-delayed version, since a 6 dB attenuator was added to the delayed sequence output prior to the summing network. Without noise present the amplitude of the non-delayed signal always exceeds that of the delayed signal due to the effect of the attenuator, and hence the delayed signal does not show in the mean DMF output, and further the mean DMF output at the code in phase position is 511, as expected. Since the amplitude of the delayed sequence is always smaller than the non-delayed sequence the hard limiter input is always of the correct sign, and hence chip errors cannot occur. At code out of phase positions the mean DMF output is equal to the expected value for the out of phase autocorrelation of the sequence used. Again no bias can be introduced by the delayed signal since each delayed chip has an amplitude below that of its corresponding chip in the delayed sequence.

The error probability is increased by increasing the attenuation between the sequence generator output and the receiver without increasing the attenuation in the noise generator circuit. Figure 6-7b is the result of increasing the error probability of the non-delayed signal to 0.1, equivalent to an input snr of about -1 dB assuming BPSK modulation. The delayed signal is now apparent in the mean DMF output, appearing as a peak 3 chip periods after the code epoch, since with noise added there is always a finite probability of incurring a chip error, resulting in a bias towards the expected mean at the DMF output. For the delayed sequence the snr is about -7 dBs, corresponding to a chip error probability of around 0.26. In figures 6-7c to 6-7f the error probability for each sequence is increased, with in phase error probabilities of 0.2 in figure 6-7c, 0.3 in figure 6-7d, 0.4 in figure 6-7e, and finally 0.5 in figure 6-7f. Again assuming BPSK modulation these figures

correspond to input snrs of -4.5 dBs, -8.6 dBs, -15 dBs, and for an input error probability of 0.5 the input snr is 0 in terms of a voltage ratio (logarithm not defined). For the delayed sequence figures 6-7b to 6-7e correspond successively to input snrs of -7 dBs, -10.5 dBs, -14.6 dBs, and -21 dBs with the same constraint applied to the definition of the input snr for an input error probability of 0.5, figure 6-7f.

The diagrams indicate the trend of decreasing amplitude in mean DMF output with decreasing snr, for both non-delayed and delayed sequences and further that each sequence appears as a distinctive peak in the DMF output. In the noise only case the mean DMF output is always close to 256, or half the sequence length. In all cases the mean out of phase output is close to the expected out of phase autocorrelation value of 256.

The experiment was then repeated with the delayed sequence logically inverted. The results are presented in figures 6-8a to 6-8f, for comparison with figures 6-7a to 6-7f. The inversion can clearly be seen as an inversion in the DMF output away from the value of 256, but a negative deviation in this case. The non-delayed peak is still apparent as a positive excursion. This is an important result as it suggests that the phase of the carrier of the delayed signal can be measured by a pair of DMFs in the complex baseband receiver, as for the non-delayed sequence. The theory of this is described in chapter 3. As a final investigation into the properties of the DMF as a tool for evaluating multipath signals the amplitude of the delayed sequence was increased by 3 dBs, to a level only 3 dBs below that of the non-delayed sequence. The experiment was then repeated for the same input error probabilities for the non-delayed sequence. The results are shown in figures 6-9a to 6-9f, where it can be seen that the peaks corresponding to delayed and non-delayed sequences are apparent. However detailed analysis shows that the mean DMF amplitude at the in phase condition is not as great as for

the previous experiment, nor as would have been predicted by the mean output for a DMF without a delayed sequence added, implying that the delayed path has some effect upon this.

To investigate this result it is necessary to return to the theory presented in chapter 2. For a sequence of m chips, each independent but with a common error probability of p_e , the mean DMF output was given by $m(1-p_e)$. In the multipath case the input signal is the result of adding 2 pulse streams together, and then adding noise to this. Some chips will add constructively, and some destructively. The mean DMF output can therefore be found by considering the input sequence to consist of two sub-sequences, corresponding to enhanced or reduced amplitude chips. Evaluating the mean output for each sub-sequence and adding will give the mean DMF output for the combined sequence. In general, we are interested in the situation whereby one sequence is in phase, and one is out of phase. For the in phase sequence there would be ' m ' agreements, 511 in the practical system, in the absence of noise. For the out of phase sequence there would be approximately $m/2$, or 255, agreements in the absence of noise, this being the out of phase autocorrelation value of the m -sequence used for the experimental procedure. Thus there are approximately $m/2$ enhanced amplitude chips and $m/2$ reduced amplitude chips. We now have the definition of the two sub-sequences, and can evaluate their composite error probabilities.

Putting p_I as the error probability of the in phase sequence, and p_O as the error probability of the out of phase sequence, we have from chapter 2

$$\text{snr}_I = Q^{-1}(p_I) \quad (6-1)$$

and

$$\text{snr}_O = Q^{-1}(p_O) \quad (6-2)$$

where snr_I and snr_O are the ratios of signal voltage to noise rms voltage for the respective sequences. Since the same noise source is added to both sequences the values may be added or subtracted directly to give the snr of the enhanced or reduced sequences. The error probability for the two sequences can now be found from

$$p(\text{IP})_{\text{EN}} = Q(\text{snr}_I + \text{snr}_O) \quad (6-3)$$

$$p(\text{IP})_{\text{RE}} = Q(\text{snr}_I - \text{snr}_O) \quad (6-4)$$

where $p(\text{IP})_{\text{EN}}$ is the resulting error probability for the enhanced sequence when the in phase code has the greatest amplitude. Similarly $p(\text{IP})_{\text{RE}}$ is the resulting error probability of the reduced sequence for this condition. The mean DMF output is then given by the mean number of agreements for the total sequence

$$\begin{aligned} \text{DMF}(\text{IP})_{\text{mean}} &= (1-p_{\text{EN}})m/2 + (1-p_{\text{RE}})m/2 \\ &= (1 - p_{\text{EN}} - p_{\text{RE}})m/2 \end{aligned} \quad (6-5)$$

For the delayed code in phase case, shown as a delay of 3 chips in the diagrams, (6-4) must be modified to take account of the relative amplitudes.

We use the property $Q(x) = 1 - Q(-x)$ to find the resulting error probabilities

$$p(\text{OP})_{\text{RE}} = 1 - Q(\text{snr}_I - \text{snr}_O) \quad (6-6)$$

giving

$$\text{DMF(OP)}_{\text{mean}} = (1 - P_{\text{EN}} - P_{\text{RE}})m/2 \quad (6-7)$$

In equation (6-6) the original in phase sequence is now the delayed, or multipath, component.

To highlight the effect of delayed sequences on the DMF output the resulting outputs and those predicted by (6-5) and (6-7) are presented in tables 6-1a to 6-1e along with the expected theoretical output assuming no delayed sequence.

TABLE 6-1
 PREDICTED AND MEASURED DMF OUTPUTS
 WITH MULTIPATH COMPONENTS

Table 6-1a

At code in phase position - 6 dB delay non-inverted

P_I	P_O	Calc. DMF output	Meas. DMF output	Thy. output no delay
0.1	0.26	437	429	460
0.2	0.34	398	393	409
0.3	0.40	354	358	358
0.4	0.45	305	301	307

Table 6-1b

At code out of phase position - 6 dB delay non-inverted

P_I	P_O	Calc. DMF output	Meas. DMF output	Thy. output no delay
0.1	0.26	315	307	378
0.2	0.34	316	305	337
0.3	0.40	302	295	307
0.4	0.45	281	274	281

Table 6-1c

At code out of phase position - 6 dB delay inverted

P_I	P_O	Calc. DMF output	Meas. DMF output	Thy. output no delay
0.1	0.26	196	204	133
0.2	0.34	195	207	174
0.3	0.40	209	216	204
0.4	0.45	230	238	230

Table 6-1d

At code in phase position - 3 dB delay non-inverted

P_I	P_O	Calc. DMF output	Meas. DMF output	Thy. output no delay
0.1	0.18	416	416	460
0.2	0.28	388	381	409
0.3	0.36	351	344	358
0.4	0.43	305	302	307

Table 6-1e

At code out of phase position - 3 dB delay non-inverted

P_I	P_O	Calc. DMF output	Meas. DMF output	Thy. output no delay
0.1	0.18	343	324	419
0.2	0.28	339	317	368
0.3	0.36	320	301	327
0.4	0.43	291	280	291

In general there is good agreement between the theoretical values predicted from the composite sequence analysis and the results obtained from the experimental rig. Comparing these results with the expected output with no delayed sequence present, that is a single sequence in the presence of noise only given by the fifth column in each graph, it is apparent that multipath components which are strong relative to the desired sequence cause considerable loss in expected DMF output at the code epoch. If many paths of equal strength were added, with randomly distributed code phases, then the DMF output statistics would approach those of the desired sequence in the presence of noise only since the collection of uncorrelated codes would appear as Gaussian noise.

Since codes with uniform out of phase autocorrelation properties are frequently advocated for 'channel sounding' applications the results obtained in this experiment are significant, since the amplitude of the delayed path relative to the direct path is not given directly by the relative DMF outputs. Instead it is necessary to use the equations presented in this section to evaluate the effect of each path on the DMF output relative to all others. Thus a knowledge of the number of significant paths is required, and this is obtainable from a DMF by examining the number of peaks detected in the delayed signature. Therefore the DMF is a valuable tool in evaluating multipath signals, since a fully implemented complex baseband receiver, as described in chapters 3 and 4, is capable of providing both amplitude and phase information about the received multipath signal.

6.7 EXAMINATION OF ALGORITHM MEAN OUTPUT

The previous experiments confirmed the theory pertaining to the operation of the DMF presented in chapter 2. Experimental results were then needed to support the arguments concerning the correct way to combine the outputs from a pair of DMFs, in the complex baseband receiver, as discussed in chapter 3. Of prime interest in this area was the need to examine the output from the algorithm whereby the DMF outputs are squared and added. Simulation had already suggested that the straightforward summation of the two DMF outputs would not lead to a smooth average output for a given input snr when the carrier phase was varied relative to the receiver local oscillator. This analysis showed, section 3.7, that a smoother output, or the virtual elimination of phase dependency, was achievable by calculating the modulus of the resultant output by treating the I and Q channel DMF outputs as a complex number. Further this approximation improved as the snr was reduced.

The DMF outputs were squared numerically, as described in section 5.3.5 of chapter 5, and the results summed. In view of the high speed of operation, no attempt was made to evaluate the square root of this number, as discussed in section 5.3.5, since threshold detection of the algorithm output is possible without resorting to this rather complex operation. Thus use of the terminology 'square root' algorithm is perhaps misleading, but is used as an alternative to the term 'summing' algorithm, used to describe the situation whereby the DMF outputs are added without squaring.

For this experiment the RF section was added, to allow phase offsets to be introduced between the carrier of the generator and the receiver local oscillator. To ensure frequency coherence both transmitter and receiver were operated from the same RF synthesiser, operating at 70 MHz, but measured lengths of coaxial cable were placed between

the synthesiser and the receiver to allow known phase delays to be introduced. As before a high speed parallel latch and microcomputer were used to collect data from the algorithm output. The carrier phase was varied between 0 and 90 degrees, in steps of 10 degrees. For each phase value the algorithm output was sampled at the code epoch, the latch again being enabled by a synchronizing pulse from the transmitter.

A total of 10^4 symbols were transmitted for each phase value, and the experiment was performed for 3 input snrs. Since the square root algorithm was not performed directly in hardware the square root of the sampled values was evaluated prior to plotting. To speed up the sampling process all 10^4 values were collected in real time, and the square root process performed at the end. This led to the requirement for a 20 KByte buffer in the microcomputer. The results are shown in figure 6-10 for voltage snr values of 0.1, 0.5, and 1.0, corresponding to energy snrs of -23 dBs, -9 dBs, and -3 dBs for bipolar PSK modulation. The bold lines represent the theoretical curves obtained by simulation, as described in chapter 3. These highlight the improved linearity of the algorithm approximation at low input snrs. The crosses represent values obtained from the practical system.

Figure 6-10a, resulting from an input voltage snr of 0.1, highlights the effect of experimental errors. This may have been due to some contribution from the 'dead zone' DMF outputs, discussed in section 5.3.5 of chapter 5. Since each DMF output can be in error by a count of 1 or 2, due to the simplifications made in the hardware design, the squared and summed result can be in error by a count of plus or minus 3, after rounding. However the sample set should have been sufficiently large to reduce any problems in this area and the likely explanations for any discrepancies between the two sets of results is the difficulty in measuring the input snr, the short term stability of the signal and noise generators, imbalance in

the RF section of the receiver, and the reduced number of samples taken. The later was a restriction imposed by the buffer size of the microcomputer system used. The results, in general, suggest a good fit between practical results and those derived from simulation of the theory.

