AD-765 327

IMAGE TRANSMISSION VIA SPREAD SPECTRUM TECHNIQUES

Robert W. Means, et al

Naval Undersea Center

Prepared for:

Advanced Research Projects Agency

1 June 1973

IMAGE TRANSMISSION
VIA
SPREAD SPECTRUM TECHNIQUES

ARPA Quarterly Technical Report

March 1, 1973 – June 1, 1973

Naval Undersea Center
San Diego, California 92132

IMAGE TRANSMISSION
VIA
SPREAD SPECTRUM TECHNIQUES

Investigated by
DR. ROBERT W. MEANS (714.225-6872), Code 608
and
HARPER J. WHITEHOUSE (714.225-6314), Code 6003
Naval Undersea Center
San Diego, California

Sponsored by
Advanced Research Projects Agency
Order Number 2303
Code Number 3G10
Contract
effective: 15 February 1973
expiration: 30 June 1973
amount: $125,000

ARPA Quarterly Technical Report
March 1, 1973 – June 1, 1973

Form Approved Budget Bureau No. 22-R0293

Contents

1. INTRODUCTION

The transmission of standard commercial television images from a remotely piloted vehicle (RPV) to a distant observer requires a large bandwidth channel in order to protect the video information against jamming and interference. This protection can be gained from spread spectrum techniques. In many RPV situations, however, the available channel bandwidth is barely sufficient or is insufficient to transmit the television signal directly. In these cases the video information can be transmitted only if the redundancy present in the original images is greatly reduced. This report discusses the feasibility of redundancy removal at real time rates with small, lightweight, low power hardware suitable for the ARPA RPV transmission. It is desirable to be able to vary the amount of redundancy removed to permit the observer to select from the available tradeoffs of resolution, bandwidth, scan rate, and jamming environment.

Image redundancy reduction by a factor of ten without serious image degradation has been achieved by the use of linear transformations and filtering in the transform domain. Such techniques are the only methods presently known to obtain such large redundancy reduction factors. They are also the only methods known which will allow the redundancy reduction ratio versus degradation to be continuously varied. Haar, Hadamard, Fourier, and Karhunen-Loève transforms have all been successfully used in this way, but the processing has been accomplished on large digital computers at much slower than real time rates. Examples of these redundancy reduced photographs are given in Appendix A. The feasibility that the required transforms can be performed with a high throughput processor structure which utilizes parallel access to a serially shifting data stream is explored. The processor structure may be implemented by small, lightweight, low-power hardware using transversal filters. Other methods of television bandwidth reduction such as slow frame rate and image sensor improvements are also presented.

2. VIDEO DATA SPECIFICATIONS

PSYCHOPHYSICAL STUDIES

The amount of information that an operator needs to effectively operate an RPV for various missions is not a well-defined quantity. Rather it is a subject of some controversy in the literature[1]. The Aerospace Medical Research Laboratory (AMRL) at Wright Patterson Air Force Base has been conducting such psychophysical experiments for an RPV mission. The speed and altitude of the Wright Patterson RPV's mission differ from the speed and altitude of the ARPA RPV. The Wright Patterson studies may therefore have limited applicability to the ARPA RPV mission.

However, Wright Patterson has sponsored work at Virginia Polytechnic Institute and State University in order to determine a more general measure of video image quality. This work has met with some success and has been reported in conference proceedings[2] and in a final report to the Aerospace Medical Research Laboratory. At this time it is undergoing review by AMRL and by USC.

In the absence of definitive results there are several tentative conclusions that have been reached. Since the ARPA RPV has a slow speed, low altitude mission, and since the human operator response time is approximately one-quarter of a second, the information rate (i.e., frame rate for television) need only be approximately four frames per second. Again, since the ARPA RPV has a relatively slow speed, the time delay of the information for control and reconnaissance is not of critical importance. However, it is important in the target designator mission and should always be less than the human response time, preferably much less.

The image quality necessary is more difficult to estimate. The resolution of home quality television is the best available estimate at this time. The Virginia Polytechnic study should answer this in more detail. The camera itself has a very important role and the vidicon is not the best camera for RPV missions. The Navy, through NAVELEX, is developing charge-coupled device cameras which will offer a better alternative to the vidicon.

The value that color will provide to an RPV operator is also a subject of controversy. Studies have been made with contradictory results. This subject will continue to be investigated. The University of Southern California has performed independent studies in this area. Their work will be detailed in a later report. If it is determined that color is necessary, then the primary implementation difficulty will be in the color camera.

BANDWIDTH COMPRESSION

The amount of bandwidth required to digitally transmit television images is determined by the bit rate. The television bit rate is given by

$$B.R. \equiv N*B*F ,$$

where N is the number of picture elements (pixels) per video frame, B is the number of quantization bits per picture element, and F is the frame rate. The number of pixels per frame is a measure of resolution. The number of bits per pixel is a measure of the dynamic range of the sensor and the display. The frame rate for broadcast television is determined by the avoidance of flicker in the display. Broadcast television has set this at thirty frames per second.

Let us consider a channel over which the digitized television is to be transmitted. Let the channel bit rate be a given quantity. It is, in most applications, a number fixed by considerations other than video bandwidth. It is determined in part by assumptions of white noise in a given bandwidth and a specified bit error rate which implies a given signal-to-noise ratio at the receiver.

A relative compression ratio (in dB) can then be defined for use in comparing various bandwidth compression schemes as

$$C.R. = 10 \log \left(\frac{\text{channel bit rate}}{N*B*F} \right)$$

United States broadcast television has thirty frames per second, 525 lines per frame with 480 useable lines and a 4:3 aspect ratio. Studio quality TV is stored digitally with 6 bits per picture element, 640 picture elements per line, and 480 lines per frame. The data rate for studio quality television is 55.3 megabits per second. Let us postulate a 20.0 megabit channel data rate. The relative compression ratio for conventional studio quality television is -4.4 dB; that is, the channel is inadequate. Home quality television does not have the resolution of studio quality. The University of Southern California has determined that a sampled image of 256 by 256 with 6 bits per pixel is comparable to home quality television. This has a data rate of 11.8 megabits per second. The C.R. for home quality television for the 20.0 megabit channel is 2.3 db.

Computer studies in picture transform encoding have shown that the transforms allow the average number of bits to be reduced to approximately one bit per picture element without picture degradation. Optimum transform encoding has reduced this to one-half bit per pixel. Let us consider the C.R. available for a 256 by 256 frame at 4 frames per second with a transform encoder

capable of preserving picture quality at one bit per picture element. The data rate is 262 kilobits per second and the C.R. for a 20 megabit channel is 18.8 db.

If it is determined that the resolution of home television is adequate for RPV missions, then the linear transform frame encoding plus a reduced frame rate will give a relative compression ratio of 18.8 db. The feasibility of a real time hardware implementation of this bandwidth reduction will be discussed in section III.

The amount of AJ available on the redundancy reduced data will depend on the subsequent coding algorithms used to encode the data. A coding scheme is being investigated that offers an additional 10 db of AJ. A computer emulation of the system has been programmed which will allow the testing of this and other proposals. This emulation is described in Appendix B.

Signal reconstruction by an inverse transform at the ground station is straightforward. Additional processing may enhance the picture. One such procedure is described in Appendix C. Reconstruction enhancement will be investigated more fully in the next phase of the program. Appendix A contains a report by Habibi and Hershel which discusses a general reconstruction method.

3. TRANSFORM ENCODING

INTRODUCTION

The use of unitary transforms for image encoding has been evaluated for use as intraframe encoding techniques[3]. In addition, these techniques may also be applied to interframe and multispectral encoding. However, all unitary transformations are information preserving and no bandwidth reduction results from the application of the transform to the image. Instead, the transforms redistribute the variance associated with each picture element (pixel); so that subsequent to the transform, basis restricting operations on the transform coefficients will result in bandwidth reduction. Upon reconstruction of the original image from the basis restricted transform coefficients, a degraded version of the original image can be obtained. Unfortunately, the interrelationship between the type of transform, the form of the noninvertible operation, and the type of degradation in the reconstructed image is very complicated and subjective. The universally used analytic criterion of the mean-square-error is, at present, the best technique for transform comparison.

