

AN ABSTRACT OF THE THESIS OF

Timothy F. Myers for the degree of Master of Science in Electrical and Computer Engineering presented on April 29, 1992.

Title: Proposed Implementation of a Near-Far Resistant Multiuser Detector without Matrix Inversion using Delta-Sigma Modulation

Redacted for Privacy

Abstract Approved: _____
Dr. Mario E. Magaña

A new algorithm is proposed which provides a sub-optimum near-far resistant pattern for correlation with a known signal in a spread-spectrum multiple access environment with additive white gaussian noise (AWGN). Only the patterns and respective delays of the $K-1$ interfering users are required. The technique does not require the inversion of a cross-correlation matrix. The technique can be easily extended to as many users as desired using a simple recursion equation. The computational complexity is $O(K^2)$ for each user to be decoded. It is shown that this method provides the same results as the "one-shot" method proposed by Verdu and Lupas.

Also shown is a new array architecture for implementing this new solution using delta-sigma modulation and a correlator for non-binary patterns that takes advantage of the digitized $\Delta\Sigma$ signals. Simulation results are presented which show the algorithm and correlator to be implementable in VLSI technology. This approach allows processing of the received signal in real-time with a delay of $O(K)$ bit periods per user. A modification of the algorithm is examined which allows further reduction of complexity at the expense of reduced performance.

**©Copyright by Timothy F. Myers
April 29, 1992**

All Rights Reserved

**Proposed Implementation of a Near-Far Resistant Multiuser Detector without
Matrix Inversion using Delta-Sigma Modulation**

**by
Timothy F. Myers**

**A THESIS
submitted to
Oregon State University**

**in partial fulfillment of
the requirements for the
degree of
Master of Science**

**Completed April 29, 1992
Commencement June 1992**

APPROVED:

Redacted for Privacy

(

Assistant Professor of Electrical and Computer Engineering in charge of major

Redacted for Privacy

Head of department of Electrical and Computer Engineering

Redacted for Privacy

Dean of Graduate School

Date thesis is presented: April 29, 1992.

Typed by Tim Myers for Timothy F. Myers.

TABLE OF CONTENTS

	<u>Page</u>
1. INTRODUCTION	1
Definition of Terms	4
Correlation Receiver Theory	5
Review of Asymptotic Efficiency.....	6
Decorrelating Detector Review.....	8
2. MATHEMATICAL BASIS OF PROPOSED SOLUTION	11
Extension for Three or More Users.....	15
Reduction of Computational Complexity	16
Simulation Results	18
3. IMPLEMENTATION OF PROPOSED SOLUTION	20
Simulation Results	23
Analysis of Error Contribution due to A/D Quantization	27
Analysis of Error due to Delay Resolution	31
4. DELTA-SIGMA CORRELATOR	34
Introduction	34
Block Diagram of System	36
Timing Used in System.....	36
Modulators Examined.....	38
2ndOrder Noise SNR Predictions	40
Simulations.....	41
SNR at Input to Detector	41
N x 1 vs. 1 x 1 Multiplication.....	49

Review of Musts and Wants	53
SUMMARY	55
BIBLIOGRAPHY	57
APPENDIX A.....	59
SECTION 1 Introduction	59
One-Shot Decorrelating Theory of Operation	60
Block Diagram of Systolic One-Shot.....	61
Ad-Hoc Approach	63
SECTION 2. Systolic Synthesis of One-Shot	65
Correlation Matrix.....	65
Correlation Matrix Inversion	68
Correlators	69
Appendix Summary.....	71

LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
Figure 1.1 - Block diagram of typical correlator and detector	5
Figure 1.2 - Block diagram of orthonormal basis correlator and detector	6
Figure 1.3 - Block diagram of one-shot decorrelating detector	7
Figure 2.1 - Illustration of one shot break-up of interfering user	13
Figure 2.2 - Three user case bit interval	15
Figure 2.3 - Normalized output from standard and modified GS Correlators..	19
Figure 3.1 - Block diagram of one-shot correlator and detector.....	21
Figure 3.2 - Block diagram of GSOSG	21
Figure 3.3 - Shaded module block diagram.....	22
Figure 3.4 - Non-shaded module block diagram	22
Figure 3.5 - Modified GS block diagram	23
Figure 3.6 - Simulation results of entire system for three users	26
Figure 3.7 - A/D Resolution required for three user case	29
Figure 3.8 - A/D resolution required for five user case	29
Figure 3.9 - A/D resolution required for ten user case.....	30
Figure 3.10 - A/D resolution required for twenty user case.....	30
Figure 3.11 - Normalized output of correlator for one chip interval.....	31
Figure 3.12 - Output of correlator with interfering user as input	32
Figure 3.13 - SNR vs. delay resolution for proposed and standard detectors. ..	33
Figure 4.1 - Block diagram.....	37
Figure 4.2 - Illustration of timing used in system.....	37
Figure 4.3 - Power spectrum of noise transfer functions used in project.....	39
Figure 4.4 - Closer view of power spectrum in range of 0 to $p/4$	39

Figure 4.5 - Simulation results of SNR for modulators	42
Figure 4.6 - Graph of SNR at detector Input.....	43
Figure 4.7 - Proj_16 modulator output	45
Figure 4.8 - Output of multiplier	45
Figure 4.9 - Frequency spectrum with sine wave input	46
Figure 4.10 - Frequency spectrum of message before modulator	47
Figure 4.11. Frequency spectrum of message after the modulator	47
Figure 4.12 - Frequency spectrum for sine wave after modulator and /8 decimator	48
Figure 4.13 - Frequency spectrum for sine wave after modulator and /16 decimator	48
Figure 4.14 - Correlation of bandlimited input with binary PNG pattern.....	50
Figure 4.15 - Correlation of bandlimited input with bandlimited PNG pattern	50
Figure 4.16 - Correlation of input after delta-sigma modulation with binary PNG pattern	51
Figure 4.17 - Correlation of bandlimited input after delta-sigma modulator with band limited PNG pattern	51
Figure 4.18 - Ideal correlation of input signal with PNG pattern	52
Figure 4.19 - Correlation of input signal after delta-modulation with binary PNG pattern using 1x1 multiplier with higher oversampling	52

LIST OF TABLES

<u>Table</u>	<u>Page</u>
Table 2.1 - Simulation results	18
Table 3.1 - Results of simulation	25
Table 4.1 - Musts and wants of a multi-level correlator	34
Table 4.2 - Comparison of SS correlator to delta-sigma modulator.....	35
Table 4.3 - Poles and zeros of error transfer function for modulators.....	38
Table 4.4 - Comparison of calculated to simulated SNR for 2ndOrder modulator	40

LIST OF APPENDIX FIGURES

<u>Figure</u>	<u>Page</u>
A1 - Block diagram for systolic one-shot decorrelating detector.....	62
A2 - Timing for one-shot decorrelating detector	62
A3 - Processor layout diagram for one-shot systolic array with two users.....	64
A4 - Dependency graph for correlation matrix	66
A5 - Processor layout diagram for correlation matrix	67
A6 - Processor layout diagram for correlation matrix with fewer processors	68
A7 - Mapping of search in time correlator.....	71

PROPOSED IMPLEMENTATION OF A NEAR-FAR RESISTANT MULTIUSER DETECTOR WITHOUT MATRIX INVERSION USING DELTA- SIGMA MODULATION

CHAPTER 1 INTRODUCTION

As the need for more communication continues to grow there is a limitation on capacity due to the finite amount of bandwidth available to support all projected users. In order to allow more users to share the same channel, different methods have been devised which achieve this multiplexing. One of the most promising methods is called code-division multiple access (CDMA). This approach allows multiple users to share the same channel simultaneously. This is accomplished via the use of separate codes to modulate the carrier. Generally, the codes used are designed to be as orthogonal as possible to each other. This type of separation allows for a gradual decay of the network as more users try to access it. It does not require the precise timing on the part of the individual users to access the network as does time-division multiplexing. As more users access the CDMA network, the error rate increases causing users to leave the network or to stay on with some loss of performance.

While CDMA works well when all users have about the same power levels, a known problem which limits its use in mobile and other environments is the "near-

far" problem. In this case, a transmitting user which is close to the desired receiver will overwhelm the signal of a user which is transmitting from some distance. Since the codes are not completely orthogonal, there remains some cross-correlation between the transmitting users.

Several approaches have been taken to try to lessen the "near-far" problem. The first is to make the codes as close to orthogonal as possible. This action requires more complicated circuitry to produce the patterns and there is also a limit on the number of patterns that can be generated for a given code sequence length. Another approach, similar to that used in cellular phones, is to allow a base station to limit the transmitting power levels of the users. This method tends to degrade the network to the performance of the weakest user. New cellular technology will attempt to solve the problem by placing the base station a far distance from the transmitters so all received power levels are equivalent. This method is easily achieved by using satellites in orbit as the receivers. Obviously, the number of satellites is limited and this method is very costly to implement.

Recent work has focused on improving the detector in the receiver so that it is "near-far" resistant. Work in this area was previously ignored as it was assumed that the optimum detector for multiple users was close to the detector used for single users. In [2], Verdu showed that an optimum detector for multiple users exists and that its performance is vastly superior to the single user detector in a multi-user environment. Verdu showed that this detector could be implemented using a Viterbi algorithm that was computationally exponential in the number of users and had a variable decoding delay time. Since this revelation, research has been directed at trying to reduce the complexity required to implement a "near-far" detector.

In [3], Poor and Verdu examined single user detectors for multi-user channels and concluded that significant reduction in complexity could be achieved by limiting the detection of each symbol to only the interval in which it is received. They also showed that by accounting only for the subset of interfering users with significant power levels, the amount of computation could be reduced further.

In [4] and [5] linear algorithms are presented to decode the desired signals by processing the output of a bank of filters matched to each user. These approaches require that the power levels of the received signals be known.

In [1], Lupas shows that the optimum linear detector is the inverse of the cross-correlation matrix of the users in the channel. It is further shown that this can be implemented as a bank of filters after the matched filters as the length of the transmitted sequence gets larger. This approach is equivalent to correlating the received signal with a pattern that is orthogonal to the interfering users' sub-space.

Although the approach proposed by Lupas is linear in complexity with the number of users, inverting the cross-correlation matrix becomes quite cumbersome as the number of users increases due to the need to use multiple time intervals. By restricting attention to the "one-shot" or single bit interval as suggested by Verdu, Lupas shows that the "one-shot" is sub-optimum and is upper bounded by the decorrelating detector in performance. This detector still exhibits good near-far resistance. In this approach, the received signal is now correlated with a new pattern. The generation of this new pattern still requires the inversion of a simpler cross-correlation matrix.

This thesis examines current correlation receiver theory and performance measures for CDMA channels in chapter 1. A new method of generating the pattern for the "one-shot" detector is proposed in chapter 2 and it is shown that it is

equivalent to the one proposed by Verdu and Lupas, yet it does not require the inversion of a matrix. Chapter 3 describes the implementation of the new solution which takes advantage of delta-sigma ($\Delta\Sigma$) modulation. Simulation shows that this approach allows VLSI implementation so that the detector can operate in real-time. A study of implementing a non-binary correlator using $\Delta\Sigma$ modulation is examined in chapter 4. Appendix A looks at the blocks necessary to implement the "one-shot" detector of Verdu-Lupas using systolic methods.

Definition of Terms

The message of a code division multiple access (CDMA) direct sequence spread spectrum communication system is characterized as

$$r(t) = S(t, b) + n(t)$$

$$S(t, b) = \sum_{i=-M}^M \sum_{k=1}^K b_k(i) \sqrt{w_k(i)} s_k(t - iT - \tau_k)$$

$$n(t) = AWGN, \text{ psd} = \sigma^2$$

$$N = 2M + 1 = \text{length of message}$$

$$K = \text{number of users in channel}$$

$s_k(t)$ is the normalized signature waveform of user k
and is zero outside of interval $[0, T]$

τ_k = relative delay of user k from start of bit interval

$b_k(i)$ is the i^{th} message bit of user k

$w_k(i)$ is the received energy of user k in the i^{th} time slot

$\tilde{s}(t, b)$ is the normalized receiver input signal with unit energy

psd is power spectral density

Correlation Receiver Theory

There are several methods used to provide coherent detection of signals. This thesis examines the role of a near-far resistant detector used in code-division multiple access direct sequence spread spectrum communication channels. Although matched filters can be used, this thesis focuses on the correlator receiver and thus requires that we have knowledge of the signals under consideration and the delays between the multiple users' bit sequences. The most common form of correlator receiver for multiple users (Figure 1.1) is composed of K active correlators, each containing a pattern matched to that used to send each user's signal. The output of each individual correlator then has a check of its sign at the appropriate time interval to determine the bit level.

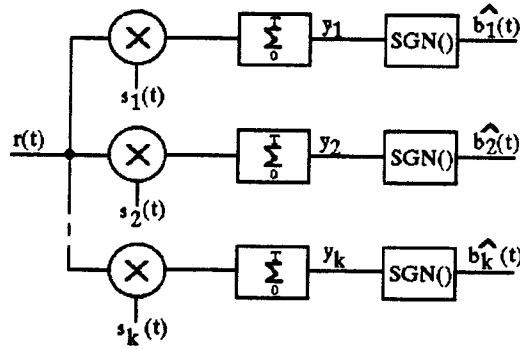


Figure 1.1 - Block diagram of typical correlator and detector

Another approach that has been used is to make use of the fact that each signal can be represented by a set of basis functions ϕ_j represented by

$$s_i(t) = \sum_{j=1}^N s_{ij} \phi_j(t)$$

The i^{th} signal can be detected by using correlators that contain the basis function patterns and weighting the outputs of the correlators by the corresponding coefficients s_{ij} . The basis vectors can be formed by using the Gram-Schmidt procedure. The block diagram for this method would be

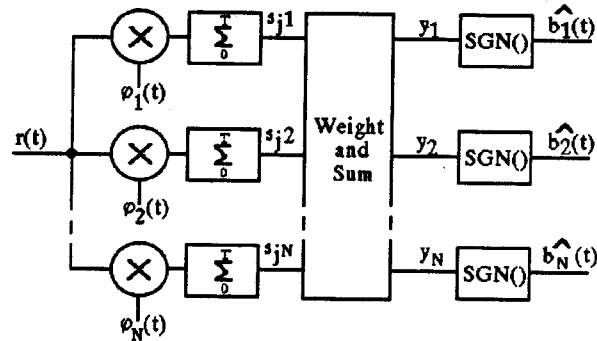


Figure 1.2 - Block diagram of orthonormal basis correlator and detector

A new method, the "one-shot decorrelating detector" was proposed by Verdu and Lupas [1] which provides resistance to the near-far problem. The received signal for this approach (Figure 1.3) is correlated with a set of patterns which are each orthogonal to the other interfering users. This approach increases the bit error rate for AWGN but only marginally in high SNR conditions. The approach by Verdu and Lupas requires that the interfering patterns be broken up into sub-intervals. A matrix containing the cross-correlations of these waveforms is inverted and the orthogonal pattern is created by multiplying the appropriate row of the matrix by the desired user and $m=2K-1$ sub-divided interfering users patterns.

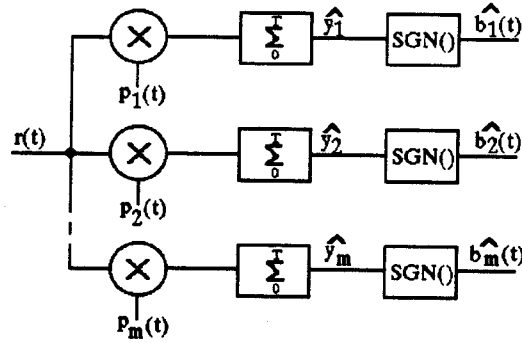


Figure 1.3 - Block diagram of one-shot decorrelating detector

This paper builds on the Verdu-Lupas method by proposing a new method to derive the orthogonal patterns without inverting a matrix. This new approach is then examined along with the Verdu-Lupas method to see how it can be implemented using systolic methods. Also presented in this paper is a correlator circuit which can perform the multi-level correlation process needed for the new detectors. This correlator uses delta-sigma modulation to provide a serial bit stream which can easily be processed using standard VLSI digital logic and analog circuitry.

Review of Asymptotic Efficiency

One of the most important measures of performance in a communication channel is its bit error rate (BER). For channels that are characterized with additive white gaussian noise (AWGN), this BER can be calculated using the "Q function", represented by

$$Q(x) = \int_x^{\infty} \frac{1}{\sqrt{2\pi}} e^{-y^2/2} dy$$

where x is a function of the SNR at the detector. Efficiency as used in [1] is defined as the ratio between the effective SNR and the actual SNR. The effective SNR is

the SNR defined with only AWGN in the channel. The actual SNR includes additional contributions due to the noise from other sources. The efficiency is always non-negative and upper bounded by 1 since the actual SNR is the best possible SNR in the channel. The efficiency definition provides a method by which we can analyze communication systems.

One measure of multiuser system performance uses asymptotic efficiency which was defined by Verdu. This measure is defined as the limit of the efficiency as the background noise level goes to zero. It provides a means to calculate the performance loss due to other active users using the same channel as the user of interest.