Figure 6-10b shows the result from an input voltage snr of 0.5, or -9 dBs, where again the expected trend is evident. Finally figure 6-10b shows the results after the input voltage snr had been further increased to 1.0, or -3 dBs. The trend away from a straight line is just becoming evident in the practical curve, a result which is extremely encouraging and in support of the theory.

The conclusion from these results is that the square root algorithm does produce an output which is largely independent of carrier phase. Use of the RF section inevitably introduced some further experimental errors due to phase imbalance between the I and Q mixers, and some uncertainty in the absolute phase difference between transmitter and receiver carriers. However it was felt that this approach was important as it would give some information regarding the feasibility of the overall system. Undoubtedly the expenditure of further time and money could have produced circuits of higher quality, but it is difficult to predict what overall effect this would have had on the practical results.

The summing algorithm was not examined practically. There were two reasons for this. Firstly the results of the experiments to determine the DMF output statistics were encouraging enough to suggest that the theory was correct, and since the theoretical outputs from the algorithms were a simulation of all possible combined outputs it seemed extremely likely that the results from the algorithm outputs would also agree with the expected values. Secondly the theoretical evidence suggested that the square root algorithm would be the most satisfactory in terms of providing a smooth output with varying carrier phase, and

hence the preferred algorithm when considering synchronization and input snr measurement. Therefore it was felt prudent to investigate this algorithm closely, despite the added complexity required in it's implementation, in preference to the summing algorithm.

6.8 PROBABILITY ALGORITHM EXCEEDS THRESHOLDS

Having examined the square root algorithm mean output against varying relative carrier phase attention was switched to considering the techniques for code synchronization discussed in chapter 4. Two cases were considered, the probability that the algorithm output exceeded a preset threshold, and the binary integration of four DMF outputs as discussed in chapter 4, section 4.3. To make the measurement 10^6 symbols were transmitted, again with the RF section in use but a constant relative phase offset of 45 degrees was introduced between transmitter and receiver local oscillators. The RF section was required to ensure that the input signal was split between the two channels, hence allowing each DMF to produce an appropriate output distribution. As described in section 5.3.8 of chapter 5 a digital comparator was added to the algorithm output, with the reference threshold provided by a set of thumbwheel switches. A symbol synchronizing pulse was again provided from the transmitter to allow the comparator output to be sampled at the correct code phase. In common with previous experiments the measuring process was under microcomputer control, the only manual operations required being adjustment of attenuators, and hence input snr, and thumbwheel switches, for threshold selection.

For the first experiment the number of threshold crossings for the code in phase and one code out of phase condition was measured, and the average probability of threshold crossing found from this and the number of symbols transmitted. The synchronizing pulse for the out of phase condition was obtained from a delayed version of the

pulse used for the in phase case, the delay being fixed at 6 chip periods. Three input chip error rates were tested, $p_e = 0.4437$, $p_e = 0.4007$, and $p_e = 0.3274$ corresponding to input snrs of -20 dBs, -15 dBs, and -10 dBs for bipolar PSK. The results are shown in figures 6-11a, 6-12a, and 6-13a, where the bold lines represent the theoretical threshold crossing probabilities as defined in chapter 4, and the crosses represent results obtained from the practical system. In each case the curve closest to the y-axis indicates results obtained from the out of phase case, and the curve furthest from the y-axis results obtained from the in phase case. The diagrams show that for all three snrs tested there is good agreement between theory and practice, with errors likely to be due to systematic inaccuracies in measuring the input snr. This would certainly seem to be the case in figure 6-12b where all experimental results lie to the right of the expected values, although it is possible that this is in part due to the unavoidable offset introduced in each DMF output, as described in section 5.3.5 of chapter 5.

For the second experiment a 4 bit binary integrator was added to the output of the digital comparator, with a new input to the integrator being provided with each synchronization pulse. The binary integrator gave a positive output whenever 3 or 4 symbol samples were positive. In this case the output sample rate was one quarter of the symbol rate since 4 symbols were required to produce a 4 bit binary integration, and once these had been tested it was necessary to reload the binary integrator with 4 new comparator results. Consequently 2.5×10^5 samples were obtained, corresponding to 10^6 transmitted symbols. The results are shown in figures 6-11b to 6-13b for comparison with the previous results and the results obtained from the theory of chapter 4. Again good agreement is shown between theory and practice, highlighting the advantage obtained from this type of threshold decision.

Overall the results obtained in this section were

extremely close to those predicted by the analysis presented in chapter 4, and within the bounds of expected experimental error. The 4 bit binary integrator produces threshold exceedance probabilities which are markedly 'steeper' than those obtained without binary integration, a property which can be used to advantage in designing systems for rapid synchronization acquisition.

Due to time restrictions it was not possible to further the experimental results to include the synchronization schemes discussed in section 4.3.1 of chapter 4. This is in some ways regrettable but the experimental results obtained agree with the most fundamental parts of the theory upon which the further techniques are based, and therefore confidence can be expressed in their potential.

6.9 CONCLUSIONS

The practical system, described in chapter 5, was used to obtain results on the performance of the DMF and combining algorithm as discussed in chapters 2, 3, and 4. With the limited availability of published results on practical DSSS systems these experiments have been useful in establishing the bounds of performance which may be obtained from such systems.

To allow the input snr to be measured, a noise source was evaluated in terms of output spectral content, and more importantly, in terms of an amplitude probability distribution. The results suggested that examination of the source on a statistical basis was a worthwhile procedure despite being rather more complex than a simple spectral analysis. It is perhaps reasonable to state that when a practical system is to be evaluated on the grounds of statistical theorems the input conditions to the practical system should themselves be characterized in this manner. A direct comparison then becomes possible in the case of the

DSSS system, since the input signal statistics are also well defined. In this statistical evaluation the use of 2 sample sets with considerably different sizes indicated the averaging effect of a larger set of data. This might have been more difficult to prove by a simulation technique without the use of many hours of computer time. The results from the spectral analysis indicated that the noise generator produced a signal with a spectrum flat to within 0.25 dBs over the range 50 to 90 MHz. The statistical analysis further showed that the noise output probability distribution was close to Gaussian.

With the noise source characterized chip error rate measurements were made to evaluate the performance of the hard limiter and set bounds on the accuracy of the input snr measurement. The resulting error rate curve followed the general 'knee' shape of the theoretical curve, and this was considered to be the region of major interest in evaluating the snr measurement. A maximum snr error of 1 dB was found between theoretical and practical curves. As this was the approximate accuracy of the input signal attenuators it is likely that the input snr evaluated from the noise statistics and the signal amplitude was accurate to within 1 dB.

To investigate the loss in processing gain incurred by hard limiting symbol error rate measurements were made. To minimize the error in input snr measurement the input chip error rate was measured prior to each symbol error rate. The accuracy of this was approximately 1 part in 10^5 , considerably greater than the input snr measurement accuracy of approximately 1 dB. This allowed the symbol error rates to be determined for an assumed input snr derived from a known chip error rate. Again the experiment was performed over a range centred on the theoretical curve 'knee' and indicated that the expected loss of 2 dBs was experienced in practice.

The DMF output probability distributions were measured

for various code conditions, code in phase with code true and inverted, code out of phase, and for several input snrs including the noise only case. These were compared with expected distributions obtained from the Binomial theorem and in each case the experimental distribution closely fitted the corresponding theoretical distribution. Slight offsets in the mean values of the distributions obtained from the practical system were attributed to errors in input snr measurement. Experimental examination of the DMF output pdf at different input snrs has thus shown that the complex baseband receiver has the capability of measuring the average phase of the incoming signal carrier relative to the receiver local oscillator.

Since m-sequences are often advocated for channel sounding applications the response of the DMF to a multipath signal was tested. A multipath signal was formed by adding a delayed version of the sequence to the non-delayed version, where the delay was introduced at the chip level by passing the sequence through a shift register network. Noise was added as before, and the average DMF output evaluated at several code phases and input snrs. In the absence of noise the sequence with the greatest amplitude dominated, completely swamping the lower amplitude sequence since chip errors could not occur. With noise added the chip error rate increased and non-delayed and delayed paths were apparent in the DMF output. This was observed for several input snrs, where an increase in the snr produced a larger average DMF output.

Measurements were made for 2 values of relative snr between delayed and non-delayed sequences and in this case an increase in the amplitude of the average DMF output at the appropriate code phase was observed, indicating that the relative signal strengths could be measured. The amplitude of the DMF output at the correct code in phase positions was reduced from that expected for a sequence in the presence of noise only, indicating that the out of phase autocorrelation value of any out of phase sequences

at this phase has an effect on the signal amplitude. Tables have been presented and a theoretical explanation for this effect given, but the net effect is that in evaluating multipath channels all significant paths must be identified. This is possible from the DMF output, although for a greater discrimination in path delay it would be necessary to increase the chip rate or the DMF sampling rate. Increasing the chip rate would obviously have implications for the required operating speed of the DMF whereas increasing the sampling rate would require a corresponding increase in the length of the DMF. To fully characterize a channel it would be necessary to use the complex baseband receiver, with 2 DMFs. The outputs from these could be analysed in matrix form, derived from the equations given in section 6.6, and the solution would give the amplitudes and phases of each path in the multipath signal. Thus the DMF is a valuable tool for providing the information required to directly assess multipath channels.

One important requirement of the complex baseband receiver was an algorithm capable of combining the outputs from the 2 DMFs in such a manner that the output was independent of the relative carrier phase between the received signal and the receiver local oscillator. The 'square root' or modulus algorithm was proposed in chapter 3, section 3.7, and results obtained from this algorithm for various relative carrier phases and input snrs. For each input snr the algorithm mean output was measured for phase offsets ranging from 0 to 90 degrees and in each case good agreement was found between theory and practice. The results indicated that the square root algorithm is suitable for combining the DMF outputs in the complex baseband receiver and also showed the deviation from a straight line approximation at higher input snrs. This is an important result as the algorithm mean output can be used to measure the amplitude of the received signal, which would be necessary for power control in an interference limited system. Also during synchronization acquisition

loss of carrier lock is less critical as the algorithm output can be used to determine the code epoch.

Finally rapid synchronization by binary integration of the algorithm mean output, as presented in chapter 4, was examined by measurements made at the algorithm output and at the output of a short binary integrator placed at the algorithm output. The average number of threshold crossings was measured for several threshold levels and several input snrs. Results were compared with theory and indicated that the use of binary integration produced steeper threshold probability curves, which in turn would result in a reduction in false synchronization detection rates provided the detection threshold was correctly chosen.

Practical results have been obtained from the DSSS system described in chapter 5. These are by no means exhaustive, particularly in the areas of combining algorithms and rapid synchronization schemes. However they support the theoretical work presented earlier in the thesis and, are in many cases, results which could only have been obtained with days of computing time by simulation means. This is particularly true in the area of symbol error rate measurements at low snrs.

The addition of a UHF RF section, and full investigation and implementation of the tracking and control functions, would allow measurements to be made in the field. This would provide the ultimate results by which the DSSS system could be compared against other schemes.

Figure 6-1a.

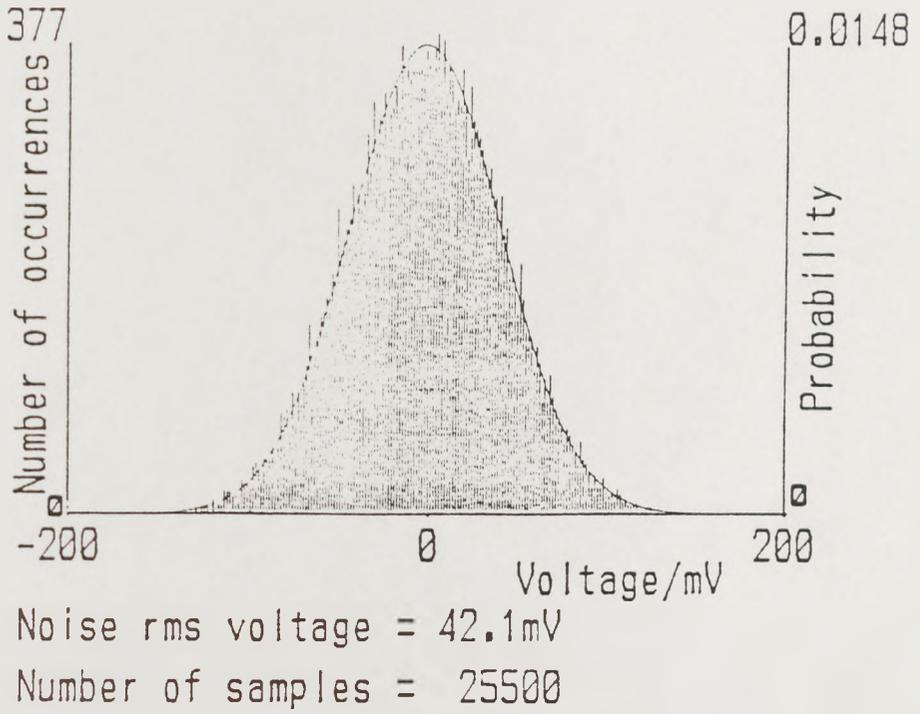


Figure 6-1b.