For the particular operation of basis restriction by truncation, a particularly simple interpretation of the bandwidth reduction can be made. The transforms may be viewed as a variance redistributing operation that approximately decorrelates the transform coefficients while transforming the variance associated with each picture element into the low-order coefficients of the transform. Under the assumption that each set of picture elements can be considered as a sample function from a wide sense stationary random process with correlation function $r^{|t|}$, there exists an optimum discrete transformation, the Karhunen-Loeve transformation, which totally decorrelates the transform coefficients and maximally compacts the variance to the low-order coefficients. All other transformations can be compared in their performance by comparing their transform coefficient decorrelation and variance compaction with this optimum transformation.

This intuitive interpretation can be made rigorous through the use of the rate-distortion criterion[4]. It has been found from experience that the closer the eigenvectors of the transformation approximate the eigenvectors of the optimum Karhunen-Loève transformation, the greater the variance compaction will be and the more the coefficients can be truncated while maintaining a fixed rate distortion or mean-square-error.

The use of two-dimension transforms can provide improved performance over the use of transformations on a line-by-line basis[5]. The most direct approach is to seek a two-dimensional transform which simultaneously decorrelates the transform coefficients and compacts the variance into a corner of the two-dimensional transform coefficient space. One method is to find a two-dimensional transform which can be represented as the product of a transform in one direction and a transform in the other direction. Assuming that a two-dimensional picture can be considered as a sample function from a random process with two-dimensional correlation $r_1^{|\tau_1|} r_2^{|\tau_2|}$ i.e., with a correlation coefficient r_1 in direction one and a correlation coefficient r_2 in direction two, then the optimum discrete transformation is the successive use of two Karhunen-Loève transformations; the first with parameter r_1, and the second with parameter r_2.

Another variable of interest in transform encoding is block size. For a one-dimensional signal the block size is the number of elements of the transform and the performance of the transform improves monotonically with increasing block size. For two-dimensional images, transform performance also increases with increased number of elements in each dimension of the transform. However, two dimensional transforms usually require intermediate memory to store the transform coefficients in the first direction while the transform is being computed in the second direction.

Also of interest in two-dimensional transform encoding are mixed transforms, e.g., one horizontal transform and one vertical transform. Although performance increases with the number in each direction of elements in the transform, performance varies with the particular transform chosen. However, memory requirements tend to increase linearly with the number of elements in the second transform direction since all of the coefficients must be stored from the first transform. The amount of intermediate memory may be minimized by the use of a small block size for the image in the second direction, but performance may not be optimized by this choice. The choice of a mixed transform thus interacts with the overall system design and the available memory for coefficient storage.

KARHUNEN-LOÈVE TRANSFORMATION

If a continuous time function of zero mean and autocorrelation function $r_\tau = e^{-\alpha|\tau|}$ is considered to be a sample function from a wide-sense stationary random process, then this time function can be expanded by the Karhunen-Loève expansion[6] and the resulting coefficients will be uncorrelated. For a discrete function of zero mean and autocorrelation function $R_\tau = r^{|\tau|}$, which may be

considered as a sample function from a first-order Markov process, a similar discrete Karhunen-Loève transformation may be defined[7]. This transformation diagonalizes the covariance matrix and is optimal in the mean-square error sense for a restricted set of basic functions that do not span the complete space.

The discrete Karhunen-Loève expansion is given by [7] for the case N = 2m as

$$G(k) = \sum_{n=1}^{2m} \frac{2}{2m + \lambda_n^2} g(n) \sin \left\{ \omega_n [k - (2m + 1)/2] + n\pi/2 \right\} \qquad k = 1, 2, \ldots, 2m$$

where

$$\lambda_n^2 = \frac{1 - r^2}{1 - 2r \cos \omega_n + r^2}$$

and $\{\omega_n\}$ are the positive roots of

$$\tan 2m\omega = \frac{-(1 - r^2) \sin \omega}{(\cos \omega - 2r + r^2 \cos \omega)}$$

Since the discrete Karhunen-Loève expansion involves both the solution of a transcendental equation and the evaluation of the autocorrelation function of the data to be transformed, real time computation of this transform is quite complex. However, Habibi and Wintz [3] have shown that Karhunen-Loève transformations using approximate autocorrelation functions are satisfactory for many applications. In addition, Pratt at USC is examining the use of corrected Fourier coefficients as a practical way of computing approximate Karhunen-Loève expansions.

The preceding considerations suggest that the Karhunen-Loève expansion should be used only for small block sizes and that under these conditions either precomputed expansions or interpolation may be used. This complexity may be justified for multispectral encoding or for reduction of the memory required for a mixed transform.

DISCRETE FOURIER TRANSFORM

Since the discrete Fourier transform is asymptotic to the Karhunen-Loève transformation[8], even though the basis vectors are picture independent, the Fourier transform represents a logical

choice for real time implementation. The Fourier transform exists for all lengths N. The basis vectors are complex and are given by

$$\phi_k^n = e^{-j2\pi nk/N}$$

If the input sequence is real, then the Fourier coefficients will be conjugate even. If the input sequence is real and even then the Fourier coefficients will be real and even. If the input sequence is non-negative and symmetric, then the Fourier coefficients will be an autocorrelation sequence.

Many methods exist for the computation of discrete Fourier coefficients. The Goertzel algorithm requires a number of computations proportional to N^2 but can be used for all lengths N. When N is highly composite "fast" transformations can be used[9]. Thus, if N is of the form 2^q, then the number of computations can be made proportional to Nq. Although "fast" algorithms have been successfully used on General Purpose Computers they are too slow for real time computation since the algorithm iterates q times before achieving a solution. This problem can be overcome by the use of q processors in a pipeline architecture[10], although this increases the complexity of the processor.

A linear filter implementation also exists for the discrete Fourier transform which is both easily implemented and suitable for real time computation. This algorithm, called the chirp-Z-transform[11] is based on the substitution $nk = [n^2 + k^2 - (n - k)^2]/2$ and can be used for any length sequence N. The transform may be summarized as a premultiplication by a discrete chirp, convolution with a discrete chirp of twice the length, and postmultiplication by a discrete chirp. This transform may be computed with either acoustic surface wave filters or charge transfer devices[12].

DISCRETE COSINE TRANSFORM

Certain properties of Fourier transforms should be observed when using the discrete Fourier transform (DFT). The DFT is the Fourier series representation of periodically extended data and as such has convergence properties which depend on the input data. If the periodically extended data is discontinuous, then the convergence will be of the order of $1/n$; if, however, the periodically extended data is continuous, then the convergence will be of the order of $1/n^2$. The cosine transform makes the data appear to be continuous.

A discrete cosine transform of a data sequence g(n), n = 0, 1, . . . , (m - 1) can be defined as

$$G(0) = 2^{1/2}/N \sum_{n=0}^{N-1} g(n)$$

$$G(k) = 2/N \sum_{n=0}^{N-1} g(n) \cos [(2n + 1)\, k\pi/2N] \qquad k = 1, 2, \ldots, (N - 1)$$

The basis vectors are a class of discrete Chebyshev polynomials which are real and are given by

$$\phi_0 = 1/2^{1/2}$$

$$\phi_k = \cos [(2n + 1)\, k\pi/2N]$$

Ahmed[13] has investigated the use of these basis vectors as substitutes for the Karhunen-Loève basis vectors and finds that they are superior to the Fourier basis vectors and comparable to the Karhunen-Loève in reducing the mean-square-error in basis restricted transformations while maintaining the computation simplicity of a transformation which does not depend on the picture statistics.

A somewhat different definition of the cosine transform is given by Cooley et al.[14] along with the necessary operations required to compute the cosine transform with the Fast Fourier Transform. However, the auxiliary operations are somewhat involved and the length of the transform must be chosen so that the modified data is correct for a fast Fourier transform.