For the type of channel examined in this paper, (e.g. binary, antipodal, AWGN) the probability of error for an optimum detector using a correlator pattern matched to the desired signal and followed by a SGN() decision is $Q(\sqrt{w}/\sigma)$. When K multiple users are also in the channel, the kth user's energy (w_k) and the error probability (P_k) are used to create the effective energy (e_k) where

$$P_k(\sigma) = Q(\sqrt{e_k(\sigma)}/\sigma)$$

From [1], "the logarithm of the kth user error probability decays asymptotically with the slope corresponding to a single user with energy $\eta_k w_k$." The asymptotic efficiency for the kth user [12] is then

$$\eta_k = \lim_{\sigma \rightarrow 0} \frac{e_k(\sigma)}{w_k}$$

$$= \sup \left\{ 0 \leq r \leq 1; \frac{P_k(\sigma)}{Q\left(\frac{\sqrt{rw_k}}{\sigma}\right)} < +\infty \right\}$$

In order to have an analytical tool for the near far scenario with multiple users, a measure is defined in [1] that predicts the performance of detectors which operate over all received energies. The "near-far resistance" is defined as the worst case asymptotic efficiency over all possible energies of interfering users. This measure is defined as

$$\bar{\eta}_k = \inf_{\substack{w_j \geq 0 \\ j \neq k}} \eta_k$$

A detector is defined as near-far resistant for user k , if the near far resistance of user k is non-zero.

Decorrelating Detector Review

From [13] the sampled output of a normalized matched filter for the i^{th} bit of the k^{th} user, $i = -M \dots M$ is

$$\begin{aligned} y_k(i) &= \int_{iT+\tau_k}^{iT+T+\tau_k} r(t) s_k(t - iT - \tau_k) dt \\ &= \int_{-\infty}^{\infty} S(t, b) s_k(t - iT - \tau_k) dt + \int_{-\infty}^{\infty} n(t) s_k(t - iT - \tau_k) dt \end{aligned}$$

since the signals are zero outside of $[0, T]$.

The linear detector for bit i of user k is characterized as $v^{k,i} \in L$, where L is the Hilbert space of square-integrable functions. The decision of the detector is given by the polarity of the inner product of $v^{k,i}$ and the vector y of the matched filter outputs, which is equal to

$$\begin{aligned} \sum_{l=-M}^M \sum_{j=1}^K v_j^{k,i}(l) y_j(l) &= \int_{-\infty}^{\infty} \tilde{s}(t, \bar{w} \bar{b}) \tilde{s}(t, \bar{v}^{k,i}) dt + n_{k,i} \\ &= \langle \tilde{s}(t, \bar{w} \bar{b}), \tilde{s}(t, \bar{v}^{k,i}) \rangle + n_{k,i} \end{aligned}$$

$n^{k,i}$ is the noise component at the output of the cascade of the matched filter, sampler and detector. It is a Gaussian zero mean random variable with a variance of

$$\begin{aligned} E[n^{k,i^2}] &= \sum_{k,l \neq j} v_k(l) v_j(i) \int_{-\infty}^{\infty} \sigma^2 s_k(t - lT - \tau_k) s_j(t - iT - \tau_j) dt \\ &= \sigma^2 \|\tilde{s}(t, v^{k,i})\|^2 \end{aligned}$$

The receiver decision on the i^{th} bit of user k is

$$\begin{aligned} \tilde{b}_k(i) &= \text{sgn} \sum_{l=-M}^M \sum_{j=1}^K v^{k,i}(l) y_j(l) \\ &= \text{sgn} \left(\left\langle \tilde{s}(t, \bar{w}\bar{b}), \tilde{s}(t, \bar{v}^{k,i}) \right\rangle + n_{k,i} \right) \end{aligned}$$

By restricting the time interval to just that of the bit we are trying to decode then

$$b_k(i) = \text{sgn} \left(\left\langle \tilde{s}(\bar{w}\bar{b}), \tilde{s}(\bar{v}^{k,i}) \right\rangle + n_{k,i} \right)$$

This restriction results in the "one-shot" senario. The goal now is to generate a pattern $s(v^{k,i})$ which is insensitive to the energies of the interfering users.

CHAPTER 2

MATHEMATICAL BASIS OF PROPOSED SOLUTION

The key point made in Lupas' paper[1] is that rather than correlating the received signal with the same pattern used to send the signal, the matched filter at the receiver is modified such that its response is orthogonal to the known interfering users. This filter and detector were shown to have the property of being optimum in terms of near-far resistance . Due to the mathematical complexity in generating the response of the filter for a large number of users, Lupas investigated the "one-shot" detector proposed by Verdu[3] and found that while it was sub-optimum, its near-far resistance was closely comparable to the optimum. In this case, the basis vectors used to despread the signal are derived by inverting the cross correlation matrix of the interfering patterns and the signal of interest. In order for this method to be stable, i.e. the matrix is invertible, the following conditions must be met:

- a. The signals must be linearly independent of each other in the bit interval under consideration.
- b. The transmission of bits of an interfering user and the desired user must not be synchronous (i.e. arrive at the same time instant).

The first condition can be easily met by selection of the proper spreading sequences. The second requirement occurs if the bit delay time of one or more of the interfering users coincide with the start of the user we are trying to decode. If this action occurs, then the left sub-bit is discarded and only the right sub-bit is used. In practice, this approach would be difficult to implement in hardware since the

matrix inversion circuitry would have to allow for differing numbers of interfering users.

Another method which avoids inverting a matrix but which can also be used to generate the orthogonal basis vector to interfering users is the Gram-Schmidt (GS) procedure. Normally, the GS procedure is used to form orthonormal basis vectors from a known set of original non-orthogonal vectors. Each of the original vectors can be created by linearly combining the basis vectors. Let the original vectors be represented as $\{\bar{v}_1, \bar{v}_2, \bar{v}_3 \dots \bar{v}_k\}$ and the basis vectors as $\{\bar{w}_1, \bar{w}_2, \bar{w}_3 \dots \bar{w}_k\}$. Each original vector is then constructed by:

$$\bar{v}_i = \sum_{j=1}^k \alpha_{ij} \bar{w}_j \quad \text{where } \alpha_{ij} = \langle \bar{w}_j, \bar{v}_i \rangle$$

The GS procedure used to generate the basis vectors uses the following steps:

- a) $\bar{w}_1 = \bar{v}_1 / \|\bar{v}_1\|$ where $\|\cdot\|$ is the Euclidean norm
- b) while $2 \leq i \leq k$

$$\bar{z}_i = \bar{v}_i - \sum_{j=1}^{i-1} \langle \bar{v}_i, \bar{w}_j \rangle \bar{w}_j$$

$$\bar{w}_i = \bar{z}_i / \|\bar{z}_i\|$$

The final set of basis vectors is related to the original vectors by:

$$\begin{bmatrix} \alpha_{11} & 0 & 0 & \dots & 0 \\ \alpha_{21} & \alpha_{22} & 0 & \dots & 0 \\ \alpha_{31} & \alpha_{32} & \alpha_{33} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \alpha_{k1} & \alpha_{k2} & \alpha_{k3} & \dots & \alpha_{kk} \end{bmatrix} \begin{bmatrix} \bar{w}_1 \\ \bar{w}_2 \\ \bar{w}_3 \\ \vdots \\ \bar{w}_k \end{bmatrix} = \begin{bmatrix} \bar{v}_1 \\ \bar{v}_2 \\ \bar{v}_3 \\ \vdots \\ \bar{v}_k \end{bmatrix}$$

It is evident from the above formulas that the last basis vector \bar{w}_k formed by the GS approach is orthogonal to the original vectors (except \bar{v}_k) due to the coefficients of $\alpha_{ik}=0$ except for α_{kk} .

In order to use this method to find the basis vector which is orthogonal to the interfering users, v_k (e.g. the last vector in the process) in the above formulas must be the original signal of the user of interest. The resulting basic vector z_k is then the same vector as that generated in the correlation matrix inversion approach taken in [1].

As an example, the 2 user case presented in [1] is re-examined here using the proposed GS procedure. Using the terminology used in [1], the 2 user's signals can be represented as 3 users. One is the signal of interest, and the other two being the left and right sub-bits of the interfering user as seen in Figure 2.1.

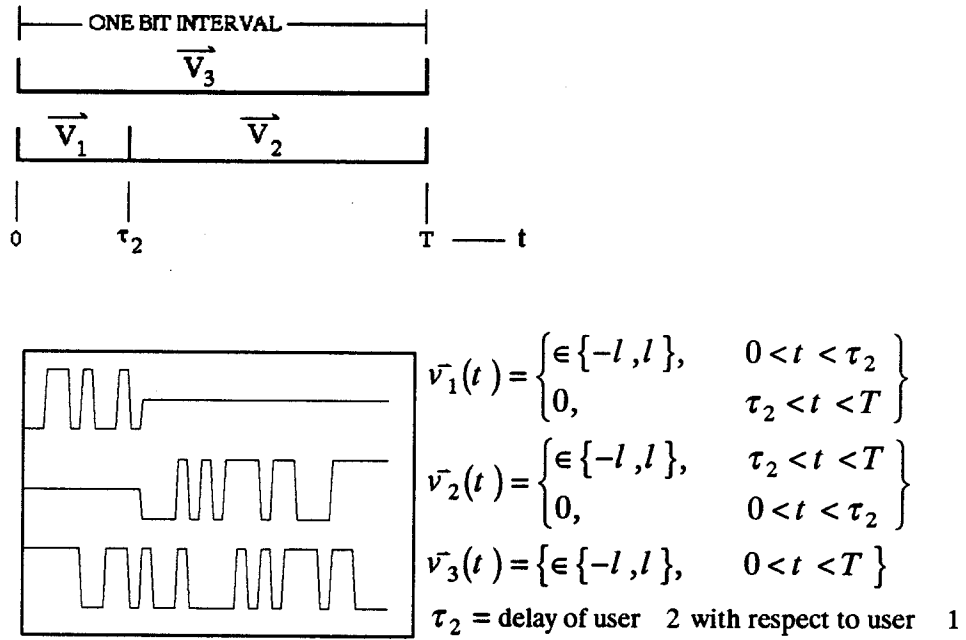


Figure 2.1 - Illustration of one shot break-up of interfering user

$$\text{let } \rho_{21} = \langle \bar{s}_2^L, \bar{s}_1 \rangle, \rho_{12} = \langle \bar{s}_2^R, \bar{s}_1 \rangle, e_2 = \langle \bar{s}_2^L, \bar{s}_2^L \rangle, 1 - e_2 = \langle \bar{s}_2^R, \bar{s}_2^R \rangle$$

$$\text{let } \bar{v}_1 = \bar{s}_2^L, \bar{v}_2 = \bar{s}_2^R, \bar{v}_3 = \bar{s}_1$$

$$\begin{aligned} \bar{w}_1 &= \bar{v}_1 / \|\bar{v}_1\| = \bar{s}_2^L / \langle \bar{s}_2^L, \bar{s}_2^L \rangle^{1/2} \\ \bar{w}_2 &= \bar{z}_2 / \|\bar{z}_2\| \quad | \quad \bar{z}_2 = \bar{v}_2 - \langle \bar{v}_2, \bar{w}_1 \rangle \bar{w}_1 = \bar{v}_2 \text{ since } \langle \bar{v}_2, \bar{w}_1 \rangle = 0 \\ \bar{w}_2 &= \bar{v}_2 / \|\bar{v}_2\| = \bar{s}_2^R / \langle \bar{s}_2^R, \bar{s}_2^R \rangle^{1/2} \\ \bar{w}_3 &= \bar{z}_3 / \|\bar{z}_3\| \quad | \quad \bar{z}_3 = \bar{v}_3 - \langle \bar{v}_3, \bar{w}_1 \rangle \bar{w}_1 - \langle \bar{v}_3, \bar{w}_2 \rangle \bar{w}_2 \\ \bar{w}_3 &= \bar{s}_1 - \frac{\langle \bar{s}_1, \bar{s}_2^L \rangle \bar{s}_2^L}{\langle \bar{s}_2^L, \bar{s}_2^L \rangle} - \frac{\langle \bar{s}_1, \bar{s}_2^R \rangle \bar{s}_2^R}{\langle \bar{s}_2^R, \bar{s}_2^R \rangle} \bigg/ \|\bar{z}_3\| \\ \bar{w}_3 &= \frac{\bar{s}_1 - \rho_{21} \bar{s}_2^L / e_2 - \rho_{12} \bar{s}_2^R / (1 - e_2)}{\|\bar{z}_3\|} \end{aligned}$$

The numerator in the final expression for \bar{w}_3 is recognized as eq. 4.152 in [1] which is the new basis pattern. The denominator $\|\bar{z}_3\|$ is the square root of both the efficiency and the near far resistance as defined in [1]. It can be calculated as follows:

$$\begin{aligned} \|\bar{z}_3\| &= \sqrt{\langle \bar{z}_3, \bar{z}_3 \rangle} \\ &= \sqrt{\left(\bar{s}_1 - \frac{\rho_{21} \bar{s}_2^L}{e_2} - \frac{\rho_{12} \bar{s}_2^R}{1 - e_2} \right) \left(\bar{s}_1 - \frac{\rho_{21} \bar{s}_2^L}{e_2} - \frac{\rho_{12} \bar{s}_2^R}{1 - e_2} \right)} \\ &= \sqrt{\langle \bar{s}_1, \bar{s}_1 \rangle - \frac{\rho_{21} \langle \bar{s}_1, \bar{s}_2^L \rangle}{e_2} - \frac{\rho_{12} \langle \bar{s}_1, \bar{s}_2^R \rangle}{1 - e_2}} \\ &= \sqrt{1 - \frac{\rho_{21}^2}{e_2} - \frac{\rho_{12}^2}{1 - e_2}} \end{aligned}$$

The normalized power in the waveform is the square of the above result. The efficiency is $1 - \frac{\rho_{21}^2}{e_2} - \frac{\rho_{12}^2}{1 - e_2}$, which is recognized as equation 4.150 in [1].

Extension for Three or More Users

In the two user case, the derivation of the orthonormal basis vectors was made easier by the fact that the interfering user is split into two parts which are automatically orthogonal to each other. This section derives the basis vector for the 3 user case and then shows how this can be expanded into N users. Also presented is a simplified formula which ignores terms that have little effect on the final waveform and efficiency. This new algorithm reduces the amount of computational complexity required to implement the GS system.

In order to make the terminology easier to follow, each interfering user's sub-bit will be numbered as a separate user. The normalization process is rearranged so that square root operations are avoided. This approach will ease the computational complexity of the circuits used later to implement this algorithm. The delays between users will be shown as increasing without loss of generality. The derivation assumes that the delay of each user is independent of the others and is equally probable on the interval $[0, T]$. A diagram showing the setup for the 3 user case is shown in Figure 2.2.

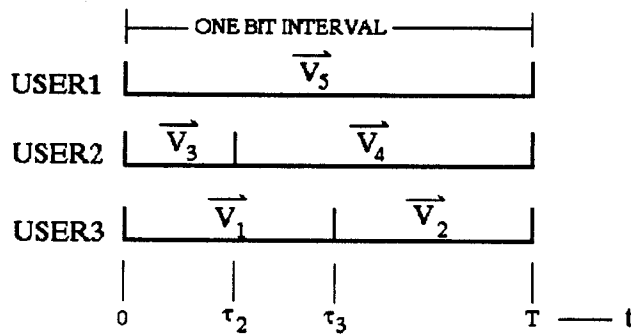


Figure 2.2 - Three user case bit interval

The steps used to compute the basis vectors using the GS algorithm are:

$$\begin{aligned}
\text{let } \rho_{ij} &= \langle \bar{v}_i, \bar{z}_j \rangle, e_i = \langle \bar{z}_i, \bar{z}_i \rangle \\
\bar{z}_1 &= \bar{v}_1 \\
\bar{z}_2 &= \bar{v}_2 - \rho_{21}\bar{z}_1/e_1 \\
\bar{z}_3 &= \bar{v}_3 - \rho_{31}\bar{z}_1/e_1 - \rho_{32}\bar{z}_2/e_2 \\
\bar{z}_4 &= \bar{v}_4 - \rho_{41}\bar{z}_1/e_1 - \rho_{42}\bar{z}_2/e_2 - \rho_{43}\bar{z}_3/e_3 \\
\bar{z}_5 &= \bar{v}_5 - \rho_{51}\bar{z}_1/e_1 - \rho_{52}\bar{z}_2/e_2 - \rho_{53}\bar{z}_3/e_3 - \rho_{54}\bar{z}_4/e_4
\end{aligned}$$

In the case where one or more of the interfering user's bit delay is zero, the terms with the discarded bits are just set to zero (The hardware circuits can account for the 0/0 division as 0). As can be seen from the formula for z_5 , as the number of users diminishes, the resulting pattern used to correlate the received signal approaches the desired user's original pattern. The above formula for the 3 user case can easily be expanded to multiple users using the recurrence equation

$$\bar{z}_i = \bar{v}_i - \sum_{j=1}^i \rho_{ij}\bar{z}_j/e_j.$$

The key point in implementation is that the user of interest must be the last vector in the GS process. The interfering users can be input in any order though normally the left and right sub-intervals would be paired together.