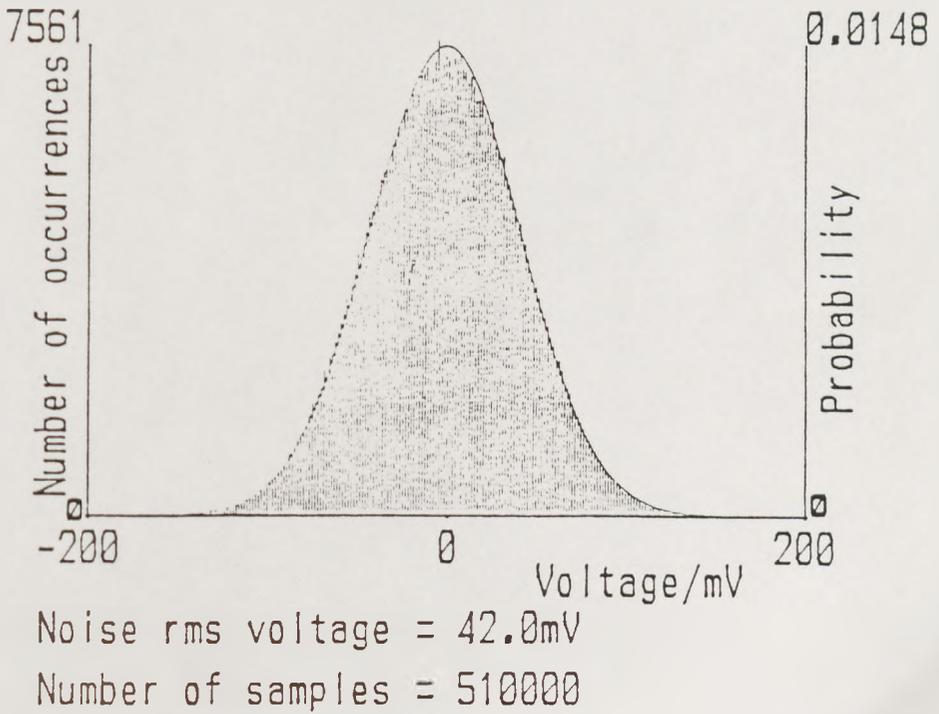


Figure 6-1. Noise generator amplitude probability distribution.

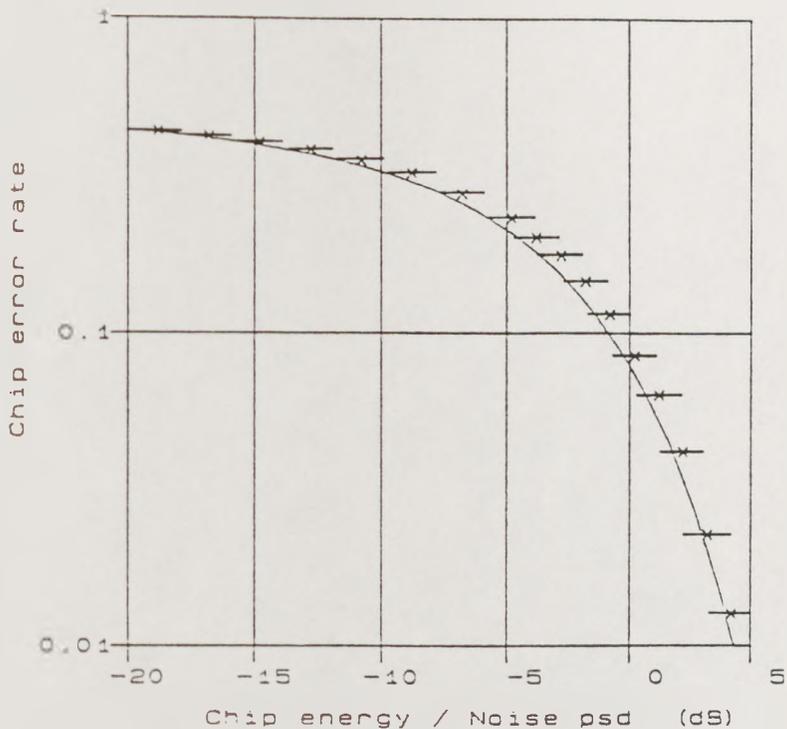


Figure 6-2. Measured chip error rate.

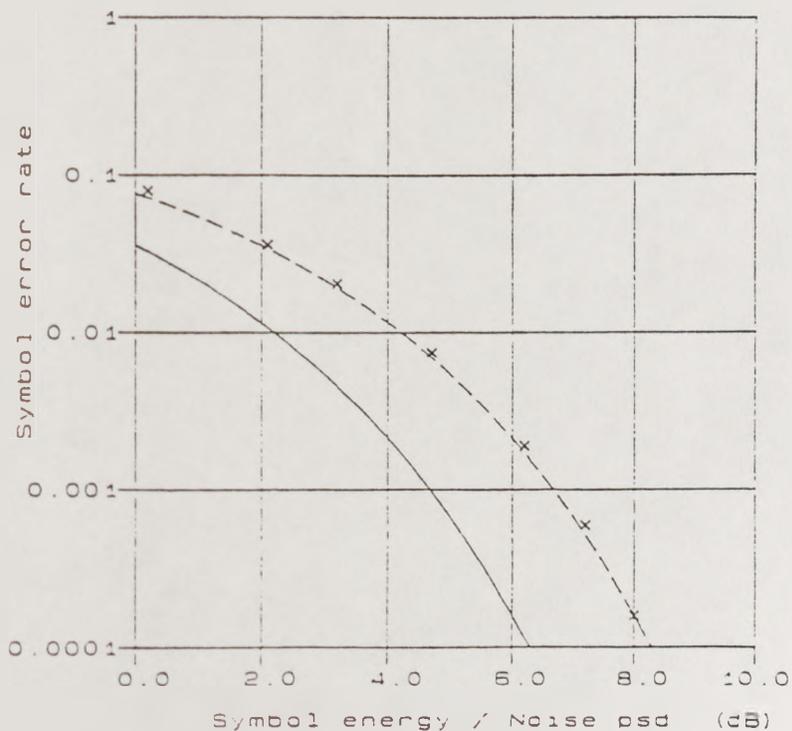


Figure 6-3. Measured symbol error rate.

- Theoretical limit.
- - - Limit less 2 dBs for hard limiting.
- x x x Measured error rate.

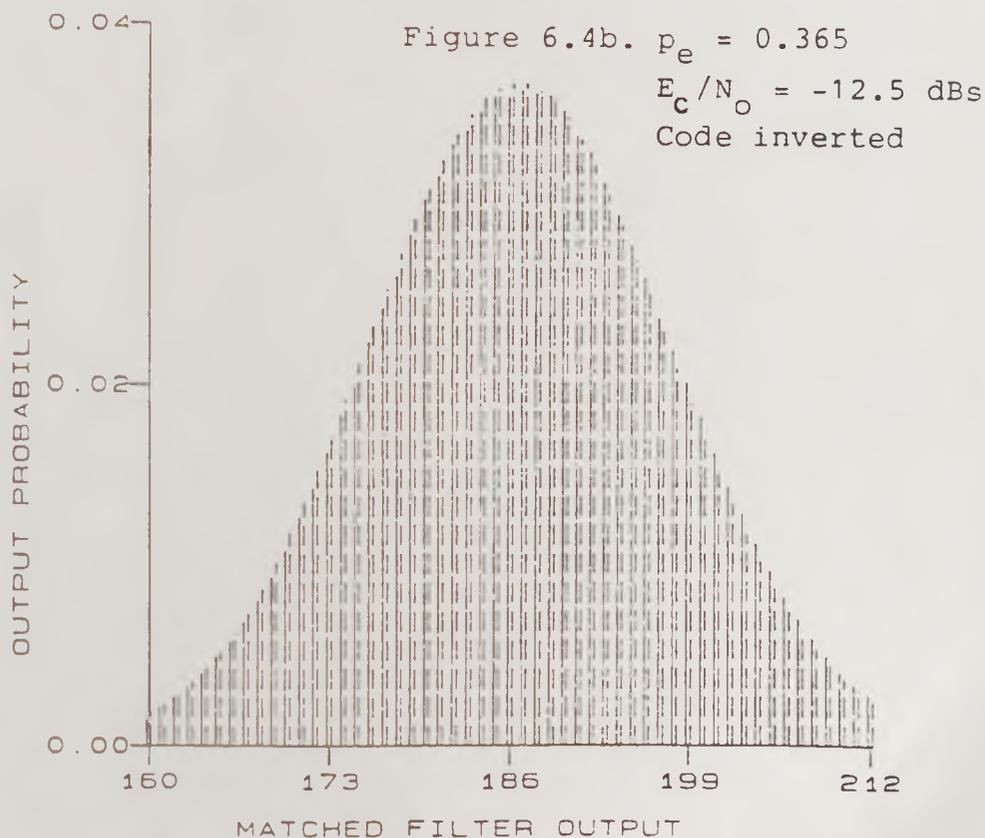
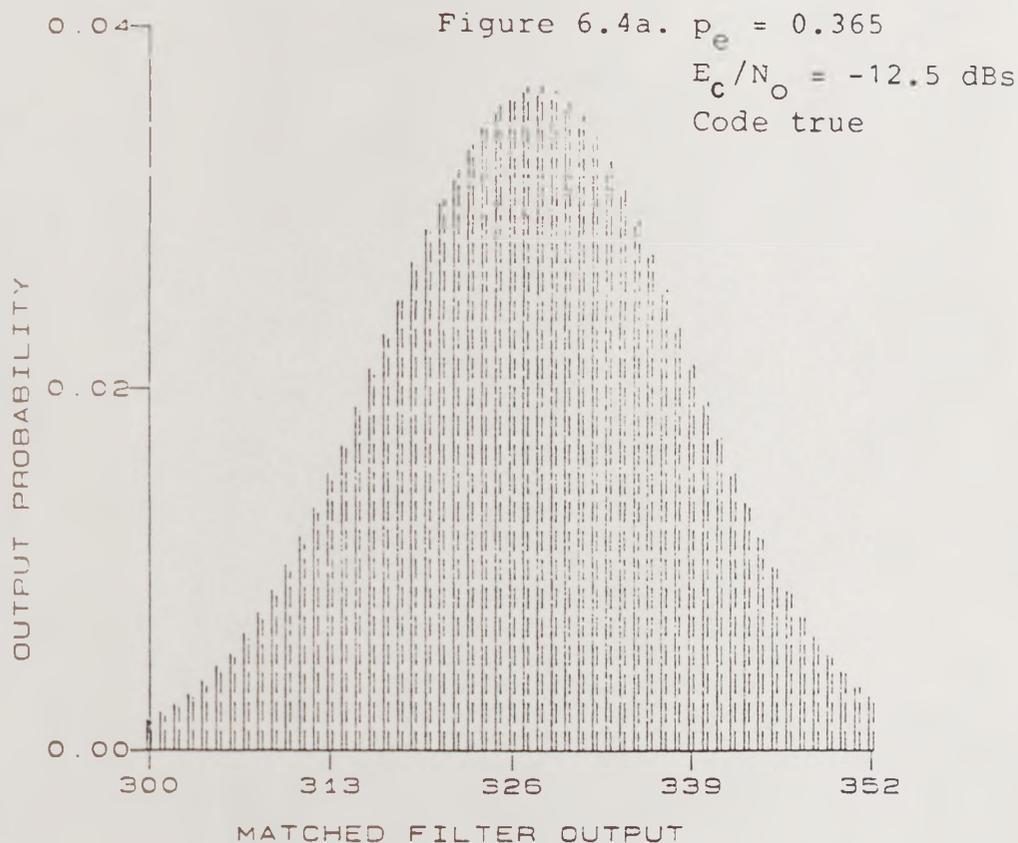


Figure 6-4. DMF output probability distribution.