In order to take advantage of subsequent processing algorithms, it may be desirable to have the Fourier coefficients be a real autocorrelation sequence. This requires that the data sequence be extended so that it is symmetric about the first data value. This automatically assures that any subsequent periodic extension is continuous but also results in a data vector which has an odd number of values. A third cosine transform can be defined for the symmetrized data even though the resulting sequence length is odd. This transform can then be computed with the chirp-Z algorithm or directly by the modification of the Fourier transform:

$$G(k) = g(0) + \mathrm{Re} \sum_{n=1}^{N-1} g(n)\, e^{-jkn/(2N-1)}$$

Even though these symmetrized transforms conceptually use a symmetrized data sequence, their implementation is no more complicated than CZT of the original data length and only requires a change in the CZT reference function. The performance will be that of a sequence which has been extended to be continuous while the transform will be real and half of an autocorrelation function.

The third cosine transform is recommended as the primary transform for image processing in the horizontal direction and may easily be used on the video in real time. Its use in the vertical direction must be compared with other transforms, however, since memory must be employed for intermediate storage and this memory increases linearly with the size of the transform in the vertical direction.

SLANT TRANSFORM

In order to have a better match between the characteristics of television images and the basis vectors of the orthogonal transformations used to transform these images, Shibata and Enomoto[15] introduced a transform of which the second basis vector decreased in uniform steps over its length. Pratt, Welch, and Chen[16] have developed the slant transform in such a manner that it preserves the "sequency" properties of the Walsh-Hadamard transform as well as maintains a "fast" algorithm for computation.

Unfortunately, the transform is no longer binary and multipliers must be used in the computation. Thus, the transform may either be computed digitally or by means of a transversal filter with as many weighting networks as there are basis vectors in the transformation. These considerations appear to limit the applicability of the slant transform to applications such as the vertical encoding of television images; here the significant improvement in performance of the slant transform offsets lack, at present, of a simple real time computation implementation.

WALSH-HADAMARD TRANSFORM

The Walsh-Hadamard transform in one dimension has basis vectors which are the discrete Walsh functions. The discrete Walsh functions are an orthonormal sequence of length N canonically defined for $N = 2^q$, q a positive integer. The values of the Walsh functions are +1 and -1. The "sequency" property of the Walsh functions exists when the basis vectors are ordered by the number of sign

changes which occur in each basis vector; the number of sign changes increases linearly through the non-negative integers from 0 to $2^q - 1$. This corresponds to the number of sign changes which occur for the discrete Fourier Transform as it increases in frequency from DC to frequency 2^{q-1}.

The canonical Hadamard matrix of length 8 is of the form

$$H(3) = \begin{array}{cc} \begin{matrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \bar{1} & 1 & \bar{1} & 1 & \bar{1} & 1 & \bar{1} \\ 1 & 1 & \bar{1} & \bar{1} & 1 & 1 & \bar{1} & \bar{1} \\ 1 & \bar{1} & \bar{1} & 1 & 1 & \bar{1} & \bar{1} & 1 \\ 1 & 1 & 1 & 1 & \bar{1} & \bar{1} & \bar{1} & \bar{1} \\ 1 & \bar{1} & 1 & \bar{1} & \bar{1} & 1 & \bar{1} & 1 \\ 1 & 1 & \bar{1} & \bar{1} & \bar{1} & \bar{1} & 1 & 1 \\ 1 & \bar{1} & \bar{1} & 1 & \bar{1} & 1 & 1 & \bar{1} \end{matrix} & \begin{matrix} \text{Sequency} & 0 \\ & 7 \\ & 3 \\ & 4 \\ & 1 \\ & 6 \\ & 2 \\ & 5 \end{matrix} \end{array}$$

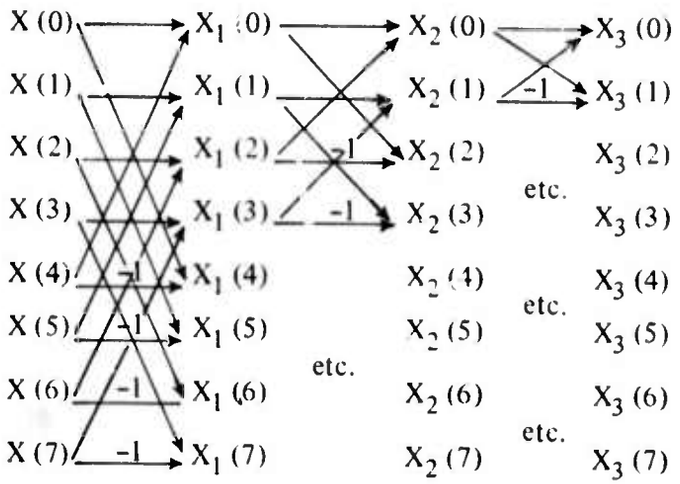
The first 8 Walsh functions ordered by sequency are of the form

$$\begin{array}{cc} \text{sequency } 0 & 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1 \\ 1 & 1\ 1\ 1\ 1\ \bar{1}\ \bar{1}\ \bar{1}\ \bar{1} \\ 2 & 1\ 1\ \bar{1}\ \bar{1}\ \bar{1}\ \bar{1}\ 1\ 1 \\ 3 & 1\ 1\ \bar{1}\ \bar{1}\ 1\ 1\ \bar{1}\ \bar{1} \\ 4 & 1\ \bar{1}\ \bar{1}\ 1\ 1\ \bar{1}\ \bar{1}\ 1 \\ 5 & 1\ \bar{1}\ \bar{1}\ 1\ \bar{1}\ 1\ 1\ \bar{1} \\ 6 & 1\ \bar{1}\ 1\ \bar{1}\ \bar{1}\ 1\ \bar{1}\ 1 \\ 7 & 1\ \bar{1}\ 1\ \bar{1}\ 1\ \bar{1}\ 1\ \bar{1} \end{array}$$

One factorization of H(3) into sparse matrices which gives a "fast" algorithm similar to the fast Fourier transform[17] is of the form

$$H(3) = \begin{bmatrix} 1 & 1 & & & & & & \\ 1 & \bar{1} & & & & & & \\ & & 1 & 1 & & & & \\ & & 1 & \bar{1} & & & & \\ & & & & 1 & 1 & & \\ & & & & 1 & \bar{1} & & \\ & & & & & & 1 & 1 \\ & & & & & & 1 & \bar{1} \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 & 0 & & & & \\ 0 & 1 & 0 & 1 & & & & \\ 1 & 0 & \bar{1} & 0 & & & & \\ 0 & 1 & 0 & \bar{1} & & & & \\ & & & & 1 & 0 & 1 & 0 \\ & & & & 0 & 1 & 0 & 1 \\ & & & & 1 & 0 & \bar{1} & 0 \\ & & & & 0 & 1 & 0 & \bar{1} \end{bmatrix} \begin{bmatrix} 1 & & & & 1 & & & \\ & 1 & & & & 1 & & \\ & & 1 & & & & 1 & \\ & & & 1 & & & & 1 \\ 1 & & & & \bar{1} & & & \\ & 1 & & & & \bar{1} & & \\ & & 1 & & & & \bar{1} & \\ & & & 1 & & & & \bar{1} \end{bmatrix}$$

The corresponding signal flowgraph is of the form



which is observed to have no multiplication other than ±1. This makes possible a simple hardware implementations of the algorithm and its computation for large block size.

The importance of large block size becomes significant when it is observed that the one-dimensional Walsh-Hadamard transform of size N^2 is the complete two-dimensional Walsh-Hadamard transform of size N if the input vector to the one-dimensional transform is obtained by appending successive rows of the two dimensional data together, starting with the first row. In terms of minimizing the auxiliary memory, this two-dimensional property may be significant. However it imposes constraints on the scanning of the original image and a "pipeline" implementation would be required for real time computation.

Alternatively, a transversal filter implementation of the Walsh-Hadamard transform is possible[18]. This implementation requires that the number of weighting networks be equal to the size of the transform which becomes large if a one dimensional transform is used for two-dimensional processing.