Reduction of Computational Complexity

If the above formulas for z_i are examined more closely, some of the terms contribute little or nothing to the final pattern. The most obvious is the last term in z_2 's equation. Since the v_1 and v_2 patterns do not overlap (by definition if the interfering users are entered paired), ρ_{21} will always be zero and the term does not need to be computed. The last term in z_3 appears to be equal to 0 by a similar

argument but had the delay for user₃ been less than user₂'s then ρ_{32} would not be 0 so this term can not be deleted. For z_4 ,

$$\begin{aligned}\rho_{43} &= \langle \bar{v}_4, (\bar{v}_3 - \rho_{31}\bar{z}_1/e_1 - \rho_{32}\bar{z}_2/e_2) \rangle \\ &= \langle \bar{v}_4, \bar{v}_3 \rangle - \frac{\rho_{31}\langle \bar{v}_4, \bar{v}_1 \rangle}{e_1} - \frac{\rho_{32}\langle \bar{v}_4, \bar{v}_2 \rangle}{e_2}\end{aligned}$$

$\langle \bar{v}_4, \bar{v}_3 \rangle = 0$, and either ρ_{41} or ρ_{32} will be 0. In the case shown in Figure 2.2, $\rho_{32}=0$ and $\rho_{31}\rho_{41}$ form a second order product of cross correlations which with typical spreading patterns will contribute little to the generation of the basis vector.

Simulation of a set of maximal patterns has shown that this term can be ignored. The new reduced equations (ignoring relative delays) are then

$$\begin{aligned}\bar{z}_1 &= \bar{v}_1 \\ \bar{z}_2 &= \bar{v}_2 \\ \bar{z}_3 &= \bar{v}_3 - \rho_{31}\bar{z}_1/e_1 - \rho_{32}\bar{z}_2/e_2 \\ \bar{z}_4 &= \bar{v}_4 - \rho_{41}\bar{z}_1/e_1 - \rho_{42}\bar{z}_2/e_2 \\ \bar{z}_5 &= \bar{v}_5 - \rho_{51}\bar{z}_1/e_1 - \rho_{52}\bar{z}_2/e_2 - \rho_{53}\bar{z}_3/e_3 - \rho_{54}\bar{z}_4/e_4\end{aligned}$$

This reduced approach can also be extended to multiple users. In the case of 4 users, the above equations hold and

$$\begin{aligned}\bar{z}_6 &= \bar{v}_6 - \rho_{61}\bar{z}_1/e_1 - \rho_{62}\bar{z}_2/e_2 - \rho_{63}\bar{z}_3/e_3 - \rho_{64}\bar{z}_4/e_4 \\ \bar{z}_7 &= \bar{v}_7 - \rho_{71}\bar{z}_1/e_1 - \rho_{72}\bar{z}_2/e_2 - \rho_{73}\bar{z}_3/e_3 - \rho_{74}\bar{z}_4/e_4 - \rho_{75}\bar{z}_5/e_5 - \rho_{76}\bar{z}_6/e_6\end{aligned}$$

If the restriction is added that the sub-interval patterns are input into the circuit with respect to increasing delays (but each interfering users' sub-intervals paired), then the number of computational blocks can be reduced further. This reduction is due to the cross correlation of z_i and v_{even} always being zero (e.g. $\rho_{\text{even}1}=0$). As the number of users increase then the cross correlation of z_i and v_j , where i is odd and j is greater than i , is also zero.

Simulation Results

In order to determine how ignoring the last term for z_3 affects the modified GS approach, simulation was performed using three 31 chip maximal sequences. The interfering users were each delayed from 0 to 30 chip intervals within each bit interval, and the output of the normalized correlator was sampled at the time of maximum SNR. Figure 2.3 shows four graphs detailing the results. The standard correlator is the output of a correlator matched to the user of interest's pattern only. The modified GS correlator is using a signal closely orthogonal to the interfering users. This signal is recalculated for each relative shift of all interfering users. The 3D plots show the relative delays of the interfering users in the X and Y directions. The Z direction is the relative output from the correlator. The lower graphs show actual levels looking left into the 3D graphs. Simulation of the full GS correlator was also performed and its output was flat at a value of 1 for all relative delays. Table 2.1 shows the average SNR, and worst case peak SNR at the output of the correlator.

Correlator	Average SNR	Lowest SNR	σ^2
Standard	14.6 dB	6.9 dB	34E-3
Modified	44.4 dB	24.3 dB	36E-6

Table 2.1 - Simulation results

From this data we see that on average, the total power of the interfering users can be increased by about 30 dB, though for the worst case delay the increase is only 17 dB. If more rejection of the users are required, the full GS approach can be used.

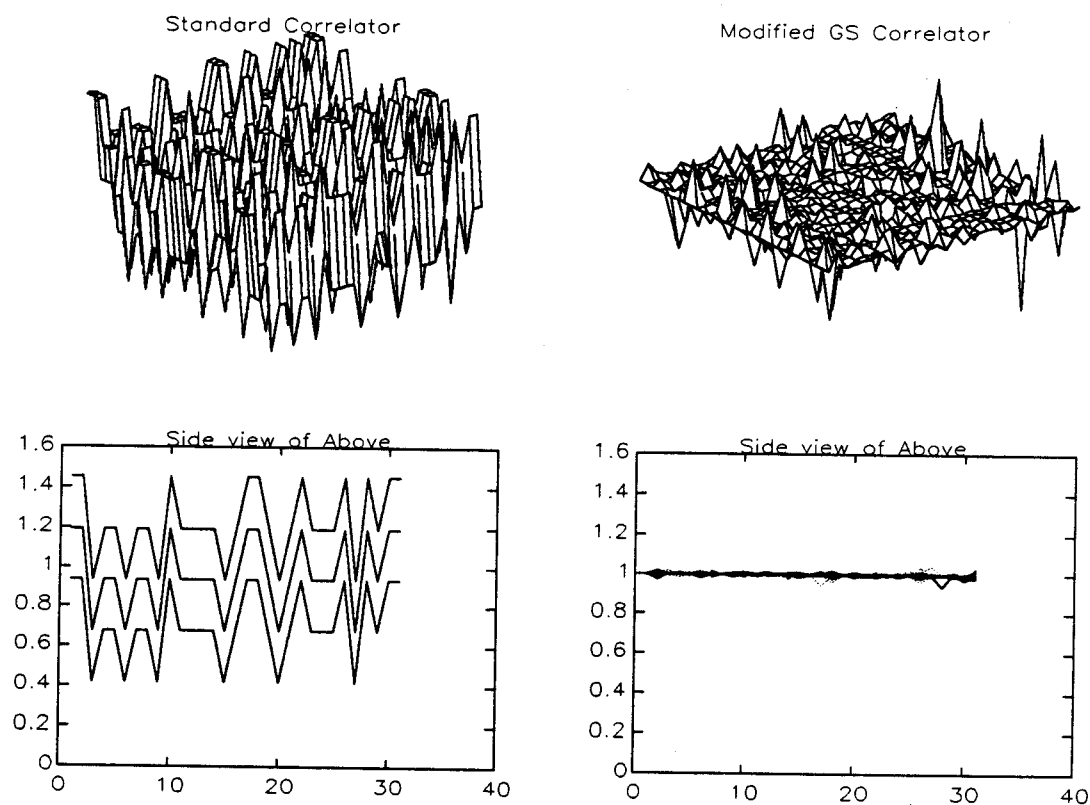


Figure 2.3 - Normalized output from standard and modified GS Correlators. The X and Y axes of the 3D plots are the relative chip delays (1-31) for the two interfering users patterns to the desired users pattern.

CHAPTER 3

IMPLEMENTATION OF PROPOSED SOLUTION

This section examines how the near-far resistant "one-shot" correlator and detector could be implemented. Appendix A contains a study on the "one-shot" proposal by Verdu and Lupas. The main difficulty found in their algorithm is the circuitry required to invert the correlation matrix. Since the proposed solution in this thesis does not require matrix inversion, this roadblock is avoided. Both the full GS method and the modified GS method are presented and the tradeoffs compared.

Figure 3.1 shows a block diagram, containing the top level functions. The GS Orthogonal Signal Generator's (GSOSG) purpose is to receive the known patterns of the interfering users and the user of interest from the synchronizer and pattern generators. The sequence of v_i patterns consist of $\pm l$, where l is a constant, in the sub-interval of interference and 0 in the other sub-interval. For the user of interest, the pattern will always be $\pm l$. Better performance can be achieved by using bandlimited pattern signals since the received waveform will also be bandlimited due to constraints on the channel and receiver circuitry. In the latter case, the v_i s would be analog signals.

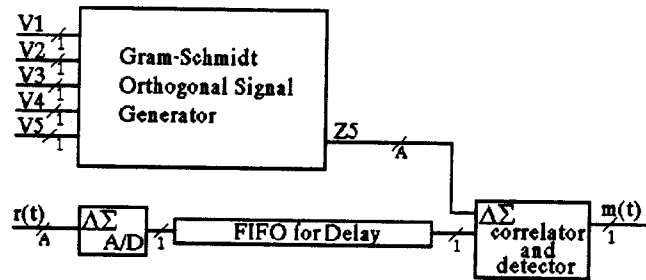


Figure 3.1 - Block diagram of one-shot correlator and detector

The received signal $r(t)$ is processed by an oversampling A/D and digitized into a one bit sequence. This digital sequence is then delayed by an appropriate interval required by the GSOSG to create the near-far resistant pattern. The digitized received signal and the near-far pattern are correlated using a "Delta-Sigma Correlator" (multiply, add and decimate portion) that is described in the next chapter. This new correlator is also used in the GSOSG block as well.

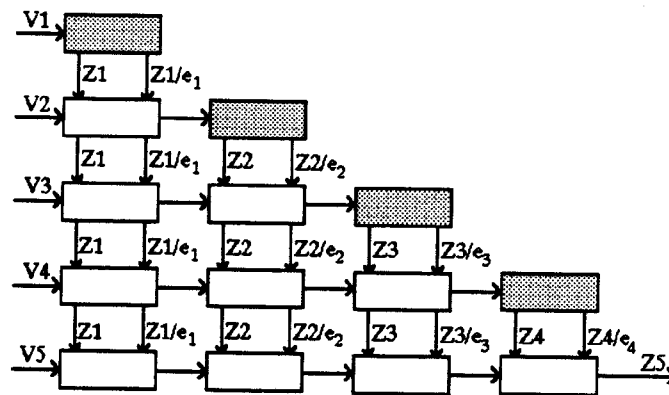


Figure 3.2 - Block diagram of GSOSG

Figure 3.2 shows the block diagram for the GSOSG which uses all of the terms in the equations. The shaded blocks functions are shown in Figure 3.3 which are responsible for determining the energy in the sub-interval and delaying the pattern by one bit cycle. The energy is provided by multiplying the signal by itself and

summing the results through the entire bit interval. The resulting desired N bits would be latched at the end of the bit interval. This result is then used to control the attenuation of a programmable amplifier which operates on the delayed digitized input signal.

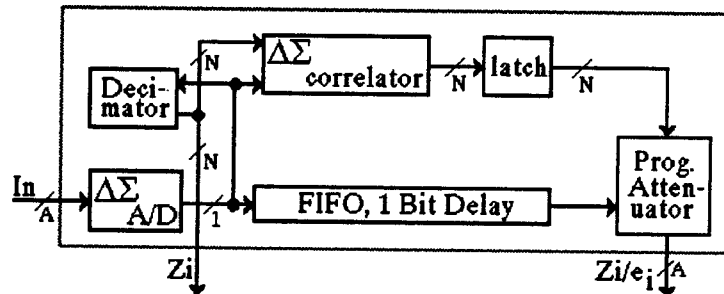


Figure 3.3 - Shaded module block diagram

The non-shaded blocks shown in Figure 3.4 perform a similar function. The input signal is converted to a 1 bit digital signal using the delta-sigma A/D convertor. This signal is then fed along with the decimated and digitized z_i into the delta-sigma correlator.

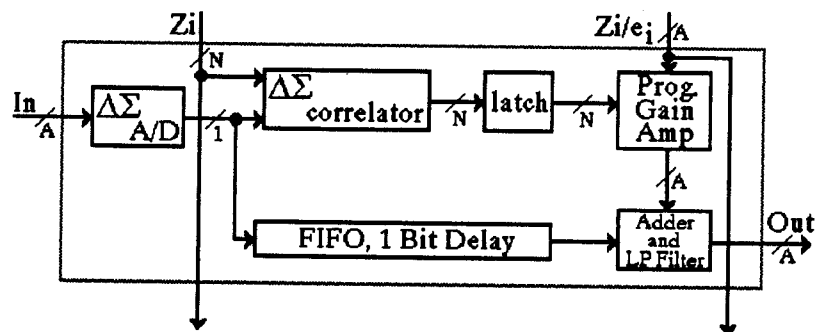


Figure 3.4 - Non-shaded module block diagram

This correlator is simplified since it only needs to be an N bit by one bit multiply. The output of the correlator forms ρ_{ij} which is the correlation between the

input signal and z_i . The result is latched at the end of the bit interval and controls the gain of the amp forming $\rho_i z_i / e_i$. This product is then added to the delayed input signal, summed and low pass filtered to remove the undesirable noise due to the convertor. This forms an analog output signal which is fed into the next block. This technique allows for variable chip delay intervals between interfering users.

To implement the full GS algorithm requires $N(2N-1)-1$ total processing blocks. It also requires a time delay of $2N$ bit intervals plus the time required for the delay through the decimators and filters. This approach, however, does allow real time on the fly processing of the incoming waveform. Although this total processing unit only decodes one user, the number of these processing units required to decode multiple users is N for N users.

As shown previously, some of the computational complexity could be removed at a cost of reduced performance. Although the number of processing element reduction is minimal, the organization of the block diagram can be changed (Figure 3.5) to allow some parallel processing and thus a reduction in the time complexity to N bit delays for N users. The detailed processing blocks are similar to those shown in Figures 3.4 and 3.3 respectively for the blank and shaded elements.

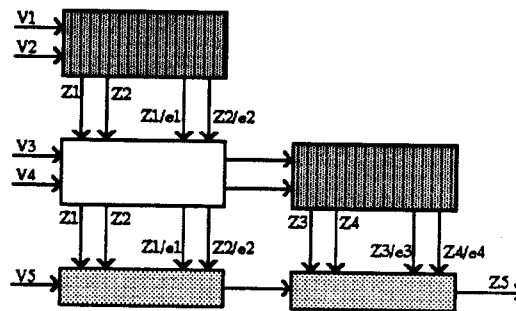


Figure 3.5 - Modified GS block diagram

Although these functional blocks were derived to use delta-sigma modulation, conventional A/D techniques could also be used. The delta-sigma approach was taken to allow easier VLSI implementation of the circuits.

Another approach that could be taken to implement the GSOSG is to use digital signal processors at each processing block. This action would keep all of the intermediate results in digital form and improve the accuracy of the final result.. The amount of delay required for $r(t)$ would depend on the number of DSPs used and how fast they operate.

Simulation Results

In order to implement the GSOSG block, the precision from the delta-sigma A/D convertors must be better than that required in correlating the final signal in the delta-sigma correlator examined in the next chapter. In the final correlation case, the object is to determine the polarity of the message bit. In the GSOSG case, the goal is to generate a new pattern, which is orthogonal to the interfering users. Any error that is introduced will tend to cause the generated signal to not be truly orthogonal. Of course, the final pattern only needs to be near-far resistant only over the range of the front-end circuitry.

To get an idea of the accuracy required, simulation was performed which investigated using some of the delta-sigma modulators from chapter 4 and various decimation strategies. It was found, for a randomly chosen pair of relative delays, that the 2ndOrder modulator with triangular decimation (~9 bits) provides good near-far resistance for the three user case. A summary of the results are shown in table 3.1. The multiplication and division operations were done with full floating

point precision. Quantization of these operations will cause additional error in the generation of the orthogonal pattern.

Over-Sampling Ratio	Modulator	Window for Decimator	m.s.e σ^2/chip	ρ_{12}	ρ_{13}	ρ_{11}
128 k=.25	2ndOrder	Boxcar	1.0E-6	0.002	0.0065	0.8865
128	2ndOrder	Triangular	2.2E-8	-1.55E-4	-0.0033	0.8908
128	2ndOrder	Blackman	4.1E-8	1.01E-4	0.0123	0.8939
128	Proj_32	Boxcar	6.78E-7	-0.0019	-0.0092	0.8915
128	Proj_32	Triangular	1.1E-7	-0.0027	-0.0039	0.8865
128	Proj_32	Blackman	8.05E-8	-3.8E-6	-0.0061	0.8880
64 k=.25	2ndOrder	Boxcar	3.925E-6	-0.0024	0.0404	0.8880
64	2ndOrder	Triangular	6.29E-7	-0.0034	0.0287	0.8976
64	2ndOrder	Blackman	3.26E-7	-0.0012	-0.0164	0.8825
64	Proj_32	Boxcar	4.0E-6	-0.0025	0.0079	0.8825
64	Proj_32	Triangular	5.269E-7	0.0189	0.0047	0.8902
64	Proj_32	Blackman	1.55E-7	0.0011	0.0113	0.8902

Table 3.1 - Results of simulation

The mean square error is the average variance of the difference between the ideal pattern and the actual output from the simulation for each chip interval. The correlation coefficients between the simulated output pattern and the three users are listed as ρ_{12} , ρ_{13} , ρ_{11} in the table. Proj_32 is a fifth order delta-sigma modulator which is optimized for minimum noise in the region of 0 to $\pi/32$ (2π is the sampling frequency).

The 2ndOrder modulator with triangular decimation at an oversampling rate of 128 provides good performance and is relatively implementable. More simulation was performed, using these selections, using all possible chip delays of the interfering users. The results are shown in the graphs of Figure 3.6. The top left graph shows a three dimensional picture with the relative delays on the X and Y axis and the

normalized output on the z axis. A sideview of this graph is shown in the lower left plot. The lower right graph is a typical output pattern from the GSOSG.