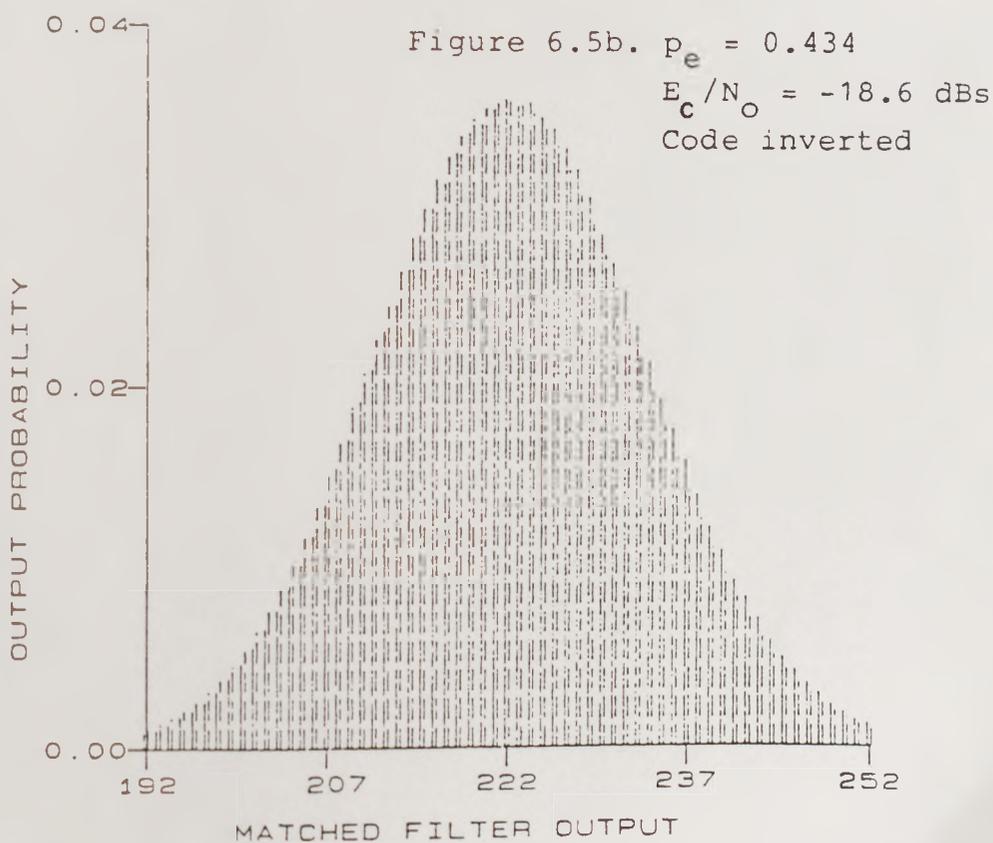
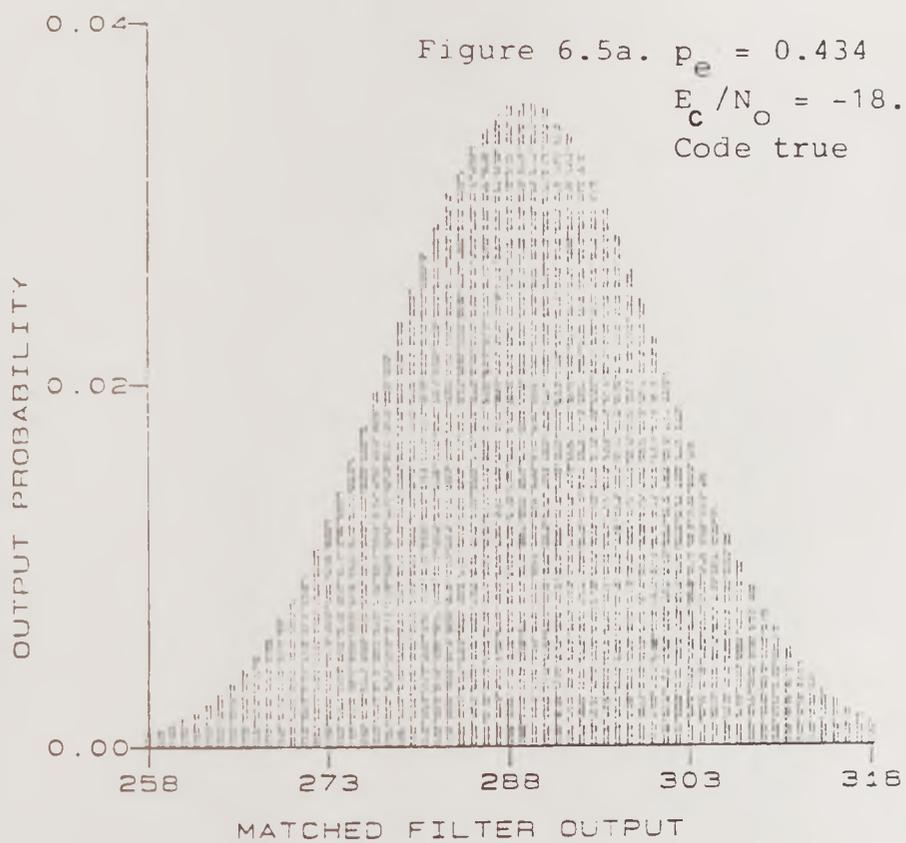


Figure 6-5. DMF output probability distribution.

Figure 6.6a. Code out-of-phase distribution

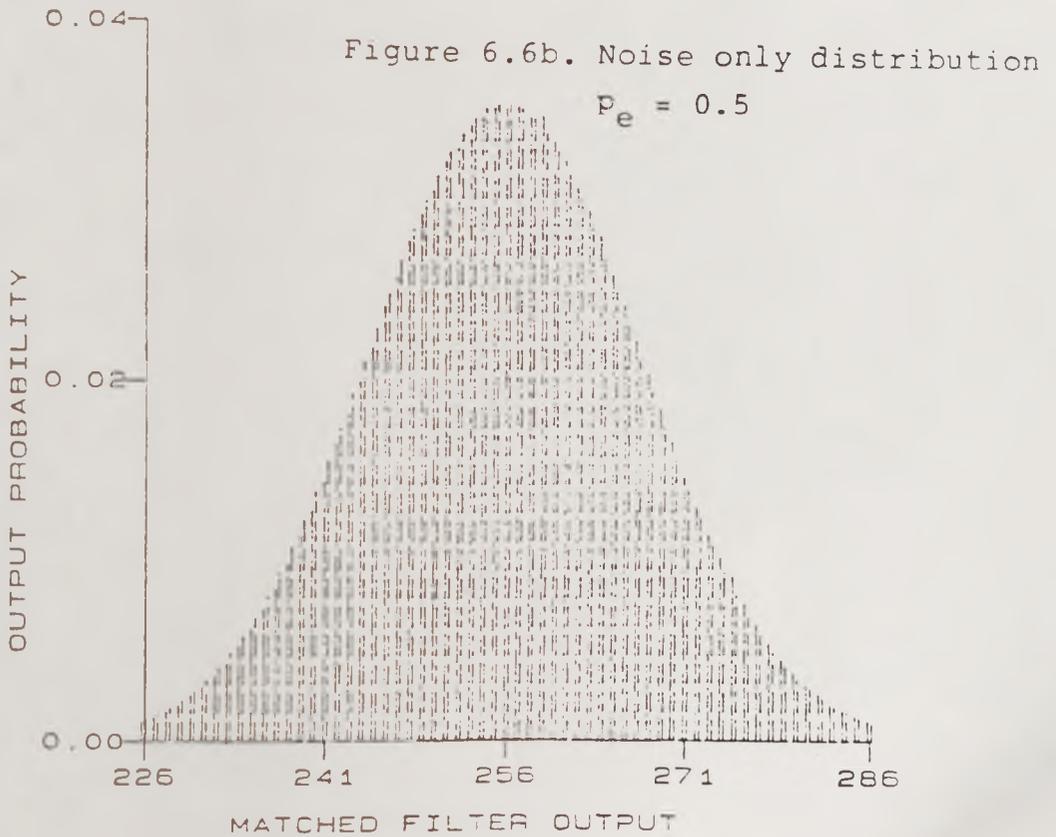
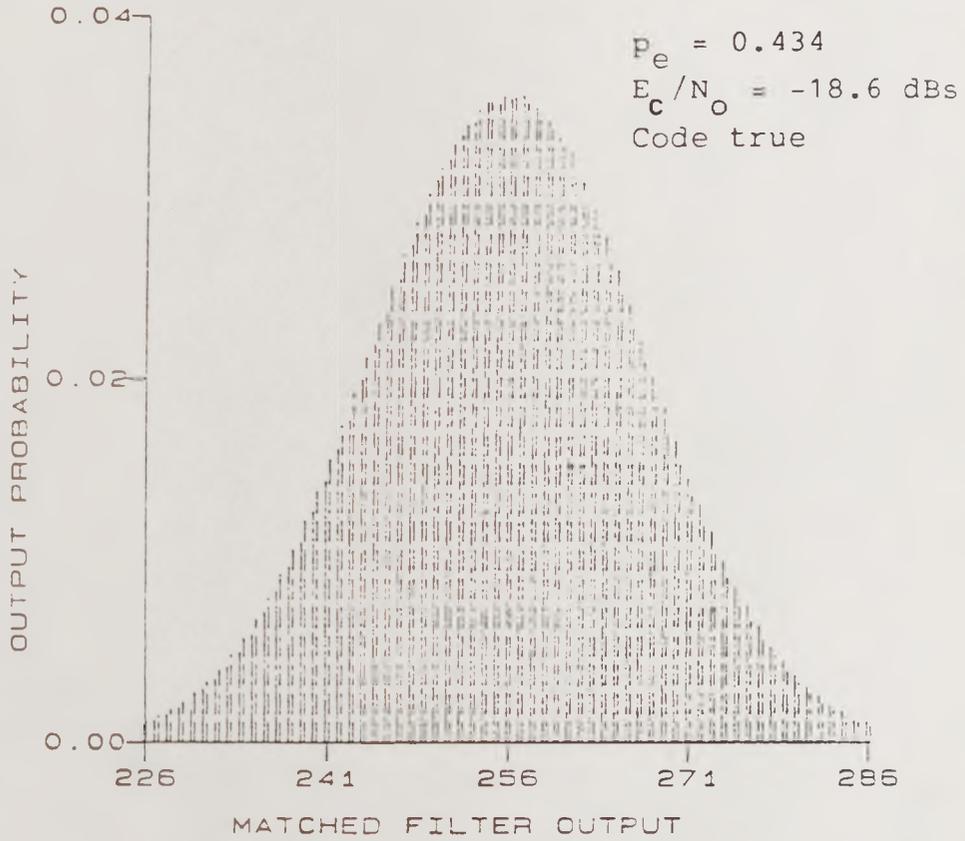


Figure 6-6. DMF output probability distribution.