The basis vectors of length 8 in "sequency" order and shifted for real time implementation by transversal filtering are of the form

The large number of weighting networks can be eliminated through the use of an electronically variable tap weight implementation. If the basis vectors of the transformation are imagined to be in "sequency" order and successively shifted to the right with increasing "sequency," then a tap weight "sequency" can be derived that will successively compute each Walsh-Hadamard transform coefficient using a transversal filter with 2 N - 1 taps. The time behavior of a 15-tap filter is shown as tap weights for an electronically variable tap weight implementation of the form

	→ time
Tap 1	1 0 0 0 0 0 0 0
2	1 1 0 0 0 0 0 0
3	1 1 1 0 0 0 0 0
4	1 1 1 1 0 0 0 0
5	1 1 $\bar{1}$ 1 1 0 0 0
6	1 $\bar{1}$ $\bar{1}$ $\bar{1}$ $\bar{1}$ 1 0 0
7	1 $\bar{1}$ $\bar{1}$ $\bar{1}$ $\bar{1}$ $\bar{1}$ 1 0
8	1 $\bar{1}$ $\bar{1}$ 1 1 $\bar{1}$ $\bar{1}$ 1
9	0 $\bar{1}$ 1 1 1 1 1 $\bar{1}$
10	0 0 1 $\bar{1}$ $\bar{1}$ $\bar{1}$ $\bar{1}$ 1
11	0 0 0 $\bar{1}$ $\bar{1}$ 1 $\bar{1}$ $\bar{1}$
12	0 0 0 0 1 1 1 1
13	0 0 0 0 0 $\bar{1}$ $\bar{1}$ $\bar{1}$
14	0 0 0 0 0 0 1 1
15	0 0 0 0 0 0 0 $\bar{1}$

HAAR TRANSFORM

The Haar transform in one dimension has basis vectors which are the discrete Haar functions. The Haar functions are an orthonormal sequence $\{\phi_n^m\}$ characterized by two parameters m and n. For a sequence of length $N = 2^q$, $n = \{0, 1, \ldots, q\}$ and $m = \{1, 2, \ldots, 2^{n-1}\}$. A transform of length $N = 2^3$ has the basis vectors in the form

$$\phi_0 = \{1, 1, 1, 1, 1, 1, 1, 1\}$$

$$\phi_1^1 = \{1, 1, 1, 1, \bar{1}, \bar{1}, \bar{1}, \bar{1}\}$$

$$\phi_2^1 = \{2^{1/2}, 2^{1/2}, \bar{2}^{1/2}, \bar{2}^{1/2}, 0, 0, 0, 0\}$$

$$\phi^2 = \{0, 0, 0, 0, 2^{1/2}, 2^{1/2}, \bar{2}^{1/2}, \bar{2}^{1/2}\}$$

$$\phi_3^1 = \{2, \bar{2}, 0, 0, 0, 0, 0, 0\}$$

$$\phi_3^2 = \{0, 0, 2, \bar{2}, 0, 0, 0, 0\}$$

$$\phi_3^3 = \{0, 0, 0, 0, 2, \bar{2}, 0, 0\}$$

$$\phi_3^4 = \{0, 0, 0, 0, 0, 0, 2, \bar{2}\}$$

In general for $N = 2^q$

$$\phi_n^m = \begin{cases} 2^{(n-1)/2} & \dfrac{m-1}{2^{n-1}} < \tau \leq \dfrac{m+1/2}{2^{n-1}} \\ -2^{(n-1)/2} & \dfrac{m-1/2}{2^{n-1}} < \tau \leq \dfrac{m}{2^{n-1}} \\ 0 & 0 < \tau \leq \dfrac{m-1}{2^{n-1}} \text{ and } \dfrac{m}{2^{n-1}} < \tau \leq 1 \end{cases}$$

The Haar transform may be of interest in image encoding since it is a generalized differential encoding. In normal differential encoding, N-1 first differences of successive data points are transmitted as well as the first data point. However, if an error occurs in any of the differences or in the first data point, then all subsequent reconstructed values are in error. The Haar transform tends to

overcome this difficulty by transmitting N/2 contiguous differences and N/2 – 1 compound differences as well as the sum of data values.

From an implementation viewpoint, the Haar transform is interesting since it has several convenient hardware implementations. There are "fast," transversal, and recursive filter implementations.

The "fast" implementation of the Haar transform is particularly convenient because the original matrix has many zeroes. That is, instead of requiring a number of operations proportional to $N \log_2 N$ the number of operations required for the Haar transform is proportional to N.

Corresponding simplifications are possible in a transversal filter implementation since the number of weighting networks is only $1 + \log_2 N$ since many of the weighting networks may be used for the computation of more than one Haar coefficient.

The use of the Haar transform as the vertical transform for image encoding should be investigated since $\log_2 N + 1$ delay line memories may be used as recirculating integrators for the recursive calculation of the coefficients of the transform. This transform is second only to a differential pulse code modulation in memory requirement as a vertical transform while simultaneously providing some protection against error propagation. The Haar transform does not have a "sequency" property, although it may be considered to sample the input waveform at progressively coarser intervals[19].

4. HARDWARE IMPLEMENTATION FEASIBILITY

Laboratory tests at NUC have been made to determine the feasibility of implementing the various candidate compression systems. A computational system architecture for performing the two-dimensional image transformation which meets the ARPA requirements of low cost, low power, and low weight has been determined. This architecture is described in more detail in NUC TN 1026 which is attached as Appendix D.

Two areas of technology have been investigated in detail: surface wave devices and charge coupled devices. These devices, along with multipliers, provide a means of implementing a real time video transform processor. The block size which can be built for a two-dimensional transform is a function of the transform technology and the memory technology. The memory primarily impacts the choice of the vertical dimension transform.

A surface wave device presently under construction at NUC has been designed to implement the chirp-Z-algorithm for the Fourier transform. The overall time delay of the filter response is related to television scan rates and has been chosen as 1/4 of 53.5 μs. It was judged reasonable to divide this into 64 sample intervals, which led to the choice of 0.209 μs as the sample interval.

Implementation of the filter at bandpass was achieved by further subdivision of the sample interval into an integer number of periods that correspond to a frequency in the range 25-30 MHz. The chosen integer was 6, and the corresponding period (frequency) was .0348 μs (28.7 MHz). Each sample of the chirp function was represented by a group of three finger-pairs separated by a space of three carrier wavelengths.

The chirps sweep from zero to about 2.4 MHz to zero to 2.4 MHz and back to zero again (an inverted-W-FM). They are represented on the surface wave device as an amplitude modulated series of 28.7 MHz finger groups, each group being weighted by the amount of overlap of its fingers. One pair of sine and cosine chirp filters is interrogated by one input signal and the other pair by a second input signal. The four outputs are combined according to the complex CZT algorithm.

A charge transfer device (bucket brigade) transversal filter has been built by Texas Instruments as part of a study funded by Rome Air Development Center. It is a 200 stage device capable of operating at around 0.2 MHz. At present it is undergoing evaluation at NUC. It appears to be capable of implementing a modified cosine transform.

The extension of these technologies to the sizes and speeds necessary for video signal processing appears to be straightforward. The preliminary production cost, power, and weight of the various system components are given in Table 1.

In the course of the television bandwidth reduction study, the television sync signal was identified as a major area which had not been thoroughly considered. It is desirable to provide a high AJ margin on the sync signal so that the display will remain stable under adverse conditions. Preliminary feasibility studies have indicated that the RPV status could be included on the sync channel.

Table 1. Preliminary System Production Costs, Power, Weight

	Development Costs	*Production Costs*
Transversal filter		
512 Taps CCD	$50-75K	$ 5.00
SWD	in house ($50-75K)	100.00
9 months – 12 months development time		
Power < 1 watt	Weight < 1 lb	
Memory for vertical transform		
16 X 256 CCD	$50-75K	$ 15.00-20.00
SWD	$100K	100.00
9-12 mo. development time		
Power < 2 watts	Weight < 1 lb	
TV sync + data encoder		
SWD	$120K	120.00
Power ≈ 1.2 watts	Weight < 1 lb	
TV Transform data encoder		
LSI	10K	$ 10.00-20.00
Power < 5 watts	Weight < 1 lb	

5. RECOMMENDATIONS FOR FOLLOW-ON TASKS

A computational system architecture has been determined for performing the two-dimensional image transformation which meets the ARPA requirements of low cost, low power, and low weight. The recommended transform for at least the horizontal scan direction is the cosine transform. It has performance equivalent to the optimal transform. The choice of the transform in the vertical direction is impacted by the memory available in the RPV. Continued system design and memory development will be necessary for the airborne computational system.