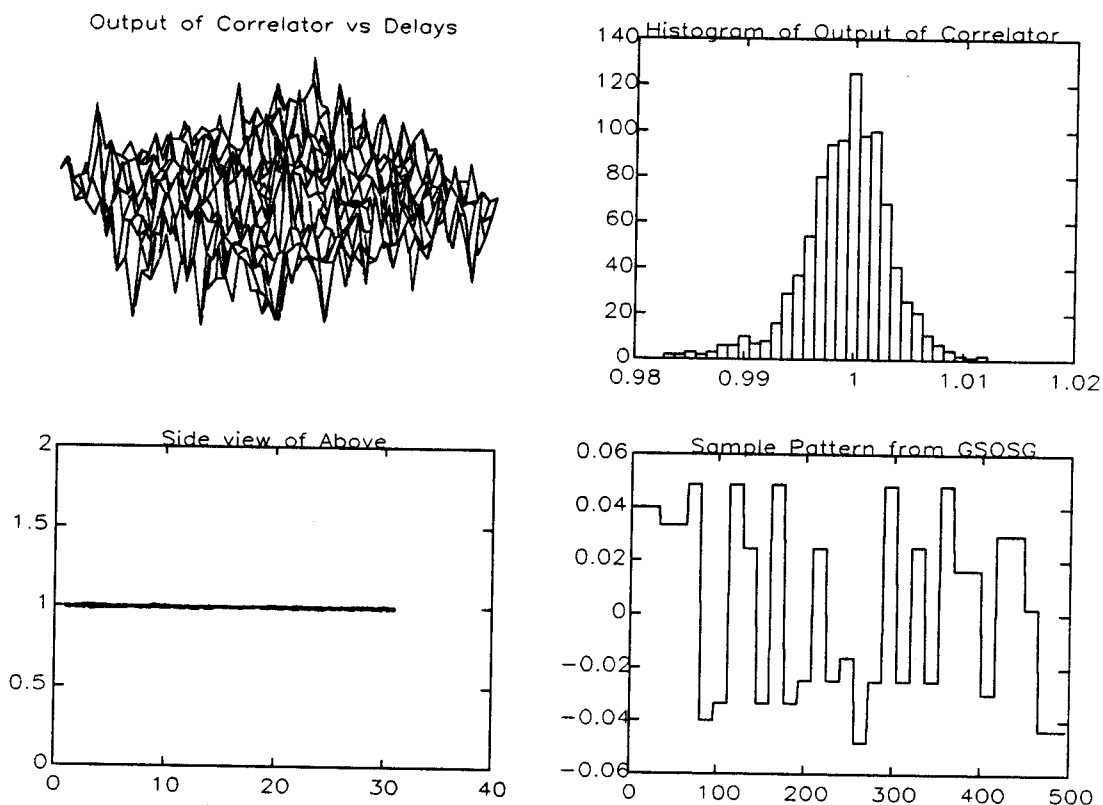


Figure 3.6 - Simulation results of entire system for three users

Note the various levels for each chip interval. This requires that a non-binary correlator be used. The top right graph is a histogram of the normalized output from all the delays showing the distributions due to the error introduced by the delta-sigma modulators. The variance for the noise caused by the interfering users for all relative delays is $1.6\text{E-}5$, which suggests a SNR due to the multiple users of about 48 dB.

Analysis of Error Contribution due to A/D Quantization

This section examines how well the system SNR performs with respect to the number of users in the channel (K), the number of chips per message bit (N), and the quantization of the internal signals due to the $\Delta\Sigma$ A/D convertors. In most practical systems N will be greater than 31 and the number of users will be larger than three. These values were used in the simulations of the system in order to limit the amount of computation. The following derivations allows some idea of what the bit resolution requirements on the A/D convertor are for other values of K , N and SNR.

When the patterns are converted from analog to digital form in the various blocks, the $\Delta\Sigma$ modulator, decimator and other functions provide an error with a variance of σ_{ds}^2 . If the assumption is made that the errors from each block are independent and uncorrelated and that there is no correlation between adjacent chip intervals, then the total variance of the signal at the output of the system is

$$\sigma_{output}^2 = K(2K - 1)N\sigma_{ds}^2$$

where K is the number of users, $K(2K-1)$ is the number of processing blocks, and N is the number of chips per message bit.

In Figure 3.6 it was shown that the error distribution was approximately gaussian. To make the estimate less complicated it will be assumed that the error at the output is uniformly distributed. The formula for equating the variance of a uniform distribution to its quantizing level is known as

$$\frac{\Delta^2}{12} = \sigma^2$$

Substituting this into the previous equation and solving for Δ_{output}

$$\frac{\Delta_{output}^2}{12} \equiv K(2K-1)N\sigma_{ds}^2$$

$$\Delta_{output} \equiv 2\sqrt{3K(2K-1)N\sigma_{ds}^2}$$

Since the noise generated from this digitization process is random, we can set a bound on the worst case SNR by limiting the size of Δ_{output} . Since the error is zero mean, then we need to only limit the maximum noise level to $\Delta_{output}/2$. Assuming that the output signal has been normalized, the SNR at the output of the system is

$$SNR_{output} = 20\log\left(\frac{1}{\Delta_{output}/2}\right)$$

Solving for the quantizing step of the A/D convertor

$$\sigma_{ds}^2 \equiv \frac{\Delta_{output}^2}{12K(2K-1)N} = \frac{\Delta_{ds}^2}{12}$$

$$\Delta_{ds} \equiv \left(\frac{\Delta_{output}^2}{K(2K-1)N}\right)^{1/2}$$

The bit resolution of the A/D convertor is then

$$\# \text{ bits} = \log\left(\frac{1}{\Delta_{ds}}\right) / \log(2)$$

A set of graphs using the above equations is shown in Figures 3.7, 3.8, 3.9 and 3.10 for 3, 5, 10, and 20 users respectively for the worst case SNR desired. In the 3 user case examined previously with the 2ndOrder modulator and triangular decimation, the worst case SNR was 35.23dB. From Figure 3.7, the equated resolution required of the A/D is 9 bits which agrees with the modulator/decimator pair. Since the 2ndOrder modulator is capable of more bits of resolution using a better decimator and bandlimiting on the input signals, the SNR could be improved. With a 16 bit resolution, it is possible to handle 20 users with a 1023 chip sequence.

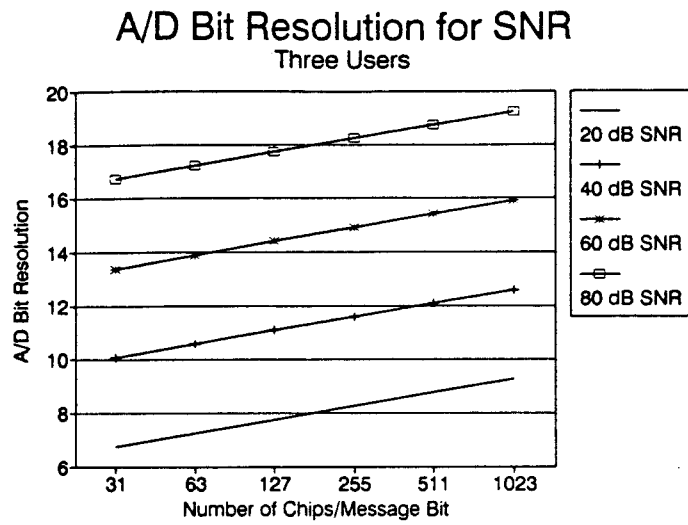


Figure 3.7 - A/D Resolution required for three user case

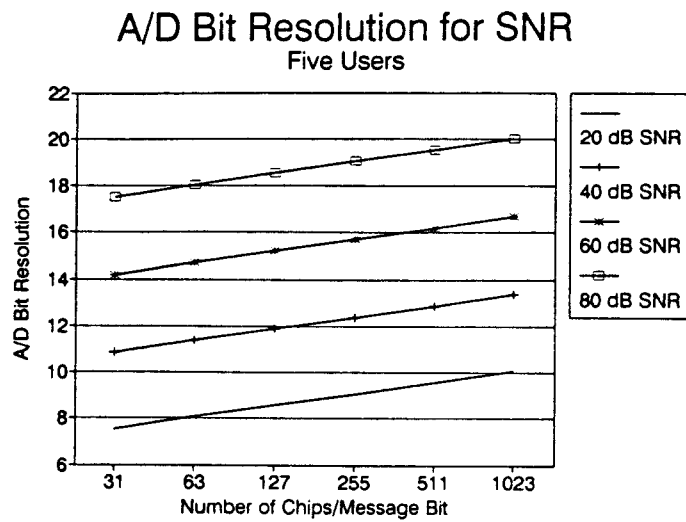


Figure 3.8 - A/D resolution required for five user case

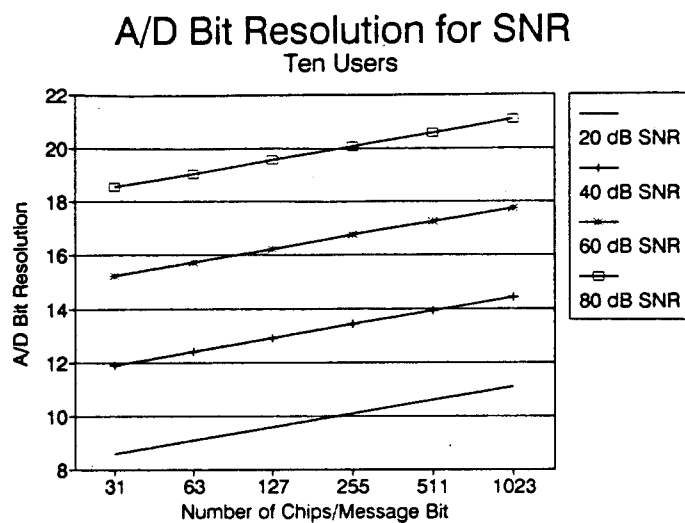


Figure 3.9 - A/D resolution required for ten user case

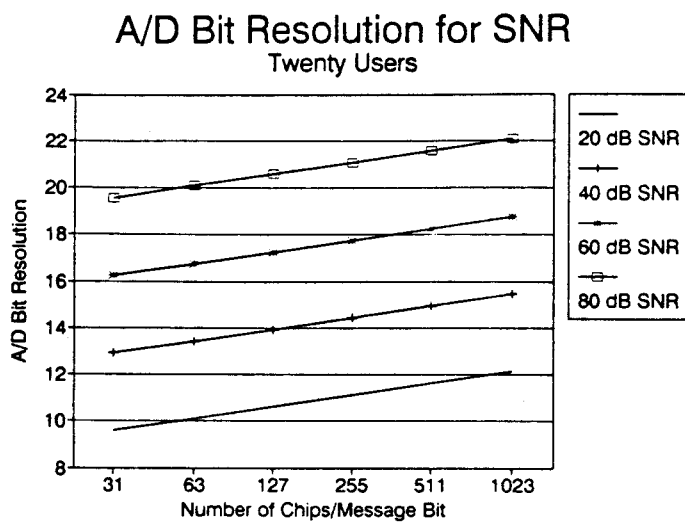


Figure 3.10 - A/D resolution required for twenty user case

Analysis of Error due to Delay Resolution

It was assumed previously that the relative delays between user chip intervals were known. This section examines how the resolution of the delay intervals affects the SNR at the output of the correlator. Since the signal will be sampled at discrete time intervals for processing by the proposed system, the sampling rate will affect the output of the final correlator.

The output of a correlator using a pattern matched to the user exhibits a familiar triangular function as seen in Figure 3.11. Here τ is the relative delay from the peak output of the correlator. Maximum output from the correlator is at $\tau=0$. As the delay varies from 0 to T (the chip interval length), the output of the correlator decays linearly as

$$R_{signal}(\tau) = 1 - \frac{|\tau|}{T}, \text{ where } |\tau| < T$$

for a normalized pattern. For the GSOSG pattern, the result is similar so the above formula will be used.

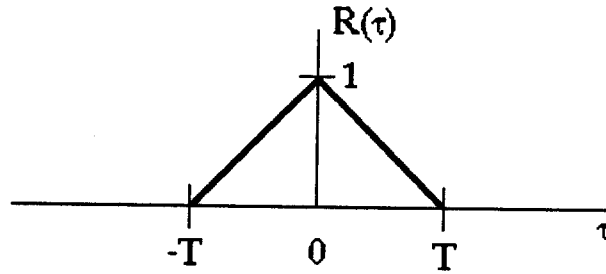


Figure 3.11 - Normalized output of correlator for one chip interval

In the situation proposed in this thesis, the received signal $r(t)$ is correlated with a pattern that is orthogonal at $\tau=0$ to the interfering users. At delays other than $\tau=0$ there will be some correlation between the new generated pattern and the interfering users which will reduce the SNR and thus have less near-far resistance. For ease of

analysis, it is assumed that one possible output (others exist) of the correlator using the generated pattern and just one interfering user signal as input is

$$R_{noise}(\tau) = c \frac{|\tau|}{T}$$

where c is the maximum value the cross-correlation achieves. This is seen graphically in Figure 3.12.

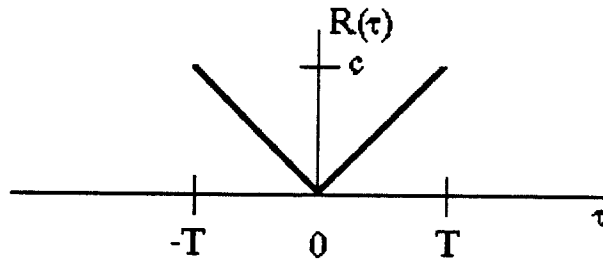


Figure 3.12 - Output of correlator with interfering user as input

The SNR at the output of the correlator when $r(t)$ is the input is

$$SNR(\tau) = \frac{R_{signal}(\tau)}{R_{noise}(\tau)} = \frac{1}{c} \left(\frac{1 - |\tau|/T}{|\tau|/T} \right) = \frac{1}{c} \left(\frac{T}{|\tau|} - 1 \right), \text{ where } \tau \neq 0$$

This result shows that the SNR is directly related to the resolution of the delay τ .

Notice at $\tau=0$, the SNR will only be limited by other noise sources in the channel. As τ increases the SNR will also decrease. This means that the tracking, synchronization and sampling of the received signal $r(t)$ will directly affect the maximum near far resistance possible by the proposed circuit. Figure 3.13 shows the simulated output of the proposed system for a three user case for a random selection of delays for the users. The x axis is the chip interval where 64 is $\tau=0$. On the left hand side of the graph, the two interfering users output from the correlator add together and the effect of sampling the correlator too early causes a dramatic decrease in the SNR. On the right side of the graph, the two interfering users' output from the correlator

tend to cancel and the SNR decays close to linearly as expected. The dashed line is the SNR from the output of a correlator using a pattern matched to the desired user rather than that generated by the GSOSG circuitry.

Since the resolution of the delays will affect the SNR, the sampling rate will tend to dominate the maximum near-far resistance of the circuit. By using higher sampling rates, the $\Delta\Sigma$ modulator's resolution will also increase which will reduce the error caused by quantizing the signal. The quantizing error will decrease faster with an increase in the sampling rate than will the delay resolution error. The $\Delta\Sigma$ modulators can thus be of a lower order which will lower the complexity of the circuit. Higher sampling rates will be limited due to the speed of the technology used to implement the system.

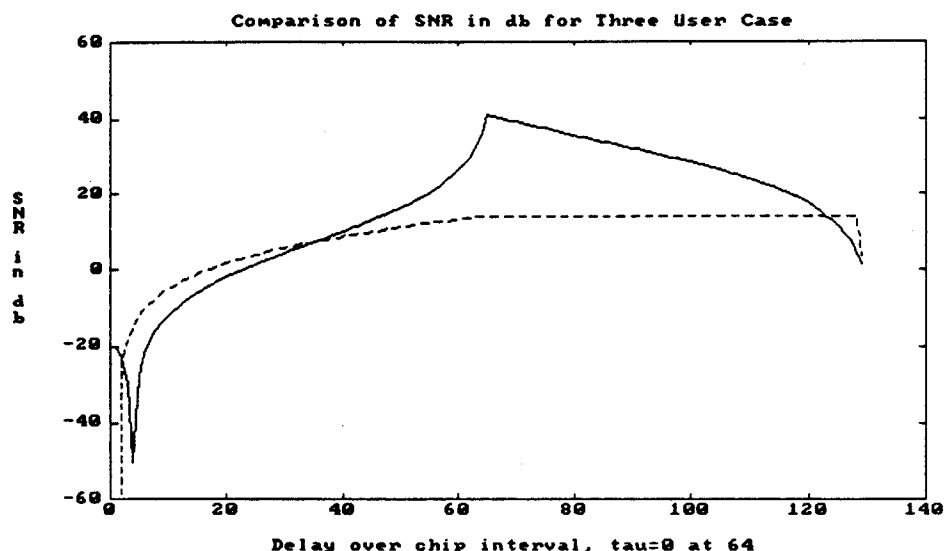


Figure 3.13 - SNR vs. delay resolution for proposed and standard detectors. The dashed line is the standard detector, the solid line is the proposed detector.