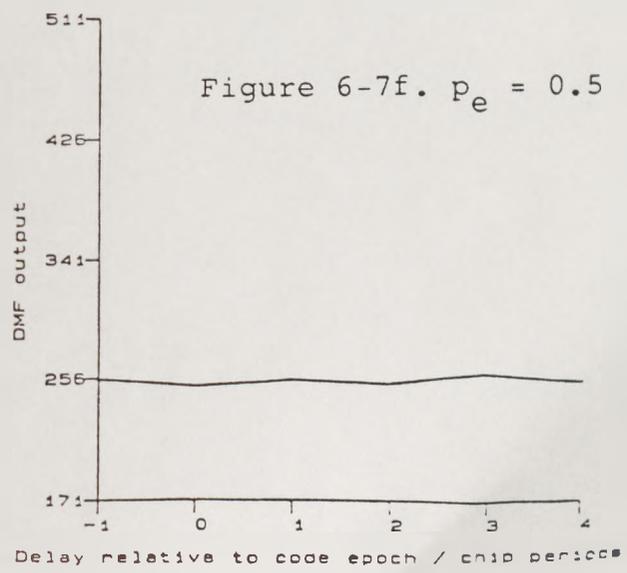
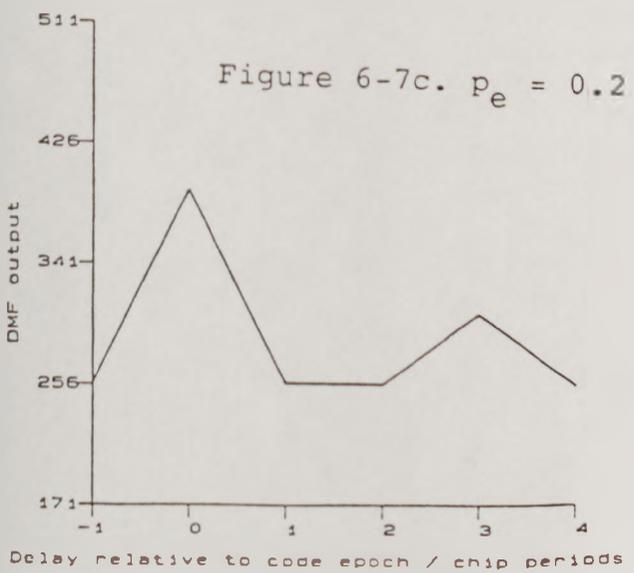
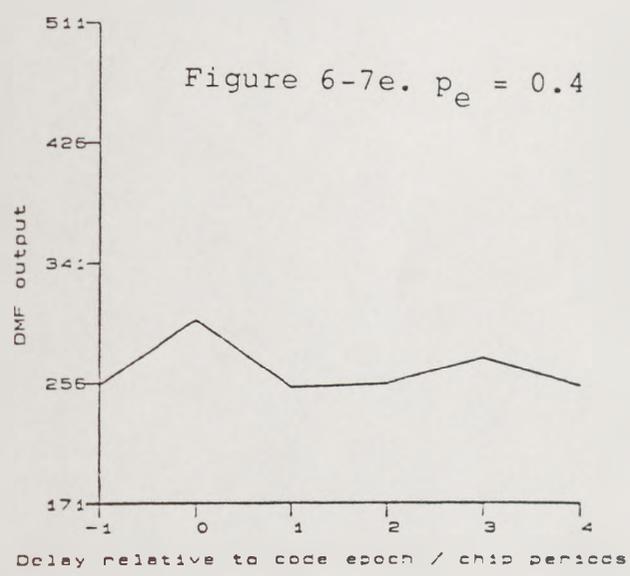
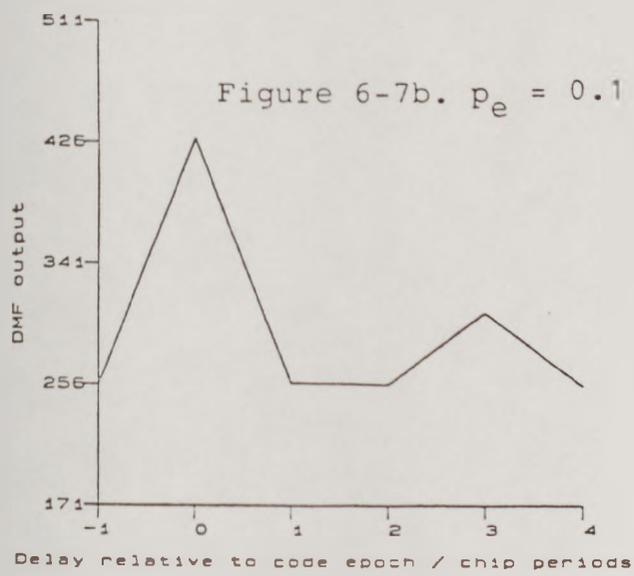
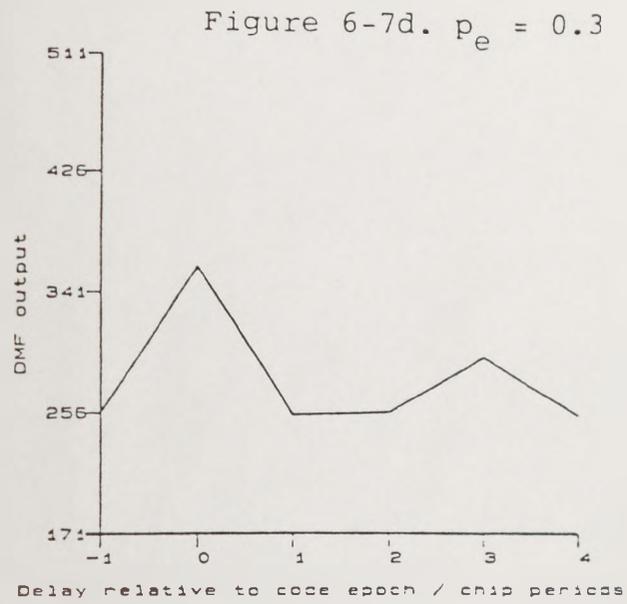
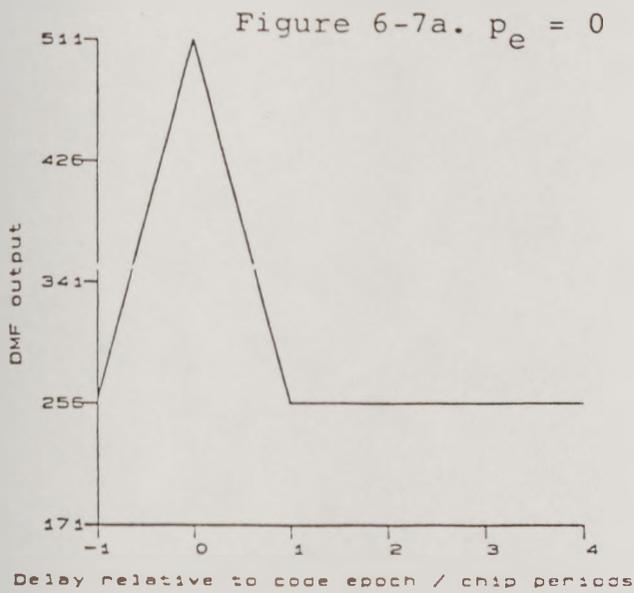


Figure 6-7. DMF mean output against delay with delayed path 6 dB below main path.
Delayed path not inverted.

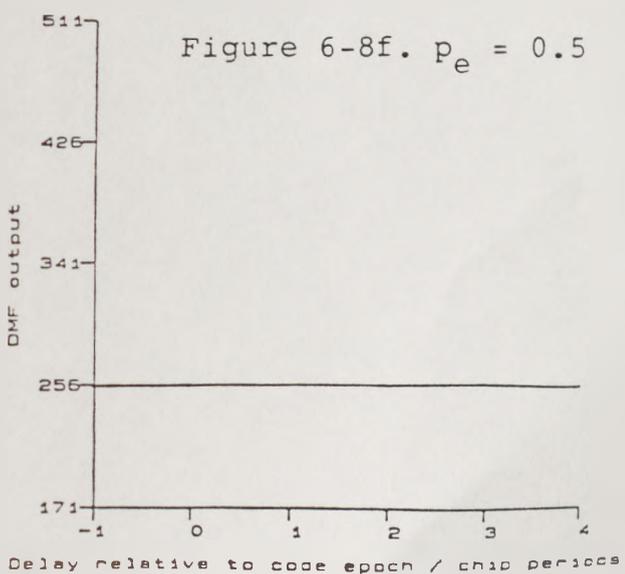
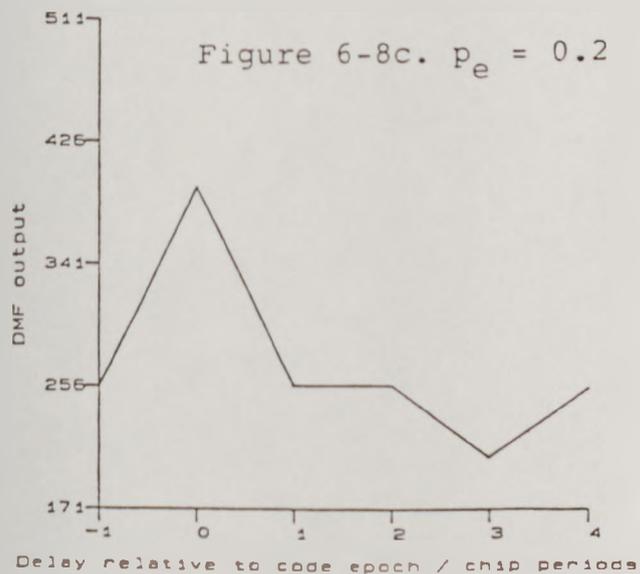
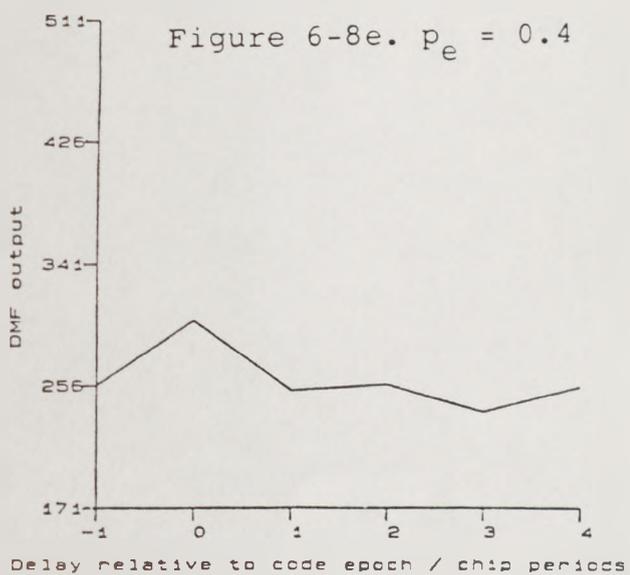
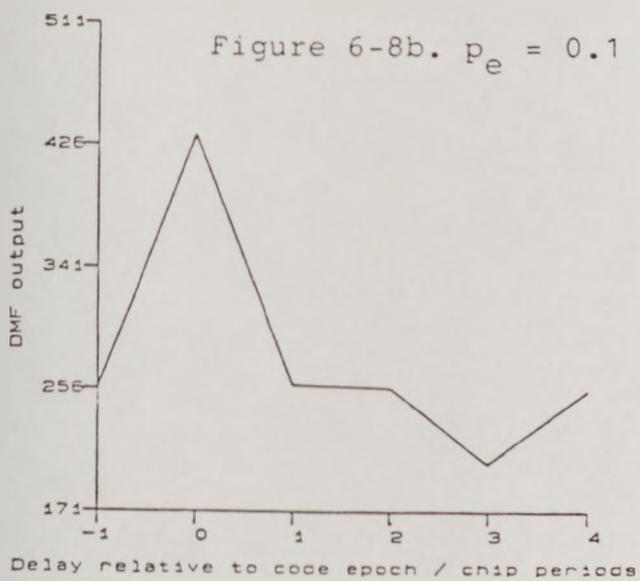
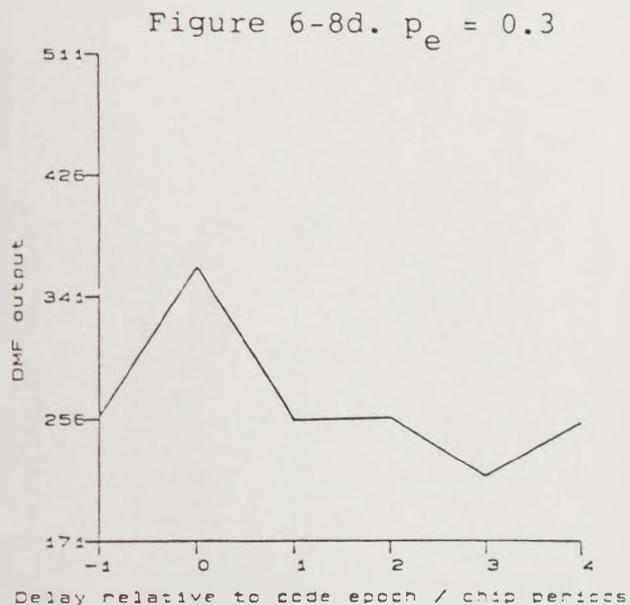
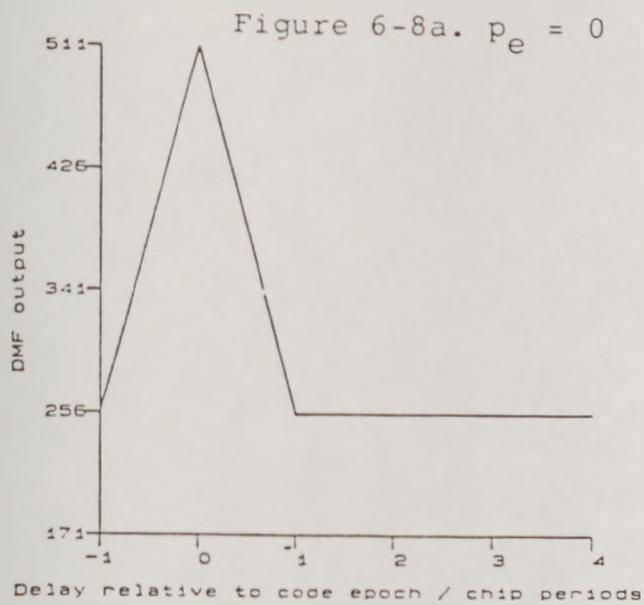
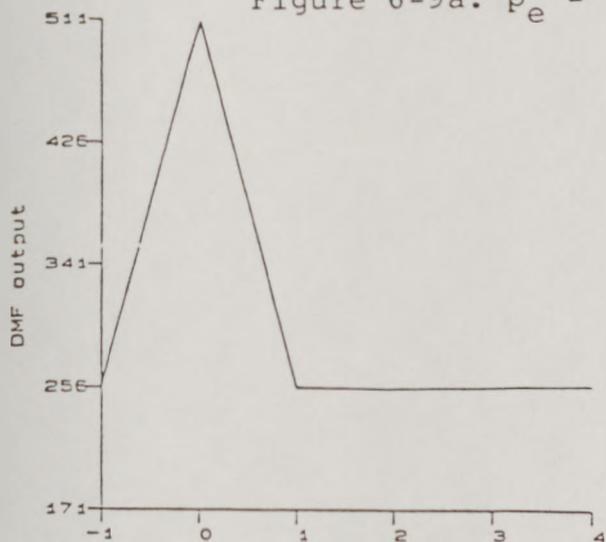
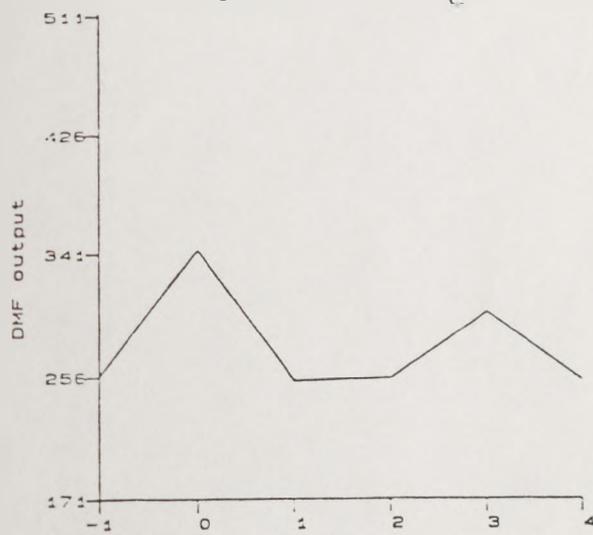