The following tasks are recommended for follow-on work:

(a) Continue collaboration with the University of Southern California to develop and evaluate various transform techniques.

(b) Continue airborne system design.

(c) Determine and develop necessary memory technology.

(d) Design and develop transform data encoder compatible with modem.

(e) Provide television sync encoding with large AJ margin. It is also recommended that the down link data be included in the television sync system.

(f) Integrate system components at NUC.

(g) Simulate the channel and evaluate the AJ of the video system.

(h) Field test the system eighteen months after receipt of ARPA order.

(i) Monitor continuing work at Wright Patterson on necessary operator requirements for various missions with Decision Science.

(j) Design and develop ground station display and postprocessing.

REPORT SUMMARY

This report details the results of the first phase of a NUC program on image bandwidth reduction for application to the ARPA RPV problem of sending television images over spread spectrum channels. This report presents primarily a study and feasibility testing phase.

It has been found that significant bandwidth reductions are available by means of linear transformations applied to the image and filtering in the transform domain. Such techniques are the only presently known methods to obtain large redundancy reduction ratios. Bandwidth reductions by a factor of 6 are easily attainable without significant picture degradation. Reductions by a factor of 12 only slightly degrade the picture.

The image redundancy reduction has been done on a high speed general purpose computer at the University of Southern California Image Processing Institute. The Naval Undersea Center has determined that the linear transform techniques developed at USC and NUC can be implemented in real time with small low cost hardware suitable for the ARPA RPV.

For example, the Fourier transform may be implemented in this manner by utilizing the chirp-Z-transform algorithm which provides an exact decomposition of a discrete Fourier transform into premultiplication by a discrete chirp, correlation with a discrete chirp, and postmultiplication by a discrete chirp. The hardware required to implement this algorithm is thus the equivalent of a complex filter, a pair of complex multipliers, and a pair of complex sequence generators. The computation time required is linear in the desired discrete transform length, and the cost is nearly linear in the length. The sequence generators may be read-only memories, and the filter may be provided by a transversal filter device. High speed acoustic surface wave implementation will permit a two-dimensional transform to be realized as a succession of one-dimensional transforms. When high speed CCD devices become available they will provide direct replacement for the surface wave filters.

The principal result of this preliminary study phase of the program is the recommendation that a hardware development and implementation phase be initiated. A flight test of the bandwidth reduction system is feasible eighteen months after initiation of the hardware development phase. An overall bandwidth reduction of the order of 20 db appears to be feasible without significant picture degradation.

REFERENCES

[1] "A Collection of Unclassified Technical Papers on Target Acquisition," Proc. of the Target Acquisition Symp. at the Naval Training Center, Florida, Nov. 1972, conducted by Martin Marietta Aerospace.

[2] H. L. Snyder, "A Unitary Measure of Video System Image Quality," Proc. of the Target Acquisition Symp. at the Naval Training Center, Florida, Nov. 1972.

[3] A. Habibi and P. A. Wintz, "Image Coding by Linear Transformation and Block Quantization," *IEEE Trans Commun. Technol.*, vol. COM-19, Feb. 1971, pp. 50-62.

[4] M. Tasto and P. A. Wintz, "A Bound on the Rate-Distortion Function and Application to Images," *IEEE Trans. Inform. Theory*, vol. IT-18, Jan. 1972, pp. 150-159.

[5] D. J. Sakrison and V. R. Algazi, "Comparison of Line-by-Line and Two-Dimensional Encoding of Random Images," *IEEE Trans. Inform. Theory*, vol IT-17, July 1971, pp. 386-398.

[6] W. B. Davenport and W. L. Root, *Random Signals and Noise*, McGraw-Hill: New York 1958, pp. 96-101.

[7] W. D. Ray and R. M. Driver, "Further Decomposition of the Karhunen-Loève Series Representation of a Stationary Random Process," *IEEE Trans. Inform. Theory*, vol. IT-16, Nov. 1970, pp. 663-668.

[8] J. Pearl, "On Coding and Filtering Stationary Signals by Discrete Fourier Transforms," *IEEE Trans. Inform Theory*, vol. IT-19, March 1973, pp. 229-232.

[9] W. T. Cochran et al., "What is the Fast Fourier Transform," *Proc. IEEE*, vol. 55, Oct. 1967, pp. 1664-1674.

[10] H. L. Groginsky and G. A. Works, "A Pipeline Fast Fourier Transform," *IEEE Trans. Computers*, vol. C-19, Nov. 1970, pp. 1015-1019.

[11] L. R. Rabiner et al., "The Chirp *Z*-Transform Algorithm," *IEEE Trans. Audio and Electroacoustics*, vol. AU-17, June 1969, pp. 86-92.

[12] H. J. Whitehouse, J. M. Speiser, and R. W. Means, "High Speed Serial Access Linear Transform Implementations," presented at the 1973 Symp. All Applications Digital Computer (AADC), Orlando, Florida.

[13] N. Ahmed, T. Natarajan and K. R. Rao, "On Image Processing and a Discrete Cosine Transform," to be published.

[14] J. W. Cooley, P. A. W. Lewis, and P. D. Welch, "The Fast Fourier Transform Algorithm: Programming Considerations in the Calculation of Sine, Cosine and Laplace Transforms," *J. Sound Vib.*, vol. 12, July 1970, pp. 315-337.

[15] H. Enomoto and K. Shibata, "Orthogonal Transform Coding System for Television Signals," *J. of the Institute of TV Engineers of Japan*, vol. 24, Feb. 1970, pp. 99-108.

[16] W. K. Pratt, L. R. Welch, W. Chen, "Slant Transforms for Image Coding," Dept. of Elec. Eng., Univ. of So. Calif., Los Angeles.

[17] N. Ahmed (Depts. of Elec. Eng. & Comp. Sci., Kansas State Univ., Manhattan, Kansas) and K. R. Rao, Dept. of Elec. Eng., Univ. of Texas at Arlington, Arlington, Texas), "Walsh Functions and Hadamard Transform," pp. 8-13.

[18] J. Henaff, "Image Processing using Acoustic Surface Waves," *Electronics Letters*, vol. 9, March 1973, pp. 102-104.

[19] H. C. Andrews, "A Generalized Technique for Spectral Analysis," IEEE Trans. on Computers, vol. C-19, Jan. 1970, pp. 16-25.

APPENDIX A

USC PHOTOGRAPHS AND REPORTS

This appendix contains transformed photographs done under contract N00123-73-C-1507 at the University of Southern California. The second part of Appendix A is an article by Ali Habibi and Ronald S. Hershel. Ali Habibi is the principal investigator for the USC contract. A more formal USC report will be presented by the Image Processing Institute at a later date.