CHAPTER 4

DELTA-SIGMA CORRELATOR

Introduction

This chapter examines the role of delta-sigma modulation in a spread spectrum (SS) direct sequence (DS) correlator and detector. The motivation for this proposal is the search for simple, low cost, flexible, implementable and predictable hardware which can produce such a correlator. Some of the musts and wants for such a SS correlator are:

Wants	Musts
Flexibility in choosing design options	High SNR for low bit error detection and low false alarm rates
Small chip area	Work at speeds close to the limits of technology
Expansibility	Be able to use patterns outside of the domain of $(-1, +1)$ or $(0, +1)$
Minimal external control of settings	

Table 4.1 - Musts and wants of a multi-level correlator

The inherent features of the delta-sigma A/D such as excellent linearity, low complexity, bit serial output and the ability to control the shape of the noise power spectrum lead to its natural selection as a block in a correlator. Both the delta-sigma modulator and the correlator for SS have common elements such as the following:

SS DS Correlator	Delta Sigma Modulator
Bit Serial Data Stream	Bit Serial Output
Noise Shaped Signal	Noise Shaped Error Spectrum
Correlator consists of Multiply and Add	Decimator consists of Multiply and Add
Pseudo Random Pattern used for spreading	Noise pattern is Random on Higher Order Modulators

Table 4.2 - Comparison of SS correlator to delta-sigma modulator

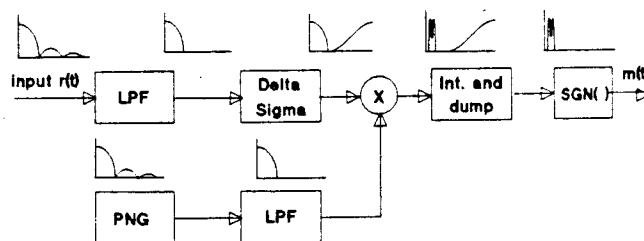
It therefore seems reasonable to expect that some of the operations could be shared by the correlator and modulator and thus reduce the complexity of the total design.

Block Diagram of System

The block diagram for the architecture used for this project is shown in Figure 4.1. The message ($m(t)$), a binary signal, is multiplied by a pseudo-random pattern which generates the signal $r(t)$. In general, $r(t)$ would be used to modulate some carrier and be demodulated at the receiver back to baseband. In this chapter, it is assumed that synchronization with $r(t)$ has been accomplished. A low pass filter is used for limiting the bandwidth and reducing the input into the modulator to prevent instability. The final signal has a power of $\sim -12.5\text{dB}$ ($\sim .225\text{V}_{\text{rms}}$). The delta-sigma modulator converts the analog signal into a serial bit stream of $\{-1, +1\}$. The output of the modulator is multiplied by the pseudo-random code (addressed later in the paper). This code is generated at the receiver and is in sync with the transmitted signal. In the block diagram it passes into an optional low pass filter which limits its bandwidth. After the multiplier, the signal is integrated and the output dumped at specific intervals of $R \times \text{spread code length}(31)$. The final signal then has the signum operation performed on it to convert it back to the original binary signal.

Timing Used in System

Figure 4.2 shows the timing used in this chapter. Each bit of the message is spread by a 31 'chip' pattern which has pseudo random properties. The pattern is an M sequence which has excellent autocorrelation properties. Each chip is then oversampled by R samples before entering the Delta-Sigma modulator. For each case examined, the data sequence consisted of '1001'.



Block Diagram of Delta Sigma Correlator

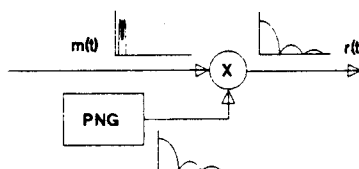
Block Diagram for $r(t)$ Generation

Figure 4.1 - Block diagram

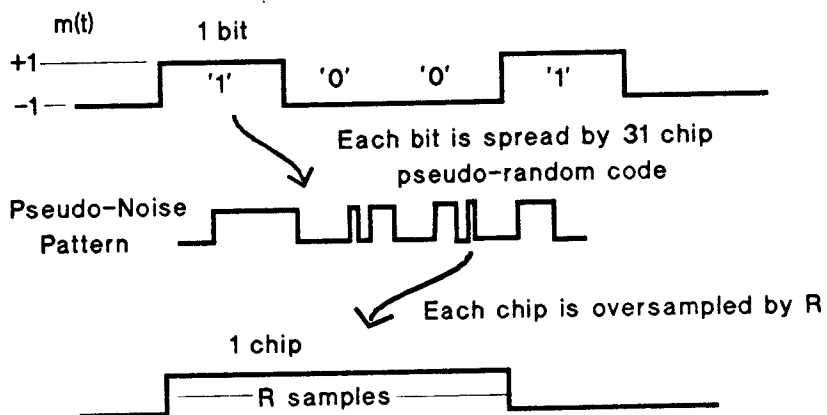


Figure 4.2 - Illustration of timing used in system

Modulators Examined

In order to see what effect the noise transfer function has on the operation of the correlator, six modulators were used which have different properties. The modulators consist of the 1stOrder, 2ndOrder, 3rdOrder-c[17], Example[17], Proj_32, and Proj_16. The poles and zeros in the Z domain of the noise transfer function are tabulated in table 4.3.

Modulator	Zeros	Poles
1stOrder	1	0
2ndOrder	1 1	0 0
3rdOrder-c	1 0.9993+/-j0.038	0.6166 0.7297+/-j0.3176
Example	1 0.999650+/-j0.026429 0.999011+/-j0.044467	0.74545 0.778479+/-j0.136363 0.880630+/-j0.249593
Proj_32	1 0.998603+/-j0.052839 0.996045+/-j0.088847	0.74450 0.777593+/-j0.136819 0.88001+/-j0.250536
Proj_16	1 0.994416+/-j0.105531 0.984213+/-j0.177699	0.74069 0.774055+/-j0.138631 0.87754+/-j0.254299

Table 4.3 - Poles and zeros of error transfer function for modulators

Example has a corner frequency at $\pi/64$, Proj_32 at $\pi/32$, Proj_16 at $\pi/16$ where 2π is the sampling frequency. These filters are designed such that the amount of noise power in the signal band is minimized. Figure 4.3 shows the power spectrum of the transfer functions. Figure 4.4 shows a closer view in the range of $0-\pi/4$.

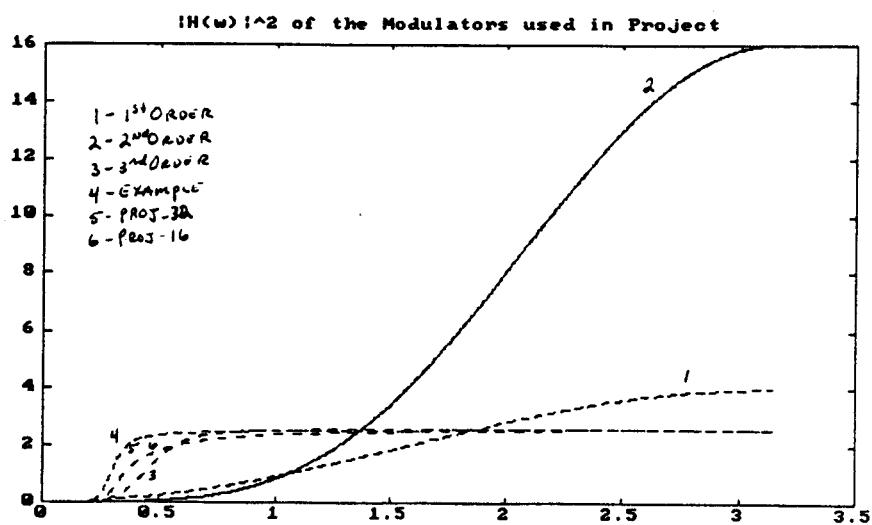


Figure 4.3 - Power spectrum of noise transfer functions used in project

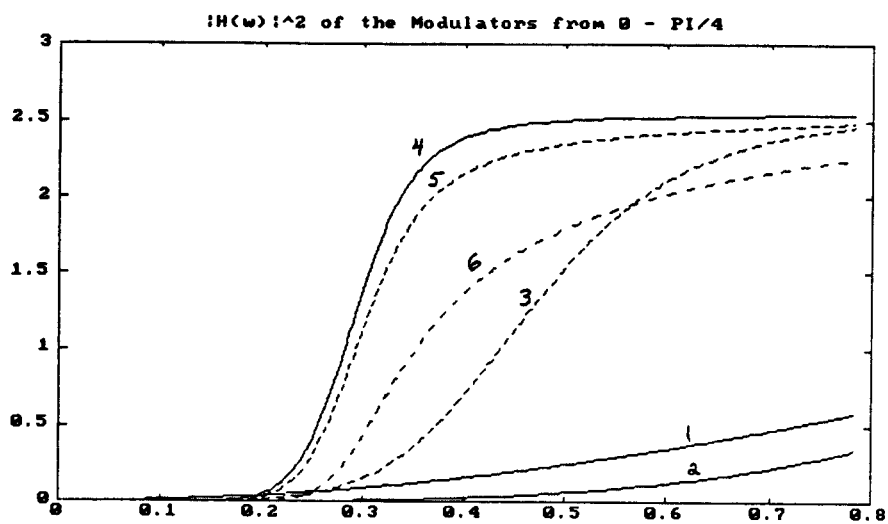


Figure 4.4 - Closer view of power spectrum in range of 0 to $\pi/4$

2ndOrder Noise SNR Predictions

In order to verify the simulation results, the 2ndOrder modulators noise spectrum was derived. When the output of the modulator and the PNG pattern are multiplied, it is equivalent to convolution in the frequency domain, and so N_o^2 , the noise power, will change. The modulators which have their zeros of the noise transfer function spread out on the interval from 0 to π/R_{filter} are more difficult to analyze but they can be estimated. The results show that the noise power is increase by a factor of 2.6. The formula for N_o^2 is (assuming σ_e^2 in the modulator = 1/3):

$$N_o^2 = 0.548/R^5$$

The calculated predictions and simulation results for the SNR for the 2ndOrder modulator are listed in table 4.4 below:

Oversampling Factor	Calculated Results	Simulation Results
R=2	5.199 dB	12.73 dB
R=8	35.30 dB	34.19 dB
R=16	50.35 dB	52.41 dB
R=32	65.4 dB	64.01 dB
R=64	80.5 dB	69.55 dB

Table 4.4 - Comparison of calculated to simulated SNR for 2ndOrder modulator

Simulations

SNR at Input to Detector

The first simulations done were to feed $r(t)$ into each of the modulators and to perform a rectangular decimation on the output from the modulator. The pattern used to multiply the output of the modulator is band limited. The decimation used consists of a rectangular window of length $R \cdot 31$. Other decimation methods could be used to reduce the noise level further. The output of the decimator is a signal at the message bit rate. This is the signal that would be sent to the detector which is just the signum function. The output from the decimator was examined to determine the SNR ($10 \cdot \log_{10}(m^2/\sigma^2)$) going into the detector. The results are tabulated in Figure 4.5 and the SNR at the detector input is graphed in Figure 4.6.

Due to the tonal nature of the noise shaping of the 1stOrder modulator, its results are difficult to predict¹, but the other modulators are more predictable. At each R , the modulators with the least noise in the signal band have the higher SNR's. At $R=8$, it is expected that the 2ndOrder would have the highest SNR since it has the lowest power and the results confirm this hypothesis. At $R=64$, it is expected that example would have the highest SNR since its transfer function indicates the least noise and this is also verified. From the graph, it is seen that as the sampling rate increases the SNR increases. Thus there is a distinct tradeoff between SNR and oversampling ratio. For the detector to have a decent (10^{-6}) error rate, only about 12

¹ If the noise consists of tones that fall on or near the zeros of the decimation filter, then it is possible to have a high SNR as seen in the case where $R=8$. The tones are very dependent on input signal level into the $\Delta\Sigma$ modulator.

dB is required. Adding 6dB for additional margin would bring the minimum SNR required to 18dB. From Figure 4.6, an oversampling ratio of 8 provides enough

MODULATOR	R=64	R=33	R=16	R=8	R=2	R=1
1stOrder:						
	0.22395	0.22321	0.22225	0.19618	0.12903	0.35484
	-0.22415	-0.22601	-0.22137	-0.19618	-0.25807	-0.22581
	-0.22406	-0.22575	-0.22236	-0.19618	-0.19355	-0.22581
	0.22419	0.22307	0.22227	0.19618	-0.19355	0.61290
Mean	0.22409	0.22451	0.22206	0.19618	0.19685	0.38844
Variance	8.93E-09	1.89E-06	1.63E-07	0.00E+00	2.11E-03	2.61E-02
SNR dB	67.50	44.26	54.82	100.00	12.73	7.62
2ndOrder:						
	0.22503	0.22532	0.22853	0.23749	0.19355	-0.03226
	-0.22517	-0.22560	-0.22742	-0.23995	-0.25807	-0.02258
	-0.22505	-0.22567	-0.22681	-0.23064	-0.12903	-0.09677
	0.22521	0.22566	0.22664	0.24330	0.19355	0.22581
Mean	0.22512	0.22556	0.22835	0.23789	0.19685	0.12440
Variance	5.62E-09	2.02E-08	2.99E-07	2.16E-05	2.11E-03	7.46E-03
SNR dB	69.55	64.01	52.41	34.19	12.73	3.16
3rdOrder-c:						
	0.22492	0.22462	0.22129	0.22816	0.19355	0.22581
	-0.22502	-0.22479	-0.22277	-0.21158	-0.19355	-0.35484
	-0.22492	-0.22460	-0.22236	-0.22071	-0.12903	-0.35484
	0.22506	0.22480	0.22285	0.20641	0.19355	0.22581
Mean	0.22498	0.22470	0.22232	0.21688	0.17960	0.29740
Variance	3.97E-09	8.33E-09	3.84E-07	6.99E-05	7.85E-04	4.21E-03
SNR dB	71.06	67.83	51.09	28.28	16.14	13.22
Example:						
	0.22498	0.22492	0.22539	0.20108	0.19355	0.35484
	-0.22503	-0.22512	-0.22124	-0.20607	-0.25807	-0.35484
	-0.22500	-0.22520	-0.22083	-0.18451	-0.19355	-0.29032
	0.22499	0.22527	0.22568	0.20758	0.32258	0.29032
Mean	0.22500	0.22513	0.22330	0.20002	0.24778	0.32419
Variance	3.08E-10	1.69E-08	5.10E-06	8.39E-05	2.90E-03	1.04E-03
SNR dB	82.16	64.76	39.90	26.78	13.26	20.03
Proj 32:						
	0.22493	0.22521	0.22750	0.20675	0.38710	0.22581
	-0.22500	-0.22538	-0.22449	-0.20988	-0.32258	-0.61290
	-0.22495	-0.22523	-0.22129	-0.21302	-0.38710	-0.22581
	0.22505	0.22559	0.22499	0.19267	0.19355	0.35484
Mean	0.22498	0.22535	0.22458	0.20573	0.33212	0.38844
Variance	2.25E-09	2.37E-08	4.89E-06	6.05E-05	6.33E-03	2.61E-02
SNR dB	73.51	63.31	40.14	28.45	12.41	7.62
Proj 16:						
	0.22497	0.22523	0.22290	0.19301	0.19355	0.22581
	-0.22516	-0.22552	-0.22431	-0.19783	-0.25807	-0.48387
	-0.22515	-0.22508	-0.22153	-0.20675	-0.25807	-0.35484
	0.22518	0.22519	0.22340	0.20823	0.19355	0.35484
Mean	0.22511	0.22526	0.22304	0.20155	0.22810	0.36638
Variance	7.39E-09	2.71E-08	1.01E-06	3.96E-05	1.05E-03	8.46E-03
SNR dB	68.36	62.73	46.91	30.11	16.97	12.01

Figure 4.5 - Simulation results of SNR for modulators

ECE 619 Project - SS Del-Sig Correlator Signal/Noise Power ratio at Detector

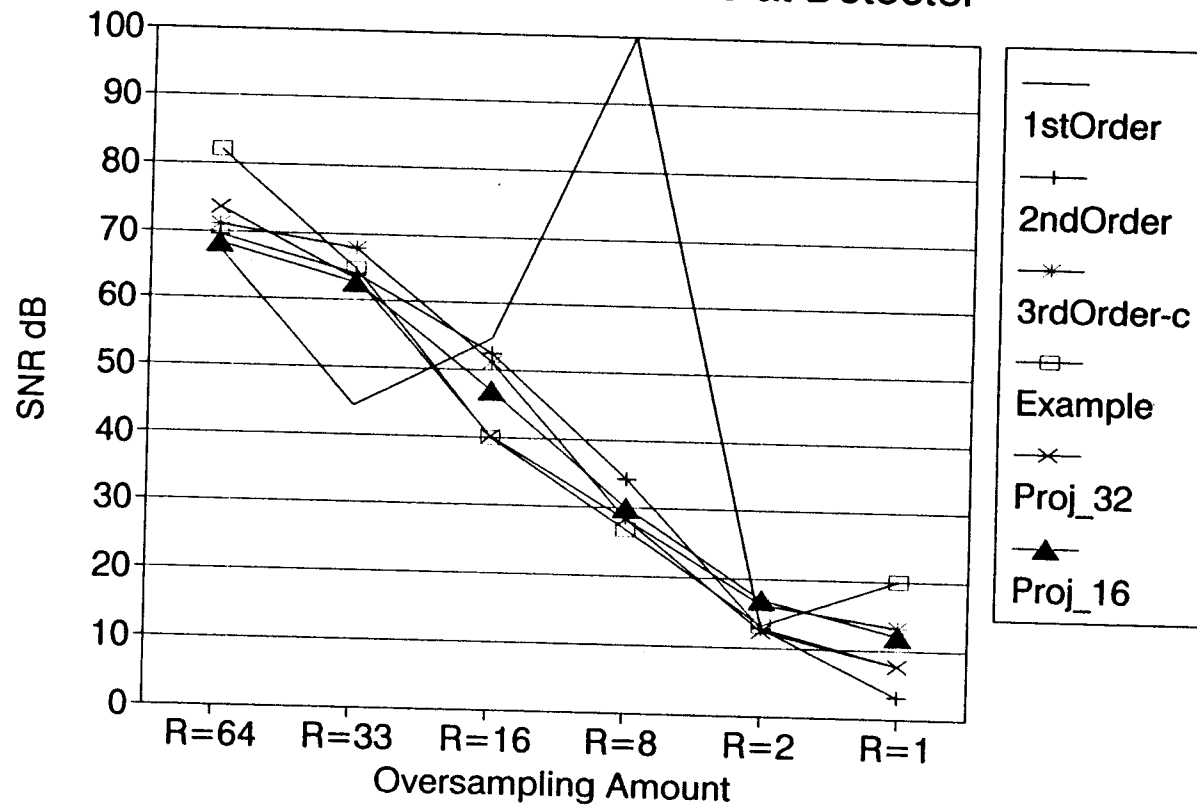


Figure 4.6 - Graph of SNR at detector Input

margin for all modulators tested. Even though higher SNRs could be achieved by higher R 's, the additional benefits would probably be outweighed by other noise degradations of the signal in an actual system. This data of course assumes a bandlimited pattern doing the correlation.