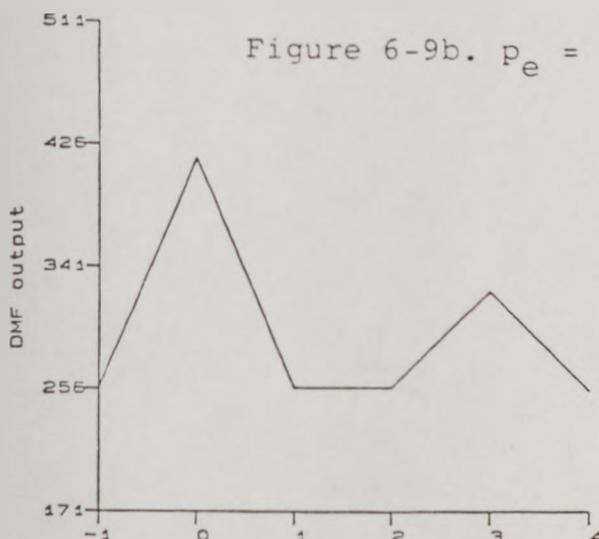
Figure 6-8. DMF mean output against delay with delayed path 6 dB below main path. Delayed path inverted.

Figure 6-9a. $p_e = 0$ 

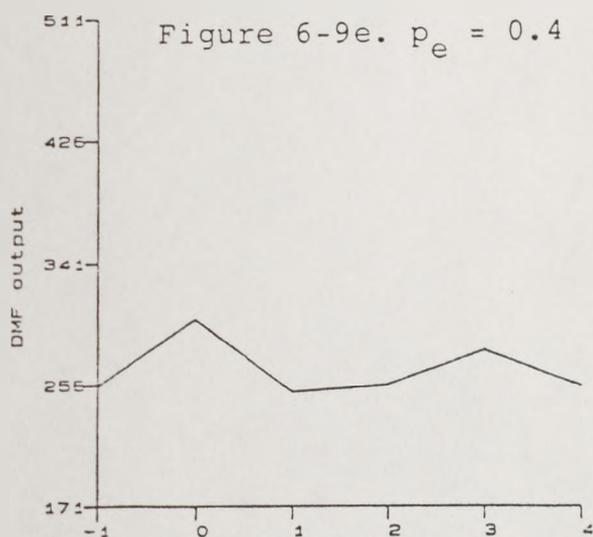
Delay relative to code epoch / chip periods

Figure 6-9d. $p_e = 0.3$ 

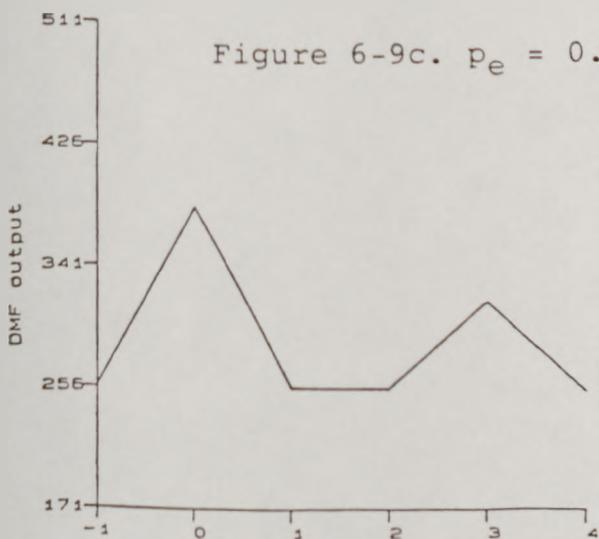
Delay relative to code epoch / chip periods

Figure 6-9b. $p_e = 0.1$ 

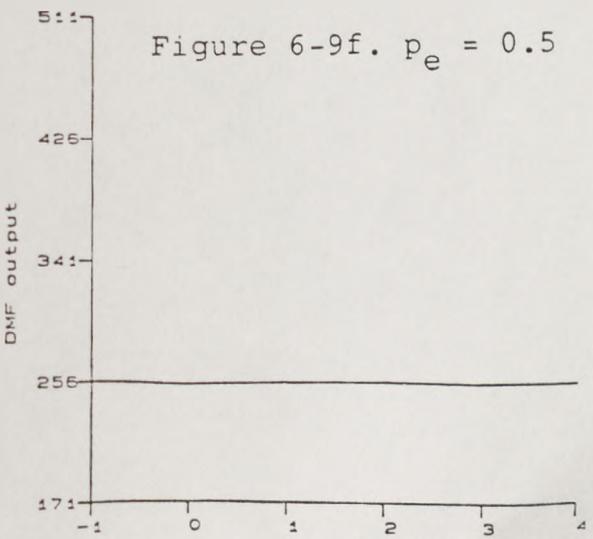
Delay relative to code epoch / chip periods

Figure 6-9e. $p_e = 0.4$ 

Delay relative to code epoch / chip periods

Figure 6-9c. $p_e = 0.2$ 

Delay relative to code epoch / chip periods

Figure 6-9f. $p_e = 0.5$ 

Delay relative to code epoch / chip periods

Figure 6-9. DMF mean output against delay with delayed path
3 dB below main path.
Delayed path not inverted.

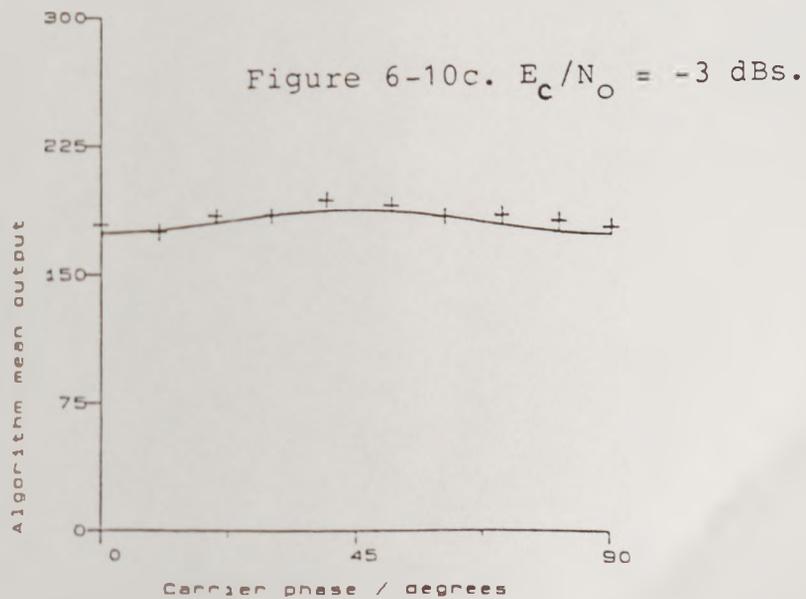
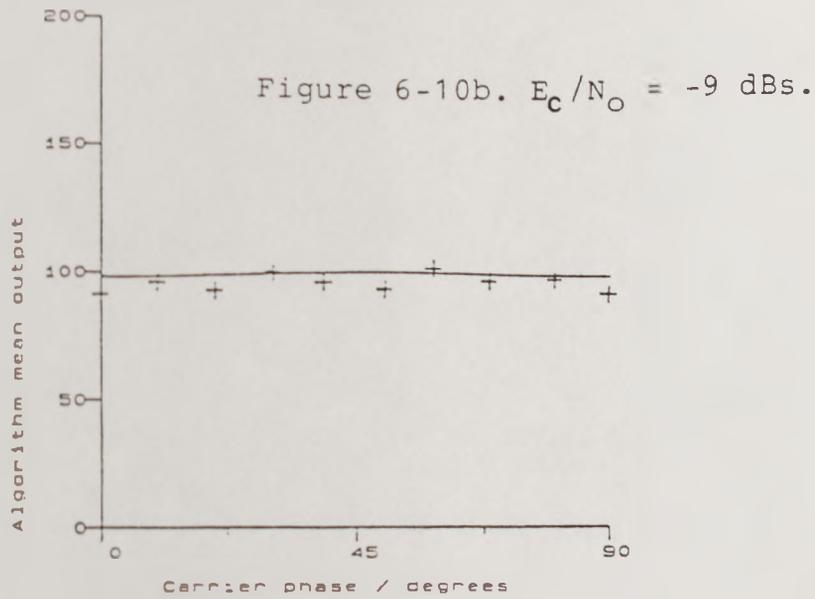
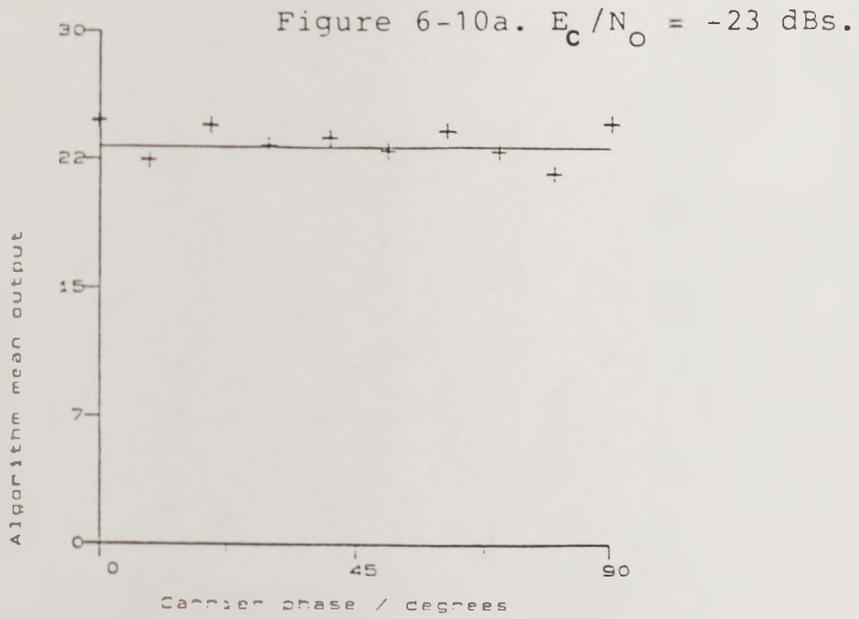


Figure 6-10. Square root algorithm mean output against relative carrier phase.