Original

12:1 Sample Reduction

8:1 Sample Reduction

Girl Scene

Fourier Transform Domain

Low Pass Filtering

(Vicar)

Original

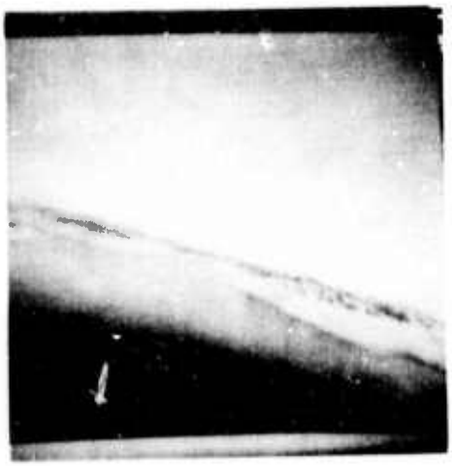
12:1 Sample Reduction

8:1 Sample Reduction

Bus Scene

Fourier Transform Domain

Low Pass Filtering
(Vicar)



Original

12:1 Sample Reduction

8:1 Sample Reduction

Three Boats Scene

Fourier Transform Domain

Low Pass Filtering
(Vicar)

Original

12:1 Sample Reduction

8:1 Sample Reduction

Baseball Diamond Scene
Fourier Transform Domain
Low Pass Filtering
(Vicar)

Fourier (N=256)

Real Fourier (N=512)

Cosine

Cosine (Horizontal)
and Haar (vertical)

Girl Scene
Circular Zone Filtering
Sample reduction: 12:1

Fourier (N=256)

Real Fourier (N=512)

Cosine

Cosine (horizontal)
and Haar (vertical)

Bus Scene
Circular Zone Filtering
Sample reduction: 12:1

Girl Scene

Bus Scene

Circular Zone

Girl Scene

Bus Scene

Rectangular Zone

Hybrid Cosine/ Haar Transform
(Horizontal Cosine and Vertical Haar)
Sample Reduction: 12:1

Original

Circular Zone
Sample reduction: 8:1

Original

Circular Zone
Sample reduction: 8:1

Bus Scene

Cosine Transform

Original

4:1 Zonal Selection

4:1 Zonal Selection
Spectrum Extrapolation

Hadamard Spectrum Extrapolation

Original

10:1 Zonal Selection

10:1 Zonal Selection
Spectrum Extrapolation

Hadamard Spectrum Extrapolation

Original

Left half; Sample reduction of 65/15
Right half: Extrapolated after sample reduction of 65/15

Left half; Sample reduction of 33/7
Right half; Extrapolation after sample reduction of 33/7

Original

Left half; Sample reduction of 65/15
Right half; Extrapolated after sample reduction of 65/15

Left half; Sample reduction of 33/7
Right half; Extrapolation after sample reduction of 33/7

(a) Original

(b) Karhunen-Loeve

(c) Fourier

(d) Hadamard

Figure 9. The original and the encoded pictures using one-dimensional transformations and DPCM systems, 1 bit/pixel, M = 16.

(a) Karhunen-Loeve

(b) Fourier

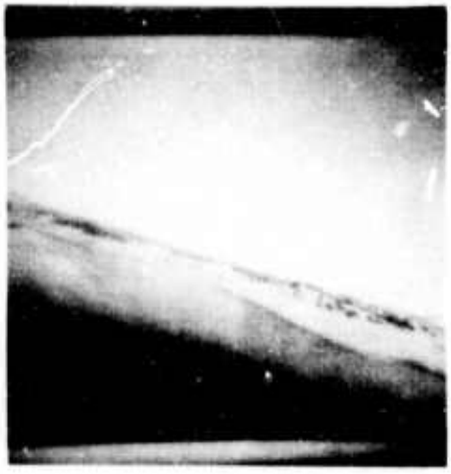
(c) Hadamard

Figure 10. The encoded pictures using the cascade of one-dimensional transformations and DPCM systems, 2 bits/pixel, M = 16.

Original

1 Dim. K. L. Transf. & DPCM
1 Bit/Pixel

1 Dim. Cosine Transf. & DPCM
1 Bit/Pixel



Original

1 Dim. K. L. Transf. & DPCM
1 Bit/Pixel

1 Dim. Cosine Transf. & DPCM
1 Bit/Pixel

Original

1 Dim. K. L. Transf. & DPCM
1 Bit/Pixel

1 Dim. Cosine Transf. & DPCM
1 Bit/Pixel

A UNIFIED REPRESENTATION OF DPCM AND TRANSFORM CODING SYSTEMS

by

Ali Habibi and Ronald S. Hershel

Abstract

We consider a transform coding system that uses a lower-triangular transformation to uncorrelate the data. Based on this transformation we propose a generalized DPCM system and show that at high bit rates it performs almost as well as coding by the method of principal components (Karhumen-Loeve transformation).

This study connects the transform coding system to the DPCM encoder by showing that the proposed system simplifies to a standard DPCM encoder for markov data.

I. Introduction

Two important classes of coding systems that are based on eliminating the correlation of the data prior to quantization and subsequent encoding are the differential pulse-code modulation (DPCM) and the transform coding systems. Both classes have received a great deal of attention in the recent literature and have been used successfully in coding pictorial data.[1-7] Historically both techniques were developed separately and are treated individually and often by entirely different groups of researchers. The only established link between the two systems is the fact that both attempt to generate a set of uncorrelated signals prior to quantization by a memoryless quantizer or a set of quantizers. The Karhunen-Loeve transformation, also known as the method of principal components for sampled data, generates the uncorrelated signal by using an orthogonal transformation that diagonalizes the autocovariance matrix of the data. The DPCM system, on the other hand, is designed based upon modeling the data by a markov process, then using a best linear prediction to obtain the set of uncorrelated signals.

This research was supported by the Naval Undersea Center, San Diego under contract N00123-73-C-1507 and by the Advanced Research Projects Agency of the Department of Defense and was monitored by the Air Force Eastern Test Range under contract F08606-72-C-0008. The authors are with the Department of Electrical Engineering, University of Southern California, Los Angeles, Ca. 90007

In this paper we will consider the DPCM system and show that it is indeed a member of the class of transform coding systems where a lower triangular matrix is used to transform the data to a set of uncorrelated signals. This approach will provide a fresh outlook to the theory of the DPCM system realizing that the DPCM system is a special case of a more general coding system where a lower triangular matrix is used to transform the data to a set of uncorrelated signals. This system, unlike coding by unitary transformations, combines the operation of the transformation with that of the quantizers. The entries of the lower triangular operation are related to the covariance matrix of the data. For an n^{th} order markov process the triangular operator has n off-diagonal bands with identical elements within each band. Thus, for this special case the transformation is performed recursively and is identical to the linear predictor in an n^{th} order DPCM system.

II. Transformation by Lower-Triangular Operators

Consider an N-dimensional data vector $X = (x_1, x_2, \ldots, x_b)^T$ and let X represent a sample vector from an ensemble of N-dimensional zero mean random variables. X can also be considered as a vector of N samples $\{x_i\}$ that results from sampling a continuous random process uniformly over a finite interval. A vector $Y = (y_1, y_2, \ldots, y_n)^T$ can always be generated from a linear combination of x_i's as

$$y_1 = x_1$$

$$y_j = x_j - \sum_{k=1}^{j-1} \ell_{kj} x_k \qquad \text{for } j = 2, \ldots N \tag{1}$$

or in vector form

$$Y = LX \tag{2}$$

where L is a unit lower-triangular matrix; i.e.,

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ -\ell_{21} & 1 & 0 & 0 & \cdots & 0 & 0 \\ -\ell_{31} & -\ell_{32} & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & & & & \vdots & \vdots \\ -\ell_{N1} & -\ell_{N-1,2} & \cdots & & & \ell_{N,N-1} & 1 \end{bmatrix} \qquad (3)$$

Denoting the covariance matrices of X and Y by C_X and C_Y respectively, (2) implies that

$$C_Y = L\, C_X\, L^T \qquad (4)$$

Cholesky has shown that for every symmetric positive definite matrix C_X there exists a real non-singular lower-triangular matrix L such that matrix $L\, C_X\, L^T$ is diagonal.[8] Martin and Wilkinson have considered numerical algorithms for finding L and C_Y and have developed efficient techniques requiring only $N^3/6$ multiplications.[9,10]

The fact that the transformation (2) diagonalizes the covariance matrix of X indicates that the elements of Y are uncorrelated.

Some of the significant properties of the transformation by the unit lower-triangular operator are:

1) The unit lower-triangular operator L is not unitary; thus transformation (2) does not preserve the length of a vector. As a result though the determinant of the covariance matrix of X is invariant under this transformation, the trace of the covariance matrix is not invariant.

2) This transformation does not share the optimum concentration of energy in the first $M \leq N$ components of Y exhibited by the method of principal components. Indeed, for an n^{th} order markov process the variances of the Y components, except the first n components, are all equal.

3) Transformation with the lower triangular operator L does not require a transformation delay.