The next set of graphs show the output of the Proj_16 modulator(Figure 4.7), and the multiplier(Figure 4.8). After the modulator, the spreading of the message and the noise shaping of the delta-sigma modulator are both evident. After the multiplier, the signal is despread and the noise band is spread by the pseudo-noise spreading pattern. Figure 4.9 shows the frequency spectrum of output of the Proj_16 modulator with the sine input. The point here is that although the input signal has different spectral components but with the same energy, the noise spectrum is very close to that of the output of the multiplier. This seems to validate our comparisons made here.

The next figures show the comparison of the frequency spectrum of the message before it enters the circuit, (Figure 4.10) and the output (Figure 4.11) spectrum after the decimator(integrate and dump). The scale difference is the .225 caused by a gain reduction of 4 and bandlimiting of the input signal. Notice that the noise is only noticeable in the higher frequency components which will be eliminated after the $\text{sigmum}()$ function. Figures 4.12 and 4.13 show the noise in the spectrum for the sine input after the decimation stage. Here the noise is shifted to the higher frequencies where in the previous, the noise has been whitened by the spreading pattern.

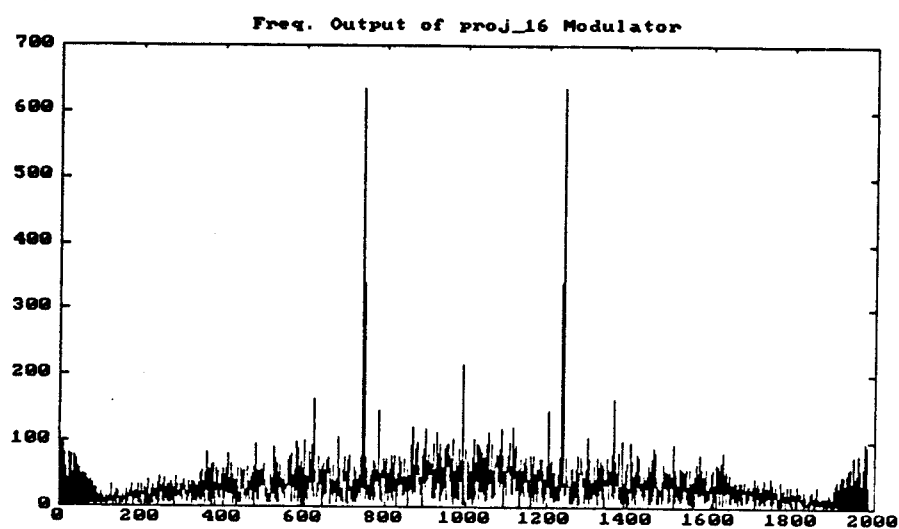


Figure 4.7 - Proj_16 modulator output

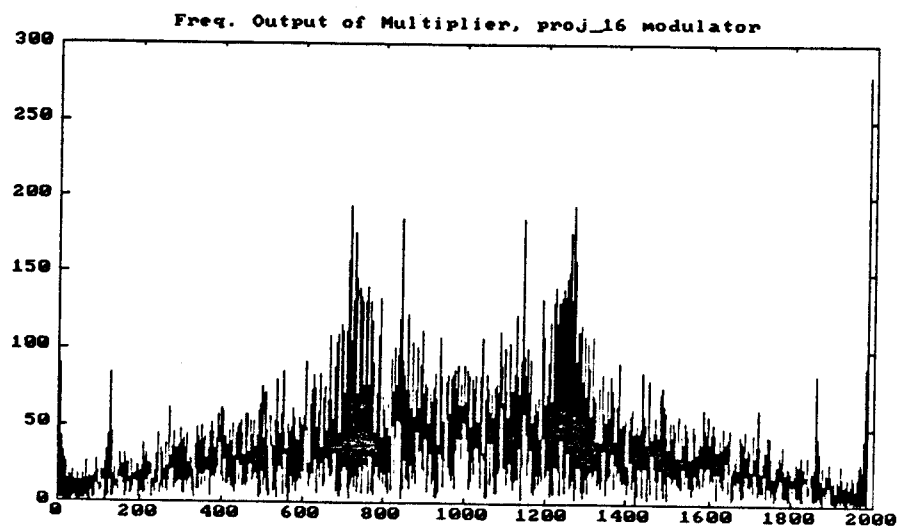


Figure 4.8 - Output of multiplier

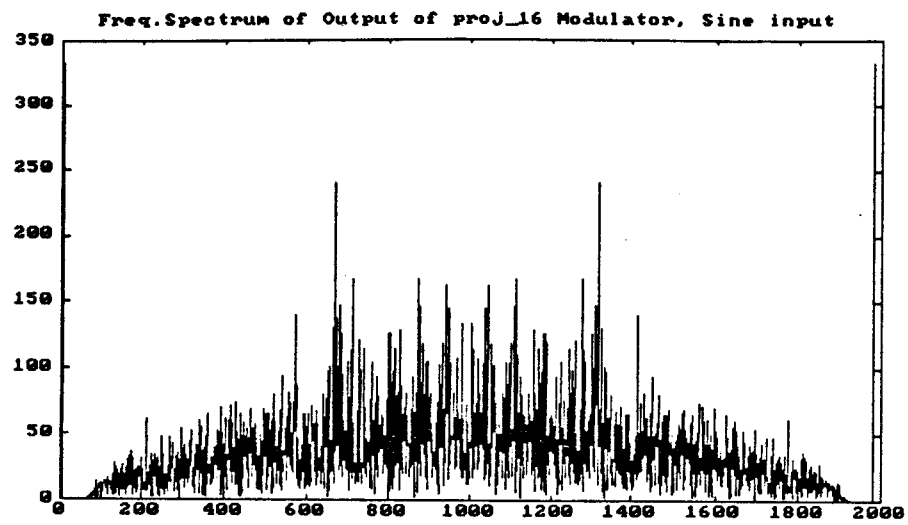


Figure 4.9 - Frequency spectrum with sine wave input

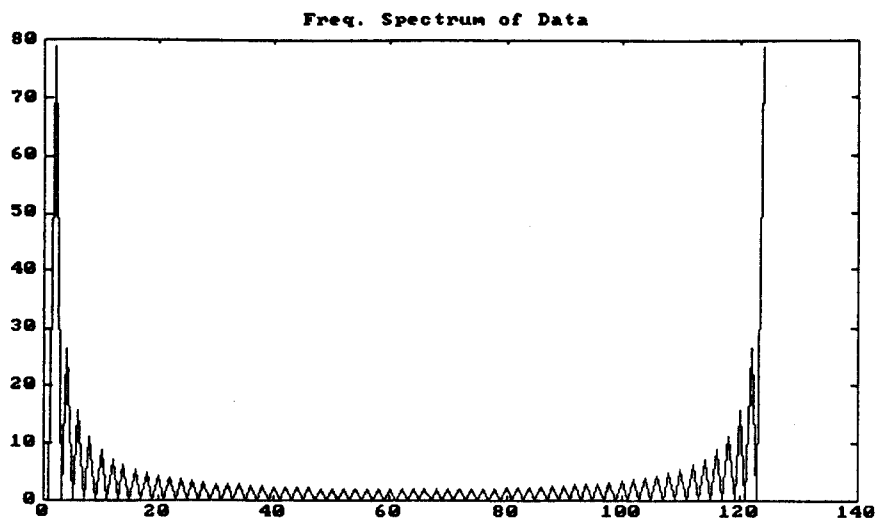


Figure 4.10 - Frequency spectrum of message before modulator

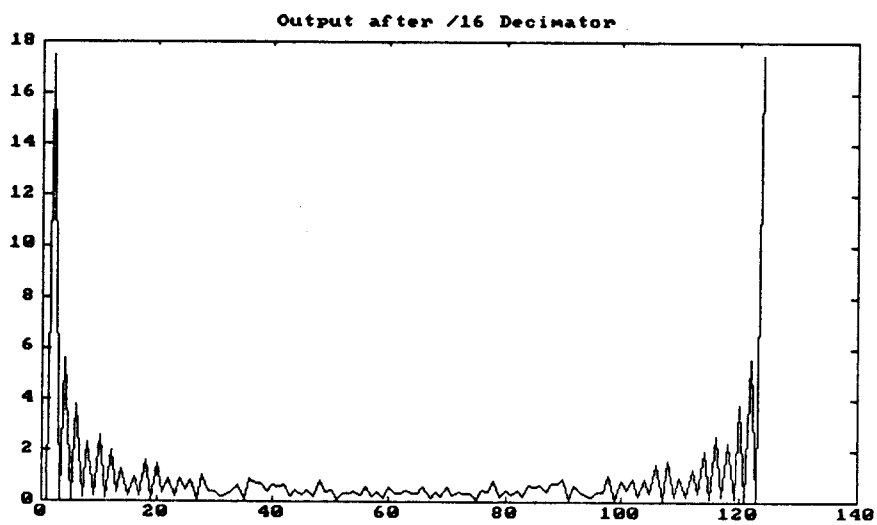


Figure 4.11. Frequency spectrum of message after the modulator

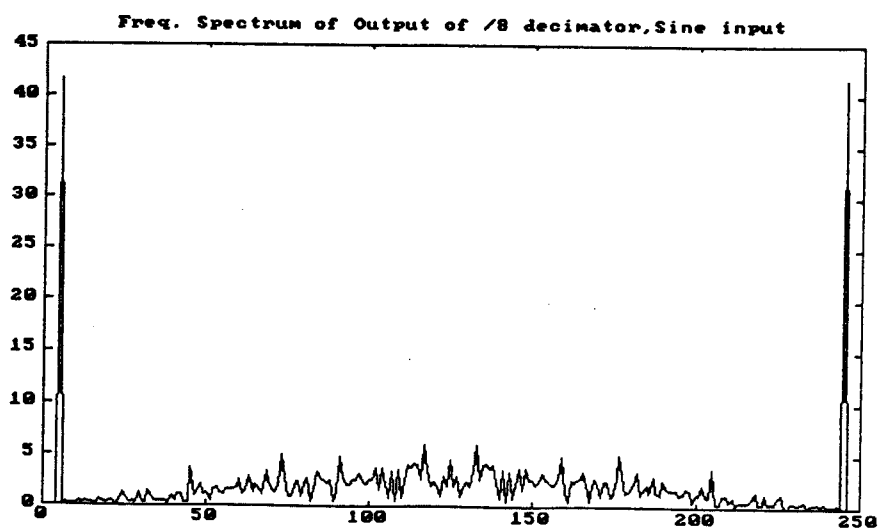


Figure 4.12 - Frequency spectrum for sine wave after modulator and /8 decimator

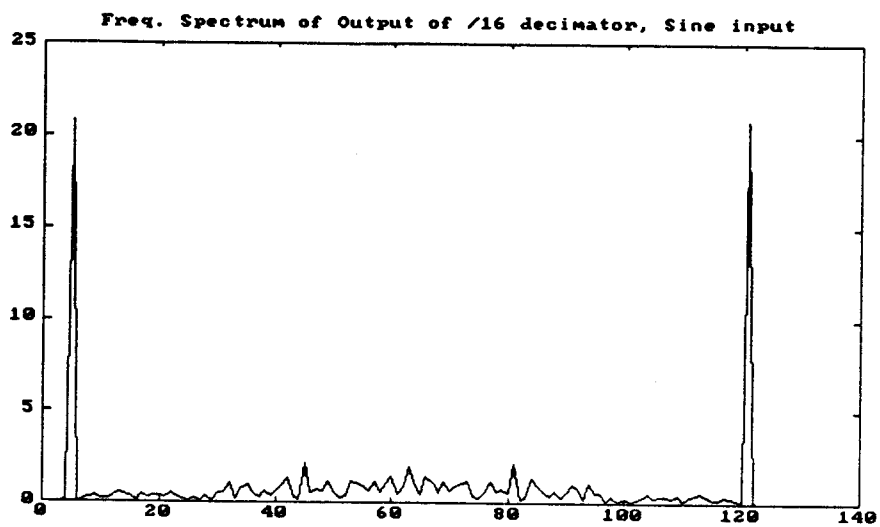


Figure 4.13 - Frequency spectrum for sine wave after modulator and /16 decimator

N x 1 vs. 1 x 1 Multiplication

This section on simulation looks at how complicated the multiplier needs to be in comparison to the sampling rate. Normally, in a direct sequence correlator, the input is multiplied by a single one bit stream of the pseudo-noise generated (PNG) pattern. Since the output of the delta-sigma modulator is bit serial, we have two options. We can multiply the two bit serial streams with a simple exclusive or, or we can chose to change the spreading pattern to a multibit sequence multiplied by the bit serial output of the modulator. The main purpose in going to a multibit pattern, is that we are able to attack other areas such as the near-far problem. It is possible to use a pattern, different from the spreading sequence, that provides good near-far resistance, but may be less optimal for some energy inputs.

The next set of graphs show the correlation, searching in time, between the output of the modulator and other patterns. Previously, it was determined that an oversampling of 8 would provide enough SNR for adequate detection. In trying to acquire initial sync, we would also want to limit the amount of false alarms from the synchronizer (correlator searching in time). Looking first at $R=8$, the correlation of the bandlimited input $r(t)$ with a binary PNG pattern (Figure 4.14) with no delta-sigma modulation shows the typical ideal response. By bandlimiting the PNG pattern and correlating with $r(t)$ (Figure 4.15) it is apparent that nothing is gained. This is why most systems do not modify the decorrelating pattern.