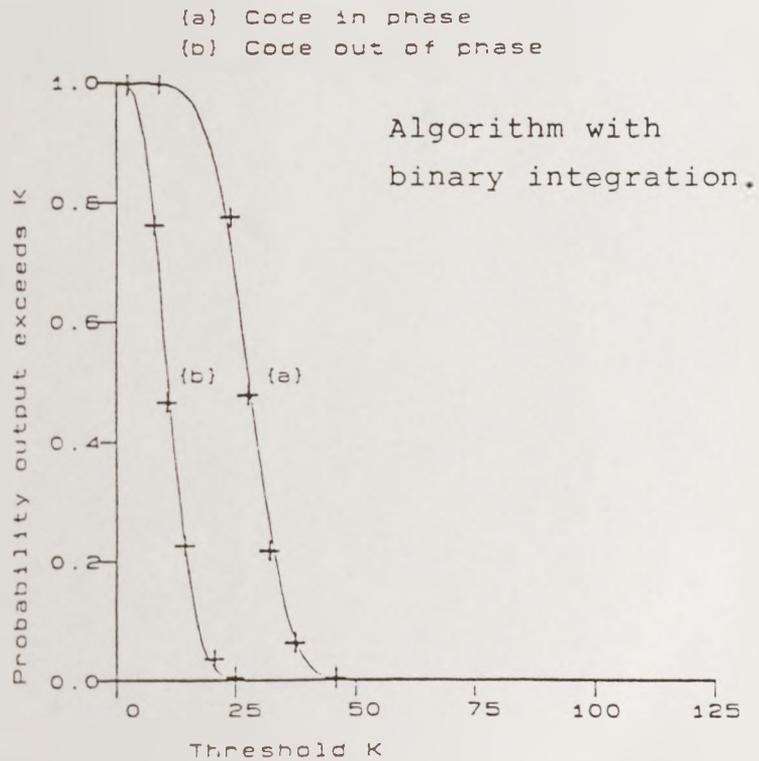
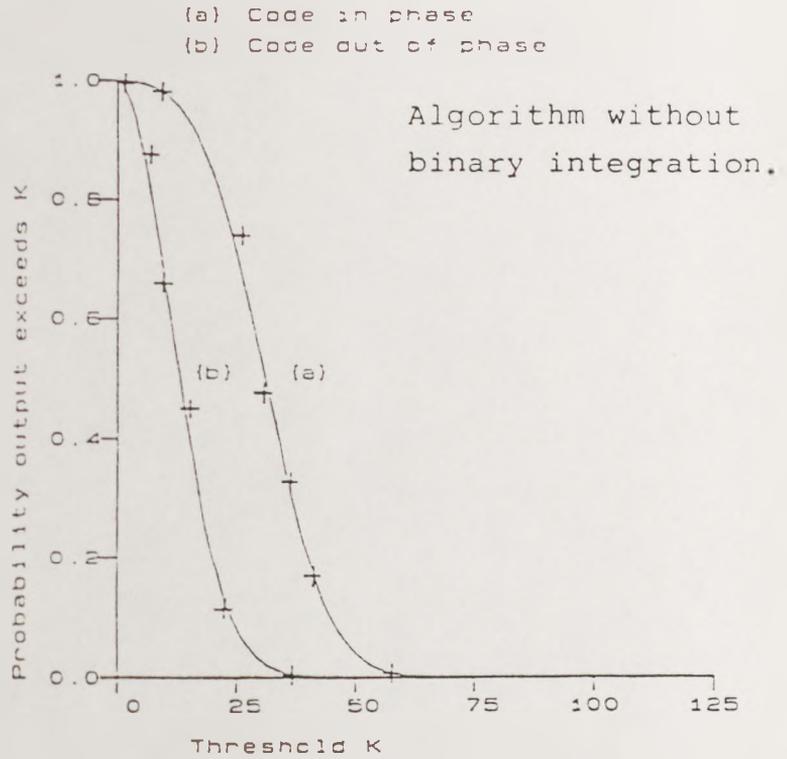


Figure 6-11. Probability algorithm exceeds threshold output.
 $p_e = 0.4437$, $E_c/N_o = -20$ dBs.

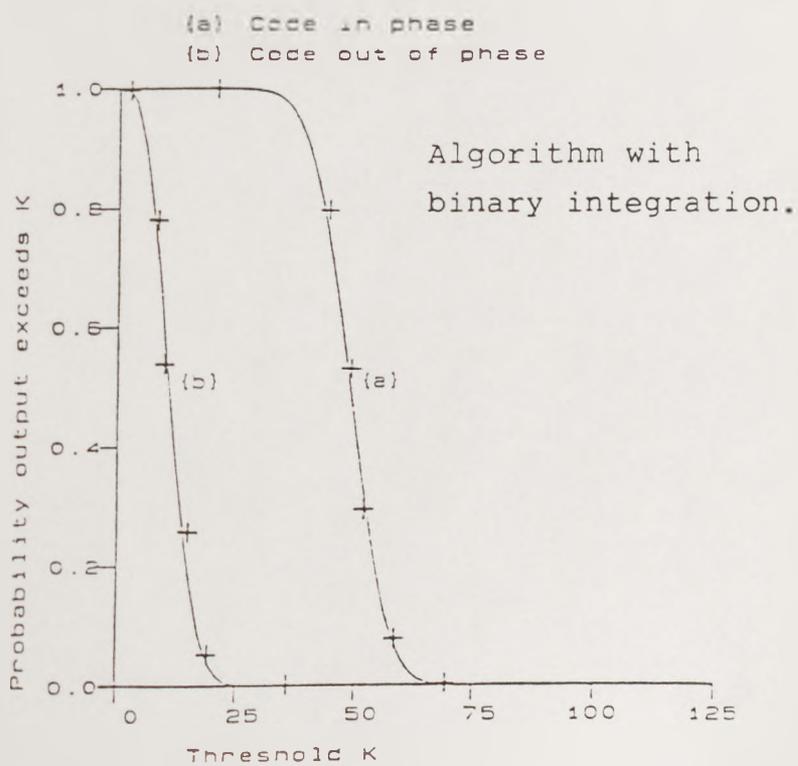
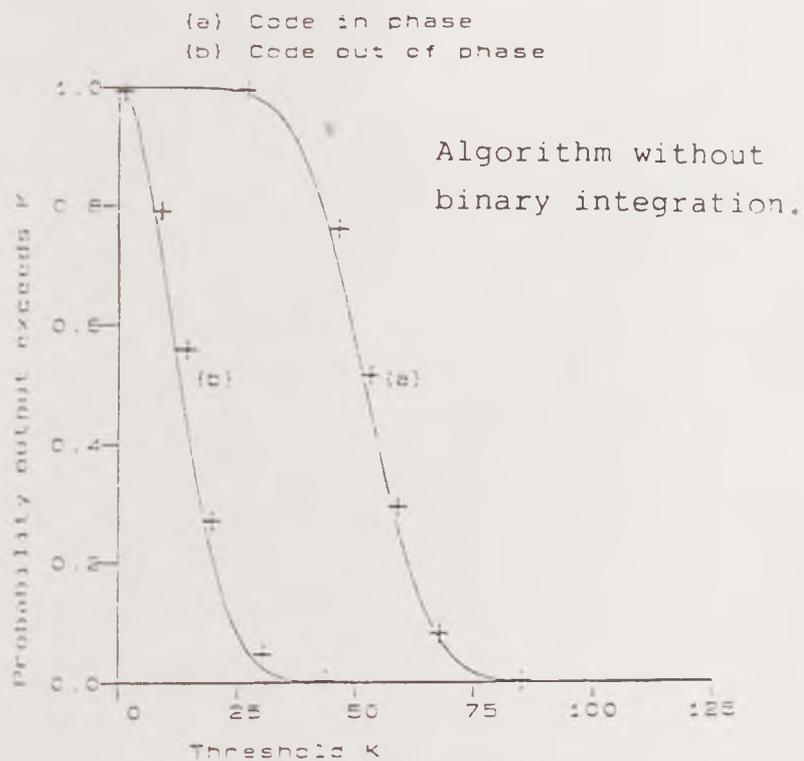


Figure 6-12. Probability algorithm exceeds threshold output.
 $p_e = 0.4007$, $E_c/N_o = -15$ dBs.

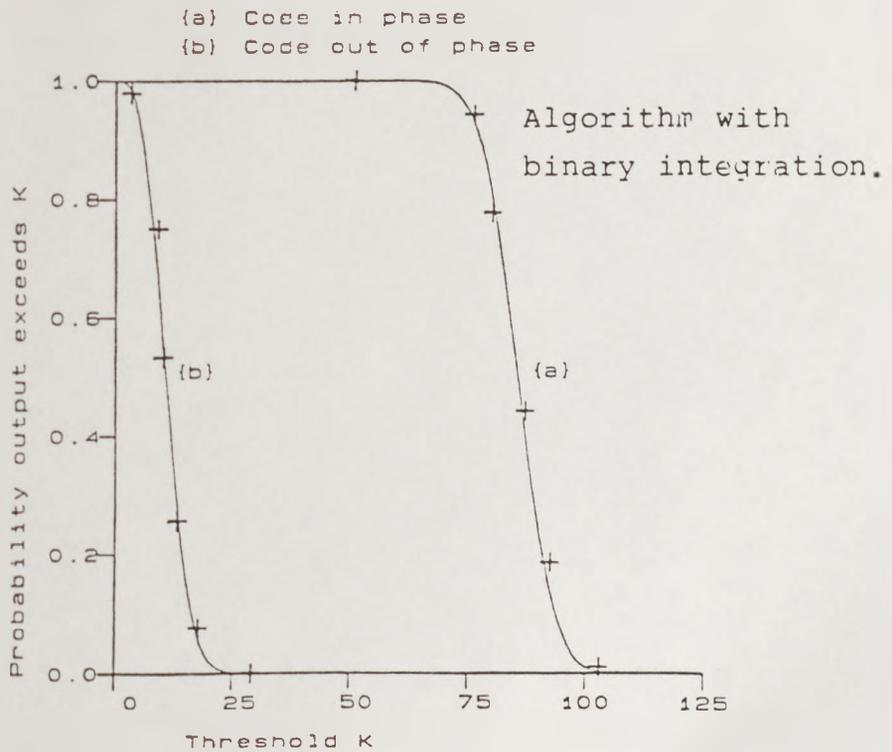
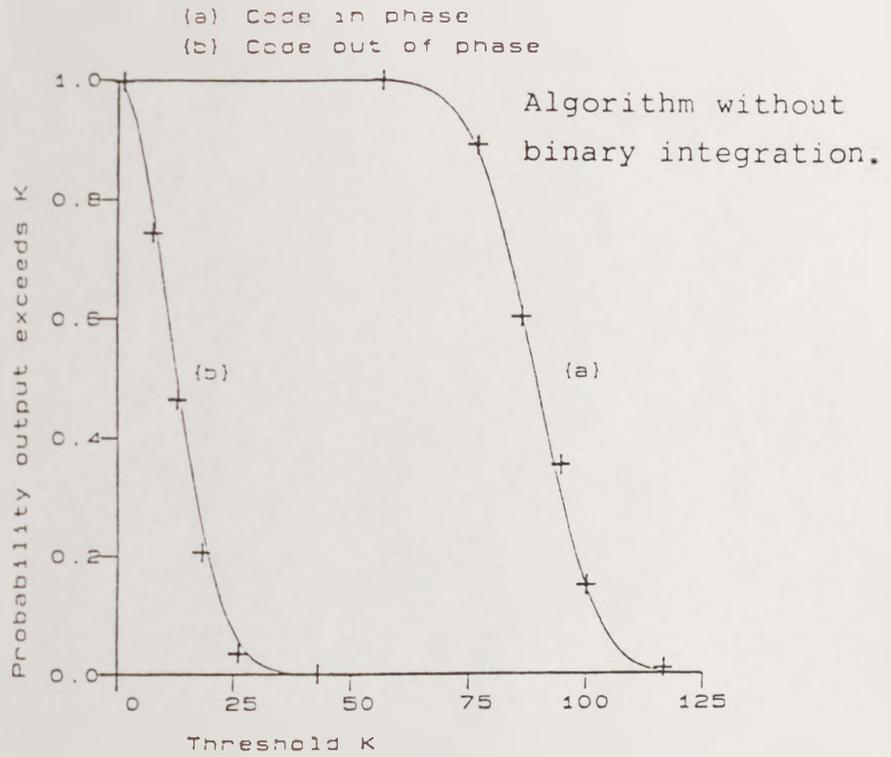


Figure 6-13. Probability algorithm exceeds threshold output.
 $p_e = 0.3274$, $E_c/N_0 = -10$ dBs.