Note that if the components of X are samples from an n^{th} order markov process the stochastic linear model of (1) will be

$$y_j = x_j - \sum_{k=1}^{n} \alpha_k x_{j-k} \qquad j = 1,2,\ldots,N \tag{5}$$

where $x_i = 0$ for $i = 0,-1,-2,\ldots$. Then the operator L_n will be a banded matrix of n+1 bands; i.e.,

$$L_n = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & & & 0 \\ -\alpha_1 & 1 & 0 & 0 & \cdots & & & 0 \\ -\alpha_2 & -\alpha_1 & 1 & 0 & \cdots & & & 0 \\ \cdot & \cdot & & & & & & \cdot \\ \cdot & \cdot & & & & & & \cdot \\ \cdot & \cdot & & & & & & \cdot \\ -\alpha_n & -\alpha_{n-1} & \cdots & -\alpha_1 & 1 & 0 & \cdots & 0 \\ 0 & -\alpha_n & \cdots & -\alpha_2 & -\alpha_1 & 1 & 0 \cdots & 0 \\ \cdot & \cdot & & \cdot & \cdot & & & \cdot \\ \cdot & \cdot & & \cdot & \cdot & & & \cdot \\ \cdot & \cdot & & \cdot & \cdot & & & \cdot \\ 0 & 0 & \cdots & 0 & -\alpha_n & \cdots & -\alpha_1 & 1 \end{bmatrix} \tag{6}$$

Transformation of an N vector with operator L_n requires less than nN multiplication as compared to $N^2/2 - N$ multiplications needed for transformation with the unit lower-triangular operator L in its general form. Furthermore, since transforming with L_n operator requires only the n most recent components of X, this transformation can be performed by using a feed-back loop identical to one in an n^{th} order DPCM system to perform the transformation recursively. In this case the complexity of the transformation is independent of the dimensionality of X and depends only on the order of markov process n. The block diagram of the system using the feed-back loop for operation $L_n X$ is shown in Figure 1.

III. Coding by the Method of Principal Components

Before proceeding with the lower triangular transformation we will briefly review coding by the method of principal components and block quantization. The block diagram of this system is shown in Figure 2. Huang

and Schultheiss[11] considered this problem first and proved that given a total number of binary digits $N\theta$ (a finite capacity digital channel) the average coding error is minimum if vector X is transformed to an uncorrelated vector Y by an orthogonal operator A and Y is quantized by block quantization. Matrix A is an operator that diagonalizes the covariance matrix of X and the block quantization of Y involves assigning $N\theta$ binary digits to N components of Y proportional to the logarithms of their variances d_i; i.e.,

$$m_i = \theta + \frac{2}{\ln 10} \ln \frac{d_i}{|C_X|^{1/N}} \tag{7}$$

where $|C_X|$ is the determinant of covariance matrix C_X and $\sum_{i=1}^{N} m_i = N\theta$.

This rule was obtained by observing that optimal quantization of Gaussian variables y_i using a quantizer with 2^{m_i} levels introduces a quantization error q_i such that*

$$E\{q_i^2\} \cong E\{y_i^2\} \ (10)^{-m_i/2} \tag{8}$$

Substituting (7) in (8) gives

$$E\{q_i^2\} \cong |C_X|^{1/N} \ (10)^{-\theta/2} = \Delta \tag{9}$$

Equation (9) indicates that the block quantization of Y results in an equal quantization error Δ for all components of Y. Note that (9) does account for the inaccuracies due to using integers for m_i, thus Δ should be viewed as the lower bound of the quantization error. Since each quantizer is independent and the quantization error is additive, the quantized vector Y^* is

---

*Huang and Schultheiss use an approximation that involves $2^{-2m_i \ln 2}$, but further studies of the problem have shown that (8) is a better approximation.[12,13]

$$Y^* = AX + Q \tag{10}$$

and

$$E\{QQ^T\} = \Delta I \tag{11}$$

where I is the identity matrix. This gives a reconstructed vector X^* such that

$$X^* = X + A^{-1}Q \tag{12}$$

The total error for the system is the trace of the covariance matrix of the error vector $X-X^*$, thus from (12) the average coding error is

$$\varepsilon_b^2 = \frac{1}{N} \text{ tr } \{\Delta I\} = \Delta \tag{13}$$

It is interesting to note that the above coding system not only minimizes the coding error but it also gives an uncorrelated error in the signal domain which is a desirable property in many applications of the coding system.

Note that if all components of Y are quantized using θ bits for each component the total coding error would be the same as it would result by coding the components of X directly. This can be seen by noting that using θ bits per component the average coding error is

$$\varepsilon_s^2 = \frac{1}{N} \text{ tr } [A^{-1} E\{\hat{Q}\hat{Q}^T\}(A^{-1})^T] \tag{14}$$

where the components of the quantization error $\hat{Q}$ are uncorrelated with variances equal to $E\{y_i^2\}(10)^{-\theta/2}$. Therefore

$$\varepsilon_s^2 = \frac{1}{N} 10^{-\theta/2} \text{ tr}[A^{-1} C_Y (A^{-1})^T] = 10^{-\theta/2} \sigma_X^2 \tag{15}$$

where σ_X^2 is the common variance of $\{x_i\}$ and (15) is the average coding error that results by quantizing the components of X vector directly. We note in passing that this result is true for any transformation that generates an uncorrelated Y vector including the transformation by the lower-triangular operator L.

IV. A Generalized DPCM System

Transformation of vector X by the lower-triangular operator L results in a vector Y where the components of Y are uncorrelated and in general have unequal variances. Components of Y are quantized uring the block quantization technique discussed in Section III and are transmitted. Vector X can be reconstructed at the receiver, within some level of degradation, by operating on the coded vector Y+Q with operator L^{-1}. The average coding error ε_L^2 is

$$\varepsilon_L^2 = \frac{\Delta}{N} \text{ tr } [L^{-1}(L^{-1})^T] \tag{16}$$

where Δ is defined by (13). The readers will note that (16) is subject to similar inaccuracies as (18) and thus it also should be viewed as a lower bound or the performance of this coding system.

Comparison of (13) and (16) show that using lower-triangular operator, a non-unitary matrix, rather than the method of principal components to uncorrelate the data prior to its block quantization results in an inferior coding system. This is evident from the fact that L^{-1} is a unit lower-triangular matrix, thus

$$\text{tr } [L^{-1}(L^{-1})^T] \geq \text{tr } I \tag{17}$$

When components of X belong to a first order markov process the block diagram of the above coding system is shown in Figure 3a. Then only two quantizers are used where one encodes y_1 and the other y_2 through y_N; since y_2 through y_N have identical variances. From published literature the performance of this encoder improves by including the quantizer in the predictor loop as shown in Figure 3b. This combines the operation of the quantizer with the transformation and is identical to a DPCM system with the stipulation that a separate quantizer is used to encode the first component of the differential signal. Naturally the effect of using only one quantizer is negligible for large N; thus two systems are identical.

Since L is a triangular matrix, the operation of the quantizers can be combined with the transformation to give a generalized DPCM system. We propose a combination of the transformation and the block quantization, similar to one in the DPCM system, and show that the performance of the generalized DPCM system at high bit rates approaches the performance of the

encoder using the method of principal components and the block quantization. This system uses $N^2/2 - N$ multiplications to transform the data and requires no coding delays. It simplifies to an n^{th} order DPCM encoder if components of data X belong to an n^{th} order markov process.

The block diagram of the generalized DPCM system is shown in Figure 4. The components of input vector $x_1, x_2, \ldots, x_N$ are operated upon in sequence at times $t_1, t_2, \ldots, t_N$ respectively. Switches S_i, $i = 1,2,\ldots N$ are closed at t_i and remain open at all other times. Variables z_i, $i = 1,\ldots,N$ are generated recursively and are stored in separate locations. z_i through z_j are read out at <u>each</u> t_j and are multiplied by variables C_1 through C_j, respectively, to form z_{j+1} which in turn is stored at time t_{j+1}. Vector C is the j^{th} row of the matrix (L-I) at time t_j for $j = 1,2,\ldots,N$.