The next set of graphs show what happens if the input $r(t)$ is processed by the delta-sigma modulator and multiplied with the bit serial PNG pattern (1 x 1 multiply, Figure 4.16). We can see that the peaks maintain the proper levels, but the output of

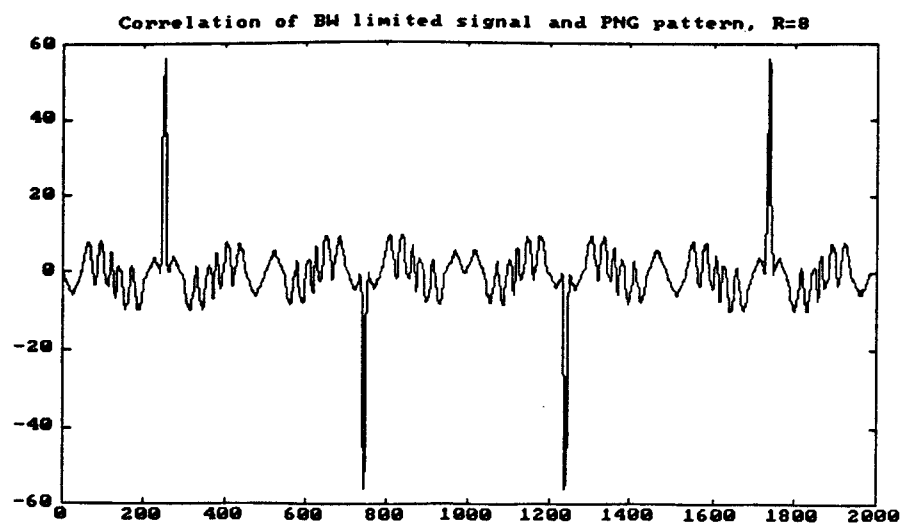


Figure 4.14 - Correlation of bandlimited input with binary PNG pattern

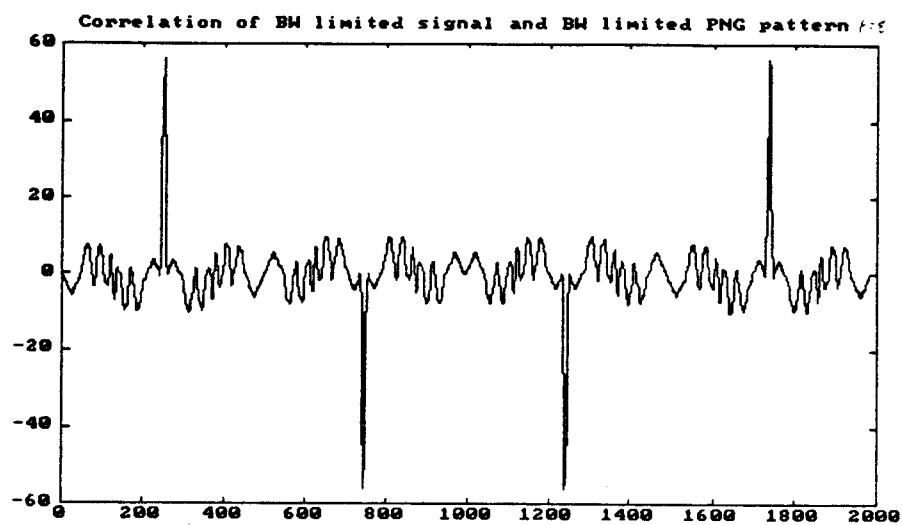


Figure 4.15 - Correlation of bandlimited input with bandlimited PNG pattern

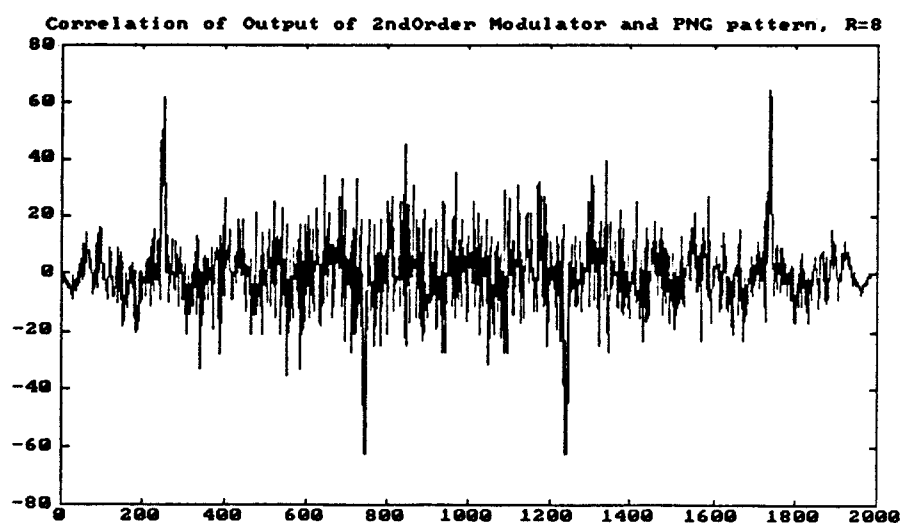


Figure 4.16 - Correlation of input after delta-sigma modulation with binary PNG pattern

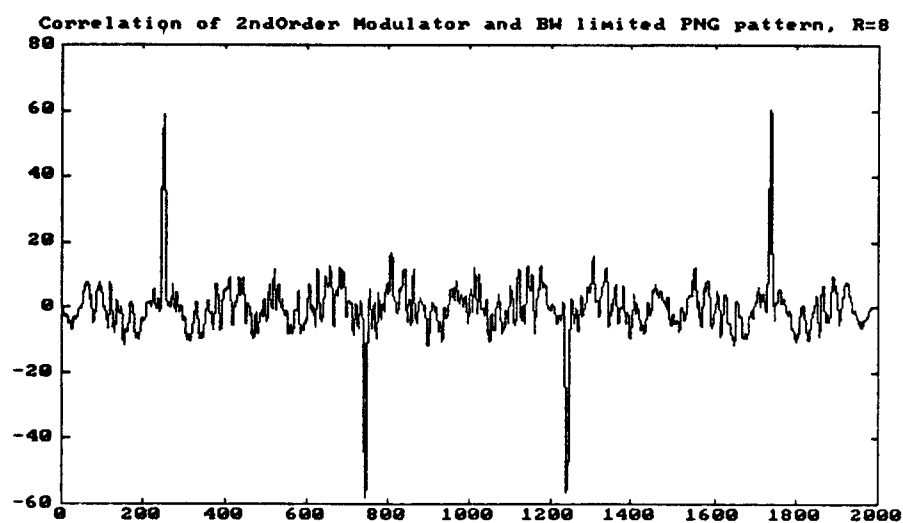


Figure 4.17 - Correlation of bandlimited input after delta-sigma modulator with band limited PNG pattern

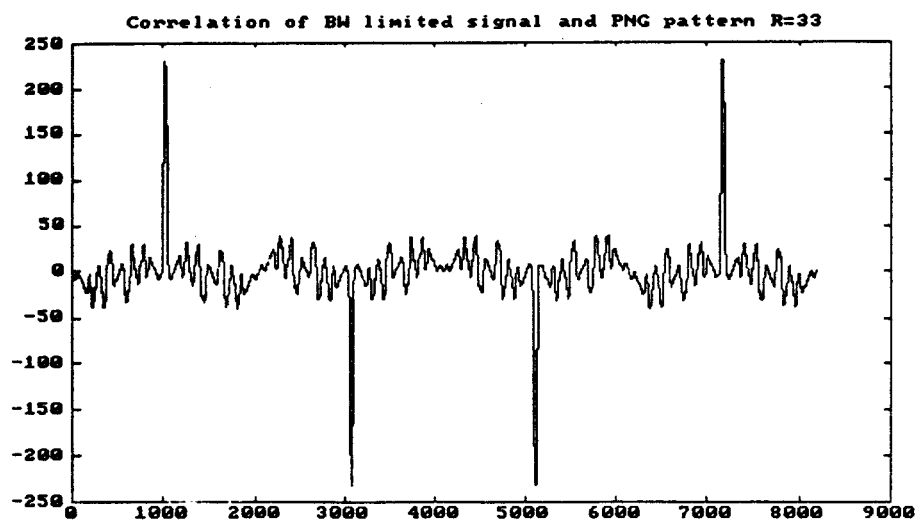


Figure 4.18 - Ideal correlation of input signal with PNG pattern

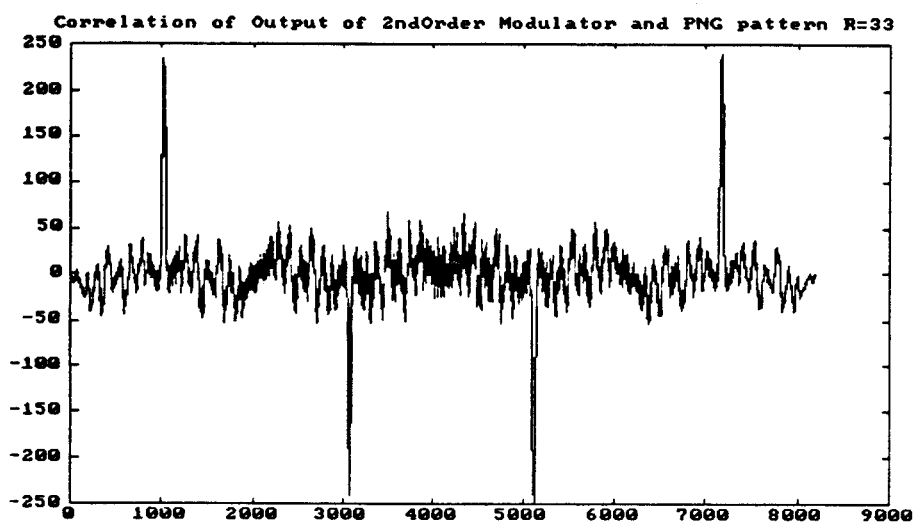


Figure 4.19 - Correlation of input signal after delta-modulation with binary PNG pattern using 1x1 multiplier with higher oversampling

the decimator has high peaks in between the correct ones. By bandlimiting the PNG pattern so that it does not add additional noise when it is being convolved with the delta-sigma bitstream (Figure 4.17), the output of the decimator is very close to the ideal but with some small noise. In order to use an oversampling ratio of 8, the decorrelating pattern should be band limited. This filtering can be achieved in a number of ways which result in an N bits/chip pattern. This N bits could be multiplied by the output of the delta sigma modulator by simply 2's complementing or not 2's complementing the PNG N bit pattern.

If a 1×1 multiplier is still required, one can oversample the signal $r(t)$ by more than 8 and reduce the amount of noise. The next graphs show the correlation outputs for $R=33$. In Figure 4.18, the bandwidth limited input signal is multiplied by the bit serial PNG pattern to show the ideal output achievable. The bottom graph, Figure 4.19, shows the result of multiplying the output of the delta-sigma modulator with the bit serial PNG pattern (1×1 mult.). Now it is seen that the noise is smaller and more spread out, thus providing good SNR for a very low false alarm rate. The designer of the system therefore can trade off complexity for performance.

Review of Musts and Wants

Comparing the earlier musts and wants with the results of the simulations shows that the use of delta-sigma modulation is a good choice. There is a wide range of options in the design process. The chip area should be minimal since only rectangular decimation is required but other decimation techniques could be used to enhance the performance. The architecture lends itself to being expandable which is

required to develop the search in time correlator. Other than selection of the operating frequency, correlating pattern and detection level, no tuning is required. There are design tradeoffs between SNR for bit detection, false alarm rates and oversampling ratios. Most important, the approach allows the use of a non-binary correlation pattern which allows optimization of detection for the near-far problem.

SUMMARY

The main contributions of this paper have been the formulation of a new method to generate the "one-shot" near-far resistant pattern and an architecture by which to implement it using delta-sigma modulation. It was shown that this approach can be easily extended to multiple users. Simulation was performed to determine the resolution required both for the A/D conversion process and the sampling intervals. The sampling rate was shown to be the most critical since any correlation of the received signal $r(t)$ and the near-far resistant orthogonal pattern at other than $\tau=0$, is directly proportional to the resolution of τ . Higher sampling rates can reduce the uncertainty of τ which allow for less complicated $\Delta\Sigma$ modulators. Higher rates, however, will require more memory in order to store the intermediate results. Current technology limits the sampling speed and thus the rate of data transmission possible.

Due to drifts in the carrier frequency because of doppler effects, clock drift, and tracking, the synchronization portion of the system must be carefully designed. Making the synchronizer near-far resistant still remains an open problem due to the difficulty of generating a pattern which is orthogonal to the users at all time intervals. Since the performance of the system will be limited to the synchronizing circuitry, the modified Gram-Schmidt approach presented can reduce the complexity required to implement the GSOSG and still maintain good near-far resistance.

The main purpose in choosing delta-sigma modulation is that the signal can be digitized into a serial bit stream. This action provides for a lossless delay of the signal by using standard digital memory elements. It also can allow reduction in the amount

of complexity required to implement the multiplying circuitry in the correlators.

Since the incoming signal $r(t)$ must be resolved with a high degree of time resolution, the high sampling rates required of $\Delta\Sigma$ modulation works to an advantage in this approach.

BIBLIOGRAPHY

1. R. Lupas, "Near-far resistant linear multi-user detection," Ph.D. dissertation, Princeton University, Jan. 1989.
- ✓ 2. S. Verdu, "Minimum probability of error for asynchronous Gaussian multiple-access channels," *IEEE Trans. Inform. Theory*, vol. IT-32, pp 85-96, Jan. 1986.
- ✓ 3. H.V. Poor and S. Verdu, "Single-user detectors for multiuser channels," *IEEE Trans. Commun.*, vol COM-36, pp. 50-60, Jan. 1988.
4. C.K. Rushforth, Z. Xie, and R.T. Short, "Method and apparatus for decoding multiple bit sequences that are transmitted simultaneously in a single channel," U.S. patent number 4,908,836, March 13, 1990.
- ✓ 5. M.K. Varanasi and B. Aazhang, "Multistage detection in asynchronous code-division multiple-access systems," *IEEE Trans. Commun.*, vol COM-38, pp. 509-519, Apr. 1990.
- ✦ 6. P. Quinton and Y. Robert, **Systolic Algorithms and Architectures**, Prentice Hall, 1991.
7. W. Vetterling, et. all, **Numerical Recipes in C**, Cambridge University Press, 1988.
8. H.T. Kung, "Why systolic architectures?," *Computer*, Jan. 1982 QA 76.73.C1S N861 1982
9. M.W. Hauser, "Principles of oversampling A/D conversion", *J. Audio Eng. Soc.*, vol 39, no. 1/2, Jan-Feb, 1991.
- ✓ 10. R.C. Dixon, **Spread Spectrum Systems**, 2nd Ed., Wiley, 1984.
- ✦ 11. G.R. Cooper and C.D. McGillem, **Modern Communications and Spread Spectrum**, McGraw Hill, 1986.
- ✓ 12. S. Verdu, "Optimum multiuser asymptotic efficiency," *IEEE Trans. on Commun.*, vol. COM-34, no. 9, Sept. 1986.

13. R. Lupas and S. Verdu, "Near-far resistance of multiuser detectors in asynchronous channels," *IEEE Trans. Commun.*, vol. COM-38, pp 496-508, April 1990.
14. J.C. Candy and G.C. Temes, **Oversampling delta-sigma data converters : theory, design and simulation**, *IEEE Press*, 1992.
15. B.W. Dickinson, **Systems : analysis, design and computation**, Prentice Hall, 1991 0A7619.588 3571091
16. M.K. Varansi and B. Aazhang, "Optimally near-far resistant multiuser detection in differentially coherent synchronous channels," *IEEE Trans. Info. Theory*, vol. 37, no. 4, July 1991.
17. R. Schreier, Class lecture notes, ECE619, Fall 91, Oregon State University.

APPENDIX

APPENDIX A SECTION 1

Introduction

This section investigates the systolic implementation of the "One-Shot Decorrelating Detector[1]". This circuit is used in code division multiple access(CDMA) spread spectrum communication to separate out multiple messages which occupy the same frequency spectrum. In [1] the one-shot was shown to be an easier to compute alternative to the optimum near-far decorrelating detector. The main benefits of such a detector are its memorylessness, linear complexity, and performance which is relatively independent of the energies of interfering users which provides its near-far resistance[1]. Since the data needs to be processed in real time at high speeds, the systolic computation approach appears to be the most plausible since it leads readily to implementation in a VLSI structure.

This appendix firsts looks at the 2 user case, both from an ad-hoc approach and then from a systolic synthesis method which allows extension to N users. Some different alternatives are examined and their features are compared in relation to number of processors, timing and signal flow.

One-Shot Decorrelating Theory of Operation

The message received by a 2 user CDMA consists of a combination of two signals which are modulated by a spreading code $s_i(t)$. The combined received signal is

$$\begin{aligned}
 r(t) &= b_1 m_1(t) s_1(t+T-\tau_1) + b_2 m_2(t) s_2(t+T-\tau_2) \\
 m_1(t) &= \text{data message of user}_1 \\
 m_2(t) &= \text{data message of user}_2 \\
 s_1(t) &= \text{user}_1 \text{ spreading code} \\
 s_2(t) &= \text{user}_2 \text{ spreading code} \\
 b_1 &= \text{energy of user}_1 \quad b_2 = \text{energy of user}_2
 \end{aligned}$$

For the case of the one-shot, the idea is to restrict attention to one bit interval at a time. This requires that we look at each interfering user as consisting of two consecutive bits which overlap with the bit of the user we are trying to decode. Thus in the two user case, we split the second users spreading code into two distinct patterns so that the received waveforms are now $\{s_1(t), s_2^L(t), s_2^R(t), i = 2, \dots, K\}$, where

$$\begin{aligned}
 s_2^L(t) &= \begin{cases} s_2(t+T-|\tau_2-\tau_1|), & 0 \leq t \leq |\tau_2-\tau_1| \\ 0, & |\tau_2-\tau_1| \leq t \leq T \end{cases} \\
 s_2^R(t) &= \begin{cases} 0, & 0 \leq t \leq |\tau_2-\tau_1| \\ s_2(t+T-|\tau_2-\tau_1|), & |\tau_2-\tau_1| \leq t \leq T \end{cases}
 \end{aligned}$$

From the above waveforms, the one shot correlation matrix has the form

$$\mathbf{R} = \begin{pmatrix} 1 & \rho_{21} & \rho_{12} \\ \rho_{21} & e_2 & 0 \\ \rho_{12} & 0 & 1-e_2 \end{pmatrix}$$

where e_2 is the energy in waveform s_2^L . From [1], the new pattern which is correlated with $r(t)$ to decode user₁ is $\mathbf{R}_{11}^{-1}*[s_1 \ s_2^L \ s_2^R]^T$ which is

$$[s_1(t) - \frac{\rho_{21}}{e_2} s_2^L - \frac{\rho_{12}}{(1-e_2)} s_2^R]$$

Block Diagram of Systolic One-Shot

Figure A1 shows the block diagram used for the approaches taken in this section in trying to implement the systolic one-shot. The received signal $r(t)$ must first pass through two correlators which determine the proper sync signals for the two messages. The systolic implementation of these correlators is examined in the synthesis portion of this appendix. The sync signals from the correlators then start the generation of the pseudo-noise pattern generators which are matched to the pattern used to spread the respective messages. The output of these pattern generators are then used to determine the correlation and energies of the time intervals shown in Figure A2. This processing is done in the block labeled "one-shot systolic array proc.". The FIFO (first in, first out) blocks are used to delay the received signal for the time required to compute the new pattern. The new pattern is then correlated with the delayed $r(t)$ in the final correlator and detector which outputs the original message $m_i(t)$. In this block diagram it is assumed that the analog signal $r(t)$ is at baseband and has been digitized by a delta-sigma A/D into a serial bit stream[2] in the systolic synchronizer correlator. Therefore all signals in the block diagram are digital levels, thus allowing the use of standard VLSI digital cells.