CHAPTER 7

CONCLUSIONS

The work presented in this thesis has examined the operation and suitability of the Digital Matched Filter as a processing element in the Direct Sequence Spread Spectrum receiver. Four areas have been emphasised

- i) Structure and operation of the DMF.
- ii) Integration of the DMF within a DSSS receiver, and the losses associated with practical receiver elements.
- iii) Application of the DMF to receiver synchronization.
- iv) Construction of a DSSS receiver and the acquisition of results in support of i to iii.

The structure and operation of the DMF has been studied in detail. The practical DMF device architecture is designed to exploit simplification in implementation, utilising a large number of simple exclusive-NOR functions in preference to more complex structures necessary for the implementation of correlation operations.

Examination of the DMF output statistics, in particular the output mean and variances, at various chip error rates indicated that the correct choice of code autocorrelation function will maximise the ratio of code in phase to code out of phase DMF output for SIK modulation. For this type of code there is an optimum code in phase decision threshold at the DMF output for SIK modulation, in a statistical sense.

OOK modulation was also considered in the context of

DSSS systems. The approximate threshold for optimum detection was evaluated, and it was shown that this type of modulation scheme has the advantage of removing the need for carrier phase coherence at the receiver. However there is ambiguity over the optimum decision threshold for an unknown input snr.

The loss incurred by hard limiting was examined by evaluating the mean and variances of the quantizer output statistics. The results closely agreed with those presented by other authors, indicating a loss in processing gain of about 2 dB for the hard limiter. The technique used to determine this value is readily extended to the cases of non-white input noise, interfering signals, and distortion, by evaluation of the probability of occupancy for each quantizer interval. The quantizer output mean and variances can be derived from these figures.

In considering the DMF within the DSSS receiver a complex baseband receiver structure was analysed in detail. This sub-system converts the received data code at RF to baseband, and provides information for synchronization, tracking and RF recovery. The receiver may then perform optimum detection for a given input snr. To recover carrier phase information a dual channel structure is necessary at baseband, with carrier in phase and quadrature phase channels.

Separate consideration was given to each system element and their effect on the overall performance of the complex baseband receiver was discussed. Provided care is taken in loading the mixing stages the process of down conversion can be assumed to be ideal. The choice of lowpass filter response is, however, a considerably more complex problem, and is based on parameters such as stopband attenuation, step response, and efficiency relative to an ideal chip matched filter. For the 5 filter types examined it was not possible to obtain zero correlation of lowpass noise and signal samples in a given

filter bandwidth. It is possible that equalization of the filter output could be used as a measure against the consequent ISI, but time limitations restricted further investigations into this possibility.

Single bit conversion was considered, since this made optimum use of the available hardware resources and, with a single sample per chip, allowed the minimum length of DMF to be used for a given code length.

It was shown that a simple but important relationship in the ratio of RF frequency to sampling frequency exists. If this ratio is an integer then the effects of the lowpass filter risetime and relative carrier phase can be evaluated for each relative sampling instant. Further once RF synchronization is achieved the correct chip sampling frequency can easily be derived by an integer frequency division.

Two algorithms for combining the DMF outputs were considered. Simply summing the DMF outputs produced an algorithm output which varied in amplitude as the carrier phase was varied relative to the phase of the received RF signal. Squaring each DMF output, followed by summing, and finally taking the square root produced an algorithm output which was considerably less dependent on the relative carrier phase. This property is required during synchronization, where RF lock may not have been achieved. For each algorithm the output was approximately proportional to the input snr, a property which could be exploited in the measurement of the input snr. This information is required in systems utilising power control, a requirement of DSSS systems proposed for use in the LMR environment.

One of the most important areas to be considered in any Spread Spectrum system is receiver synchronization, that is code synchronization acquisition, tracking, and RF recovery. In these areas it was shown that the DMF based

DSSS receiver has significant advantages over techniques previously described in the literature. Published techniques were reviewed, and compared to simple digital implementations. Since the DMF produces a new correlation output for each input sample it can perform the same task as an array of m binary integrators, but is an efficient integrated hardware solution. In comparison the sliding correlator is the minimum hardware solution but is less efficient in terms of average synchronization acquisition time.

It was shown that DMF structures can be used to perform code tracking by delay lock and Tau dither loops but these are not efficient solutions, requiring additional DMFs to form the references.

Analogue RF recovery loops were shown to have a serious disadvantage if used incorrectly in DSSS systems. If a DSSS system is operated in an overlay mode, which would be likely in a LMR environment, false carrier lock is possible if received signals have slightly different centre frequencies, the loops tending to lock onto the strongest signal. This arises since the loops do not have the benefit of the processing gain of the DSSS DMF.

Investigation of code synchronization acquisition using digital techniques identified the need for a threshold detection scheme to initiate the synchronization process. From earlier discussions this proved that the DSSS receiver can never have the full benefit of the available processing gain until code synchronization and hence optimum chip sampling, has been achieved. The statistics of threshold crossing were examined for several input snrs and it was found that a high false alarm rate can only be avoided if each code phase can be separately identified, the accumulation of all out of phase cases as an alternative to the in phase case providing an excessive weighting against the probability of detecting the in phase case. Unique code phases can be identified by sampling the

DMF outputs at the symbol rate. Applying a binary integration algorithm to a succession of such samples produced a marked improvement in the ratio of in phase to out of phase threshold crossing probability. This technique suffers a slight penalty in response time.

To increase hardware efficiency in the code synchronization acquisition area a single threshold detector was investigated. In common with RASE and RARASE synchronization techniques it was found that this approach could require many 'retries' before synchronization is achieved, and at low input snrs this would lead to an unacceptably high mean time to synchronization. This was attributed to the large number of out of phase positions within a code period and the resulting likelihood of a search initiation at a false synchronization position.

This problem is considerably reduced by a modest increase in the number of threshold detectors, each operating independently over a subrange of the code period. This reduced the number of out of phase locations included in the search and tabulated results of an increase from 1 to 4 detectors indicated the enormous benefit of this simple enhancement, particularly at low input snrs.

It was shown that RF recovery is possible with the complex baseband receiver, utilising information obtained from the DMFs. At low snrs the carrier phase is reasonably represented by the arctangent of the quadrature channel DMF output divided by the in phase channel DMF output. At higher snrs this output becomes biased towards 45 degrees, but the requirement of accurate RF recovery can be partially relaxed at higher input snrs, since optimum chip sampling becomes less critical.

To consolidate the techniques described a DMF receiver was proposed and analysed, in particular consideration was given to its application to the LMR environment. It was shown that such a scheme could achieve code synchronization

and RF recovery by utilising the processing gain available within the DMF, but a minimum acceptable snr must be present at the receiver input. In terms of code synchronization acquisition time the proposed scheme will outperform a sliding correlator of similar length. Acquisition time is sufficiently short to allow the receiver to be used in a burst transmission mode. In LMR schemes Doppler effects and local oscillator stability must be considered.

In the proposed scheme the Base transmits to all Mobiles in an overlay mode, using the same centre frequency. At the Mobiles an analogue RF recovery scheme can be used, this has a wider tuning range and false lock is prevented since all signals have the same carrier centre frequency. Further each Mobile receives all signals but these can be discriminated by utilising the processing gain available within the DSSS receiver. The independent movement of each Mobile results in a difference in absolute carrier frequency at each Mobile receiver, due to the effects of Doppler. If the Mobile transmits to the Base at the received centre frequency then the Base receives the signals overlaid, but each may have a unique centre frequency, shifted by twice the received Doppler shift, assuming reciprocity. This precludes the use of an analogue RF recovery scheme at the Base receivers, but phase information can be obtained from the Base DMFs to allow the accurate recovery of the desired carrier, since the Doppler shifts are very small in comparison to the offsets produced by local oscillator instability. In addition the received signal level can be estimated from the DMF outputs allowing power control to be implemented by the Mobiles, reducing the effects of the near-far problem.

The analysis indicated that such a system could work in the LMR environment when considering the effects of Doppler, and DSSS systems have been advocated on the basis of the inherent frequency diversity of the wideband signals. This will reduce the effects of fading, resulting

from multipath signals.

The apparent lack of published information on practical implementations of DSSS systems led to the construction of a DSSS transmitter/receiver pair. The purpose was to investigate the problem areas such schemes might pose in practical terms, and to experimentally verify some of the theoretical ideas presented earlier in the thesis.

The first major problem to be solved was that of cascading the available 64 bit correlators to form a 512 bit correlator. Considerable extra hardware was required and pipelining techniques applied to compensate for propagation delays experienced by signals in the adder sections of the circuits.

The combination of high speed logic devices and sensitive RF sections led to problems of EMI pickup, a direct result of constructing the receiver in a modular fashion to allow flexibility in system configuration. Such problems would be solved in commercial equipment by adopting suitable layout procedures. Construction and operation of the DSSS receiver identified areas requiring careful design, which would have been difficult if not impossible to identify by simulation techniques alone.

Results were obtained from the practical receiver, and established the bounds of performance to be expected for such a system. The noise generator used for experiments was validated in both a spectral and a statistical sense, and was shown to have an amplitude probability density function close to Gaussian. Measurement of error rates indicated that the practical system incurred a loss in processing gain of about 2 dBs due to hard limiting, as predicted. Further the DMF output distribution was validated for a noisy signal, and shown to be a near Binomial distribution.

A multipath signal was simulated and it was shown that

a weaker delayed signal appeared as a peak in the DMF output provided noise was present in the signal. In the absence of noise the amplitude of the delayed signal never exceeds that of the primary signal, and thus cannot influence the hard limiter output. Under these conditions the strongest path is the only signal to appear in the DMF output.

Evaluation of the square root algorithm output showed good agreement with the curves derived in chapter 3. There was little variation in output with carrier phase, and this was consistent for three values of input snr. The combined DMF outputs can thus be used to measure the approximate input snr, an essential feature in any interference limited system requiring power control.

One of the most important areas covered in this work is that of rapid synchronization. Experimental validation of the probability of threshold crossing at the algorithm output showed excellent agreement with the theoretical values derived in chapter 4, both for the algorithm alone, and for the algorithm with 4 bit binary integration.

During the course of this work there has been considerable debate in the literature regarding the efficiency of DSSS schemes in terms of the number of users per unit bandwidth and complexity of implementation. As a result interest in CDMA for the LMR environment has declined, and favour turned towards schemes employing Time Division Multiple Access (TDMA), utilising channel equalization to reduce the effects of multipath distortion. However DSSS schemes presently find widespread use in satellite navigation, and in secure military communication applications. In these areas rapid synchronization acquisition is required. It is hoped that the results presented in this thesis will find application in these areas.

It is possible that as further radio spectrum in the

GHz region is released for LMR usage interest in DSSS may be revived. There is considerable scope for further work on the proposed DSSS receiver, particularly in the area of tracking and control functions. With the addition of this, and a UHF radio section, the practical receiver could be used to fully investigate the practical performance of DSSS in the LMR environment.