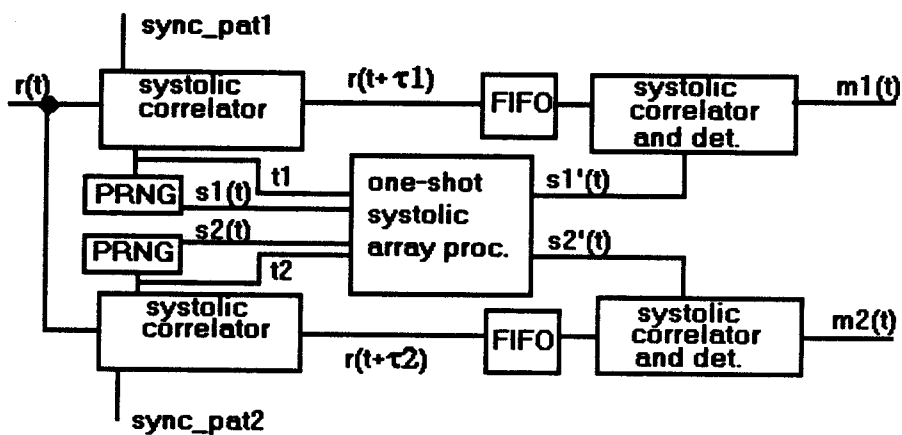


Figure A1 - Block diagram for systolic one-shot decorrelating detector

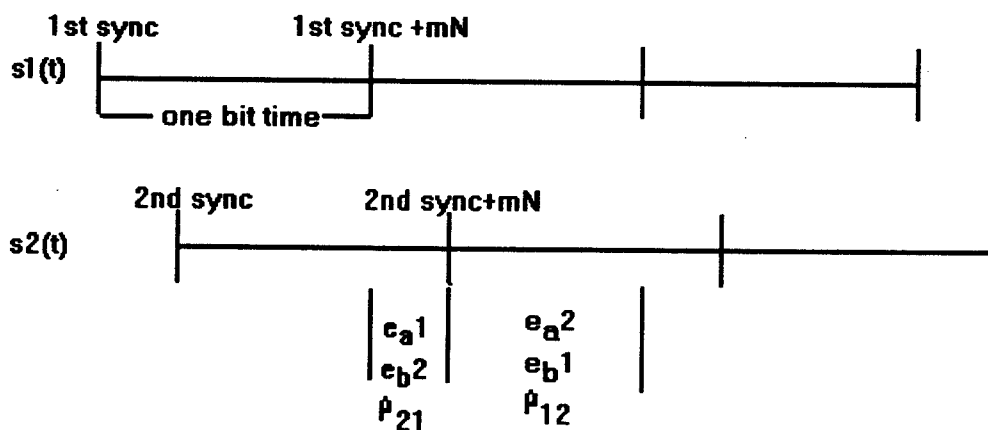


Figure A2 - Timing for one-shot decorrelating detector

Ad-Hoc Approach

From the timing, the following procedure can be used to determine the values of the correlation matrix and then computation of the final pattern.

Procedure:

- 0) initialize circuit
- 1) wait for the first sync pulse

For (m=0;inf;m++) {

- 2) From the 1st sync to the 2nd sync:

- a) $e_a1 = \sum s^{L_1^2}(t)$
- b) $e_b2 = \sum s^{L_2^2}(t)$
- c) $\rho_{21} = \sum s^{L_1}(t)s^{L_2}(t)$

- 4) at 2nd sync + mN (N=number of clock periods in one bit time)

$$a) s'_2(t) = s_2(t) - (\rho_{21}/e_a1)s^{L_1}(t) - (\rho_{12}/e_a2)s^{R_1}(t)$$

- 3) From 2nd sync to 1st sync+mN

- a) $e_a2 = \sum s^{R_1^2}(t)$
- b) $e_b1 = \sum s^{R_2^2}(t)$
- c) $\rho_{12} = \sum s^{R_1}(t)s^{R_2}(t)$

- 4) at 1st sync + mN

$$a) s'_1(t) = s_1(t) - (\rho_{21}/e_b2)s^{L_2}(t) - (\rho_{12}/e_b1)s^{R_2}(t)$$

}

The processor layout diagram for the above equation is shown in Figure A3. There are nine processing units required. The far left multiply and accumulate processors must operate at the chip clock rate. At any sync pulse, the accumulators are output to the register pipeline and reset. The registers contents then move to the

divide processor which divides the cross correlation by the energy in the signal. The output of the divisor is then latched and it is multiplied by the pattern of the other user and subtracted from the users pattern. While the diagram shows the original pseudo noise patterns being delayed by a bank of shift registers, the pattern could be replicated by a delayed clock as well. This would be more economical in VLSI real-estate if the amount of time required for the divider is long.

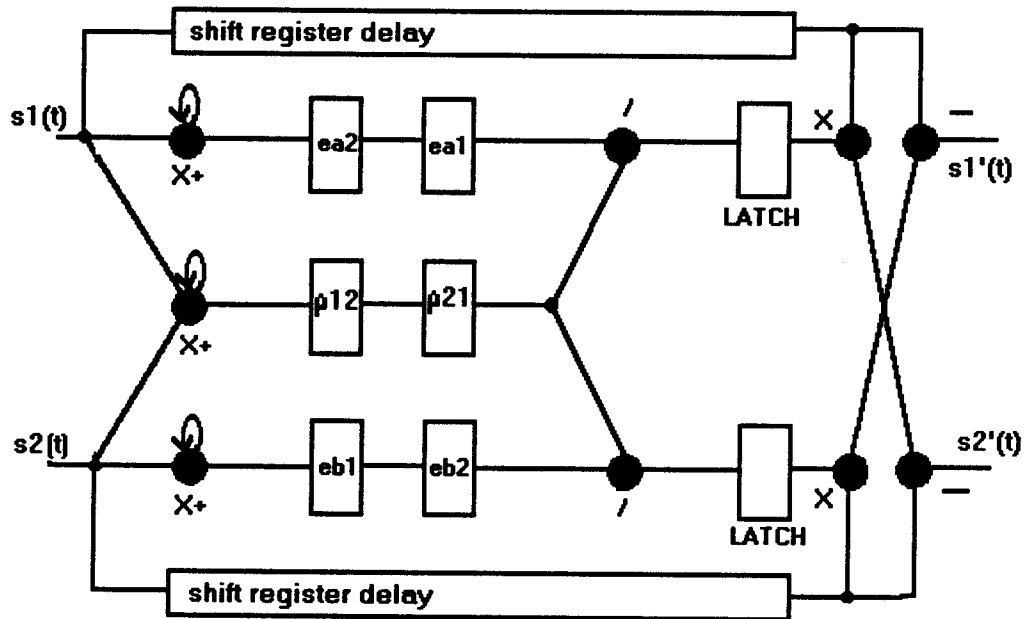


Figure A3 - Processor layout diagram for one-shot systolic array with two users

Since the two user case's formulae were relatively simple, it was fairly easy to derive a systolic version of it. When the number of users are increased, it is very difficult to use an adhoc approach. Therefore the next section tries to come up with recurrence equations and architectures that can be easily expanded.

SECTION 2

Systolic Synthesis of One-Shot

The difficult areas in extending the architecture to N users is in the generation of the correlation matrix and inverting it. Another concern is the size of the synchronizing correlators since one is required for each user. Each of these areas are examined in this section. In order to make the circuits realizable, the minimal number of processors used is considered more optimal than other possible solutions.

Correlation Matrix

The correlation matrix can be derived by letting the rows of a matrix S contain the pseudo-noise patterns of each of the users and multiplying S by S^T . S is a $n \times m$ matrix where n is the number of users and m is the number of chips (or smaller time intervals if using delta-sigma modulation) per bit. After S is multiplied by its transposed, the result is a symmetric matrix of size $n \times n$. Therefore only the upper or lower triangular portion of the matrix needs to be computed. The uniform recurrence equations are derived as:

$$R_{ij} = \sum_{k=1}^n s_{ik} s'_{kj} \quad | \quad i = \{1, \dots, N\}, j = \{1, \dots, M\}, s' = s^T$$

FOR $i = 1$ to N

FOR $j = i$ to N

FOR $k = 1$ to M

- 1) $s(i, j, k) = s(i, j-1, k)$
- 2) $s'(i, j, k) = s'(i-1, j, k) \quad | \quad j \geq i$
- 3) $R(i, j, k) = R(i, j, k-1) + s(i, j, k) s'(i, j, k)$

initial conditions:

$$s(i, 0, k) = s(i, j)$$

$$s'(0, j, k) = s'(i, j)$$

$$R(i, j, 0) = 0$$

Final output:

$$R(i, j) = R(i, j, M+1)$$

All the variables are localized. If we take advantage of the fact that $s'_{ij} = s_{ji}$, and change the direction which s' propagates in the DG then the following DG emerges:

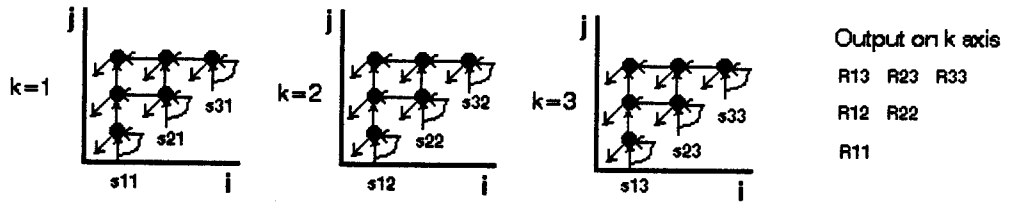


Figure A4 - Dependency graph for correlation matrix

After changing the direction on s' , equation 2 becomes $s'(i, j, k) = s'(i+1, j, k)$.

Now the final array in the space time domain can be derived from the above D.G.

Since the correlation coefficients will be used in the next sub-block of the design, the allocation vector should be orthogonal to R_{ij} . It should also be orthogonal to s_{ij} since it needs to be input to the array. Two are presented here and compared.

The first array sets $\bar{d} = [-100]^T$ and $\bar{s} = [-111]^T$. The final allocation matrix is

$$\lambda = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 1 & 1 \end{bmatrix}$$

The processor layout diagram is shown in Figure A5 for $N, M=3$. The number of processors required is NM . Total throughput time is $M+N$ clock cycles with 1 clock delay between successive outputs for a pipeline rate of 1.

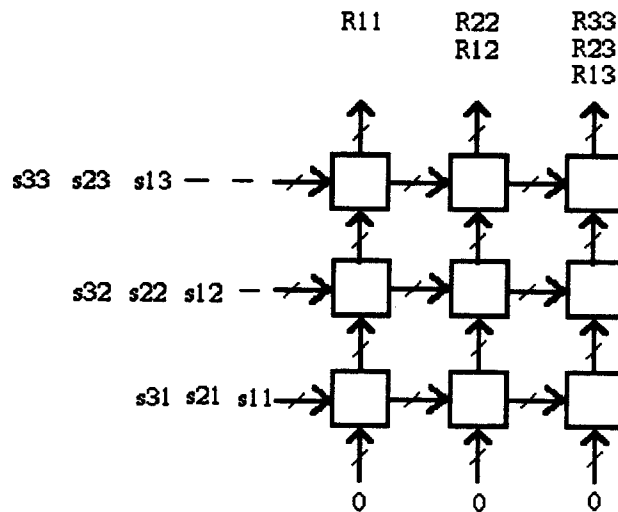


Figure A5 - Processor layout diagram for correlation matrix

The above implementation has the nice feature that the correlation coefficients leave the array at one boundary of the array. It has the disadvantage in that it has many more processors than if we leave the R_{ij} 's in the computation cell. The results will then have to be output via a separate bus for each column of processors. This smaller processor array is formed by choosing $\bar{d} = [0 \ 0 \ 1]^T$ and $\bar{s} = [-1 \ 1 \ 1]^T$.

The allocation function is then $\lambda = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 1 & 1 \end{bmatrix}$. The processor layout graph is shown in Figure A6.

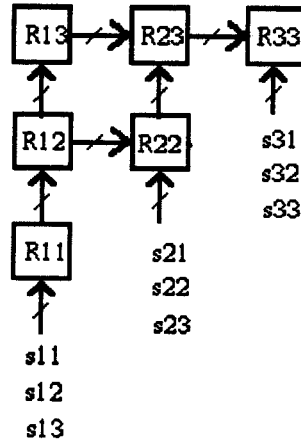


Figure A6 - Processor layout diagram for correlation matrix with fewer processors

This allocation provides for approximately $N^2/2$ processing elements with a total computation time of $M+N$ and a pipeline rate of 1.

Correlation Matrix Inversion

This is the area where there is the most difficulty. It may be possible to exploit the symmetry of the correlation matrix by using the fact that its inversion is also symmetrical. This would mean that only the upper or lower triangular portion of the inverse would need to be computed. The first row would consist of the minors of each element and the determinant could easily be found by multiplying the first rows

new elements by the previous first row elements and summing them. A good solution extendable to N users is found in [6] and [7]. This approach is known as the Gauss-Jordan Elimination.

The Jordan diagonalization method is described in section 4.3 of (6) and section 2.1 of (7). This method can compute the inverse of R in $5N-2$ clock cycles and requires $N(N+1)$ processors.

One problem with the inversion of R deals with its stability. If the delay between users is small or zero, the row in R for that sub-interval is nearly zero or is zero. This will not allow the Gauss-Jordan method find the inverse of R . In implementation, the inversion block should have the ability to invert a variable N size R matrix which would have the zero rows removed.

Correlators

This portion describes the systolic implementation of the correlators in the block diagram. The one-shot needs two types of correlators. The first correlator must search in time trying to match the incoming signals with a sync pattern. The second correlator has a much easier job (since sync with input is known) as it is only required to multiply its inputs and sum up M samples of them, dump the results and reset. Since this second block has been evaluated in chapter 4, its function is not described in this portion.

The search in time correlator equations are basically the same as convolution except in this case it always is assumed that the summation encompasses the entire length of the bit sequence. In practice, most correlators have used the approach in Figure 5. of [8] where the pattern is stored in the cells, the input moves systolically and are multiplied by the pattern weights and the output is formed by fanning in the results from all the cells. This approach was used since the multiply was only of two bits and the summation was performed using currents. This resulted in a simple architecture that was easily implemented. However, to make the synchronizer near-far resistant, we need to multiply the incoming signal by a multi-bit unit in each multiplier. By using a systolic implementation we can remove the global interconnections and perform the summing digitally. It would help if both the pattern, incoming signal and output are moving through the cells.

A promising architecture is shown in Figure A7. In this mapping, the outputs stay in the processing cells but since they are done at different time intervals, the data can be transmitted out on a bus that is parallel to the array. The architecture has a latency of $2N-1$ and a pipeline rate of 1. It requires N cells (N being the length of the sync pattern). It is shown in chapter 4 that since the incoming data is a one bit stream from a delta-sigma A/D, the multiplication in the cells could be performed by simply 2's complementing or not the sync pattern which is a multi-bit data stream. The summation is a simple multi-bit digital adder. Since each processor is only used on every other timing cycle, two adjacent processors could be combined to share the same adder, with the addition of temporary holding registers and some control logic. This technique allows for longer sync patterns by compacting the array.

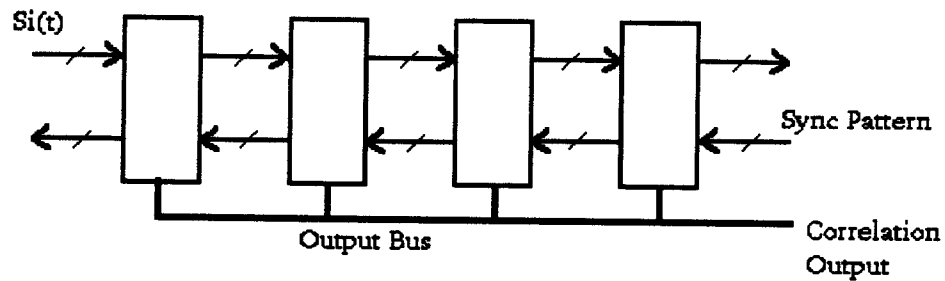


Figure A7 - Mapping of search in time correlator

Appendix Summary

The results of this investigation shows that the One-Shot detector can be implemented using a systolic approach. The synthesis methods provide a far easier tool to work with than an ad-hoc type approach. The complexity in terms of the number of processors, however, is proportional to N^2 . This is due to the generation of the correlation matrix and its inversion. Also in order to limit the number of processing elements, not all data flows systolically, e.g. some must stay in the P.E.s. This requires that there be busses and additional control circuitry to unload the P.E.s. It was shown that some reduction of P.E.s are possible by exploiting the symmetry in the correlation matrix and by grouping adjacent cells in the correlators. Further work may be able to exploit the symmetry in the inversion of R and reduce the complexity. Further study should also focus on how to reduce the number of P.E.s so that it is proportional to N and how to interconnect the processing sub-blocks.