Denoting the components of the quantized signals by $\{y_i^*\}$, the recursive relation for $\{z_i\}$ from the block diagram is

$$z_i = y_i^* + \sum_{j=1}^{i-1} \ell_{ij} z_j \qquad \text{for } i = 2,3,\ldots,N \tag{18}$$

and $z_i = y_1^*$, furthermore

$$Y^* = \hat{Y} + Q \tag{19}$$

and

$$\hat{Y} = LX + (L-I)Q \tag{20}$$

where $\hat{Y}$ is the signal at the input of the quantizer, Q is the vector of quantization error. These equations indicate that vector $Z = (z_1, z_2, \ldots, z_N)^T$ is identical to the reconstructed vector X^* at the receiver. Thus from (19)

$$X - X^* = Q \tag{21}$$

and

$$\hat{Y} = Y + (L-I)Q \tag{22}$$

where Y refers to the uncorrelated vector LX. From (22) and realizing the fact that $E\{y_i q_j\} = 0$ for $j = 1,2,\ldots,i-1$

$$E\{\hat{y}_i^2\} = E\{y_i^2\} + \sum_{j=1}^{i-1} \sum_{k=1}^{i-1} \ell_{ij}\ell_{ik}\, E\{q_i q_j\} \qquad i = 2,\ldots,N \qquad (23)$$

and $E\{\hat{y}_1^2\} = E\{y_1^2\}$. Note that $\{q_i\}$ are in general correlated since they result from quantizing correlated variates $\{\hat{y}_i\}$. This hampers an accurate analysis of this system however assuming fine quantization of $\{\hat{y}_i\}$ the correlation of $\{q_i\}$ are negligible and if block quantization of $\{y_i\}$ is performed optimally, using the variances of $\{\hat{y}_i\}$, from (8) and (23)

$$E\{q_i^2\} = \frac{(10)^{-\theta/2}\,|C_X|^{1/N}}{1 - (10)^{-\theta/2}\left(\dfrac{|C_X|^{1/N}}{d_i}\right)\sum_{j=1}^{i-1}\ell_{ij}^2} \qquad i = 2,\ldots,N \qquad (24)$$

and $E\{q_1^2\} = \Delta$. Equations (23) and (24) specify variances of $\{\hat{y}_i\}$ in terms of the covariance matrix of the data C_X and the bit rate θ. From (21) and (24) the average coding error for the generalized DPCM system is

$$\varepsilon_G^2 = \frac{\Delta}{N}\left[\sum_{i=2}^{N} \frac{1}{\left(1 + \dfrac{\Delta}{d_i}\sum_{j=1}^{i-1}\ell_{ij}^2\right)}\right] + \frac{\Delta}{N} \qquad (25)$$

where Δ, defined by (9), is the coding error by the method of principal component. Analogous to (9) the error given by (25) also should be treated as a lower bound for the performance of the generalized DPCM system. Figures 5 and 6 show the performance of this system as given by (25) and the performance of the optimum system (method of principal components) for two non-markov processes. These processes are defined by their autocovariance functions

$$R_1(\tau) = \frac{1}{1 + 0.2\,\tau^2} \qquad \tau = 0,1,2,\ldots \qquad (26)$$

$$R_2(\theta) = \frac{1}{1 + 0.05\,\tau} \qquad \tau = 0,1,2,\ldots \qquad (27)$$

These figures also show the coding performances of PCM and the encoder that uses transformation by operator L and block quantization separately.

For the special case where $\{X_i\}$ is an n^{th} order markov process the operator L is replaced by L_n of(16).Then the elements of vector C remain fixed for all t_j, except for $j = 1, 2, 3, \ldots, n$. This also makes the variances of the components of the transformed vector, again except the first n components, identical. Thus the block quantizer will consist of only n+1 quantizers. This reduces the generalized encoder to an n^{th} order DPCM system with the stipulations that; first, n+1 quantizers are used to encode the differential signal, second the quantizers are designed using the variance of the differential signal where the effect of the quantization error is accounted for. Naturally the degradation of the encoder due to using only one quantizer becomes negligible for large values of N.

For the case of the first order markov process the coding error from (25) for large values of N is

$$\varepsilon_G^2 = \frac{(10)^{-\theta/2}\ (1-\alpha_1^2)}{1 - (10)^{-\theta/2}\ \alpha_1^2} \tag{28}$$

The numerator of (28) is the coding error of the first order markov process by the method of the principal components for large values of N. The excess error due to using an improved DPCM rather than the method of the principal components is less than 1.1% for $\alpha_1 = 0.9$ and $\theta = 2$, and reduces sharply at higher bit rates.

V. Conclusions

We have considered a transform coding system that combines a lower-triangular transformation with a block quantizer to convert a set of sampled data to a string of binary digits which can be converted back into a replica of the data. Separating the operation of the quantizer from that of the transformation, as in the unitary transform encoders, results in a correlated coding error at the receiver and a rather inefficient coding system. However, combining the operation of the quantizers with the transformation of the data such that the coding error at the receiver is uncorrelated improves

the coding efficiency of the system significantly. The resultant encoder is only slightly suboptimum to the system that uses the method of principal components at low bit rates. At high bit rates the performance of both systems are identical. This was verified for a number of processes with different autocovariance functions.

Transformation by the lower-triangular operator involves no coding delay and requires less than half as many multiplications as the method of principal components requires.

Combination of the transformation and the quantization is similar to the procedure employed in the DPCM system. Furthermore the proposed encoder reduces to an improved form of DPCM encoder for markov processes, thus it is referred to as a generalized DPCM encoder.

The fact that the generalized DPCM system reduces to a DPCM encoder for markov processes establishes a link between the DPCM and the transform coding systems making a unifoed approach to coding by these two techniques possible. Furthermore we have shown that the theoretical performance of a modified DPCM system is optimum at high bit rates. At low bit rates it is only slightly suboptimum to the encoder using the method of principal components.

References

1. C. C. Cutler, "Differential quantization of communication signals", U.S. Patent 2 605 361, July 29, 1952, and U.S. Patent 2 724 740, Nov. 22, 1955.

2. J. B. O'Neal, Jr., "Delta modulation quantizing noise - Analytical and computer simulation results for Gaussian and television input signals", Bell Syst. Tech. J., Vol. 45, pp. 117-142, Jan. 1966.

3. J. O. Limb, C. B. Rubinstein and K. A. Walsh, "Digital coding of color picturephone signals by element-differential quantization", IEEE Trans. Commun. Tech., Vol. COM-19, No. 6, pp. 992-1005, Dec. 1971.

4. W. K. Pratt, J. Kane and H. C. Andrews, "Hadamard transform image coding", Proc. IEEE, Vol. 57, pp. 58-68, Jan. 1969.

5. P. A. Wintz, "Transform picture coding", Proc. of IEEE, Vol. 60, No. 70, July 1972, pp. 809-821.

6. A. Habibi and P. A. Wintz, "Image coding by linear transformations and block quantization", IEEE Trans. Commun. Tech., Vol. COM-19, pp. 50-63, Feb. 1971.

7. A. Habibi, "Comparison of n^{th}-order DPCM encoder with linear transformations and block quantization techniques", IEEE Trans. Commun. Tech., Vol. COM-19, No. 6, pp. 948-956, Dec. 1971.

8. L. Fox, "Practical solution of linear equations and inversion of matrices", Appl. Math. Ser. Nat. Bur. Stand. 39, pp. 1-54, 1954.

9. R. S. Martin and J. H. Wilkinson, "Symmetric decomposition of a positive definite matrix", Numerische Mathematik 7, pp. 362-383, 1965.

10. R. S. Martin and J. H. Wilkinson, "Symmetric decomposition of positive definite band matrices", Numerische Mathematik 7, pp. 355-361, 1965.

11. J. J. Y. Huang and P. M. Schultheiss, "Block quantization of correlated Gaussian random variables", IEEE Trans. Commun. Syst., Vol. CS-11, pp. 289-296, Sept. 1963.

12. P. A. Wintz and A. J. Kurtenbach, "Waveform error control in PCM telemetry", IEEE Trans. Inform. Theory, Vol. IT-14, pp. 650-661, Sept. 1968.

13. A. Habibi, "Performance of zero-memory quantizers using rate distortion criteria", to be published.

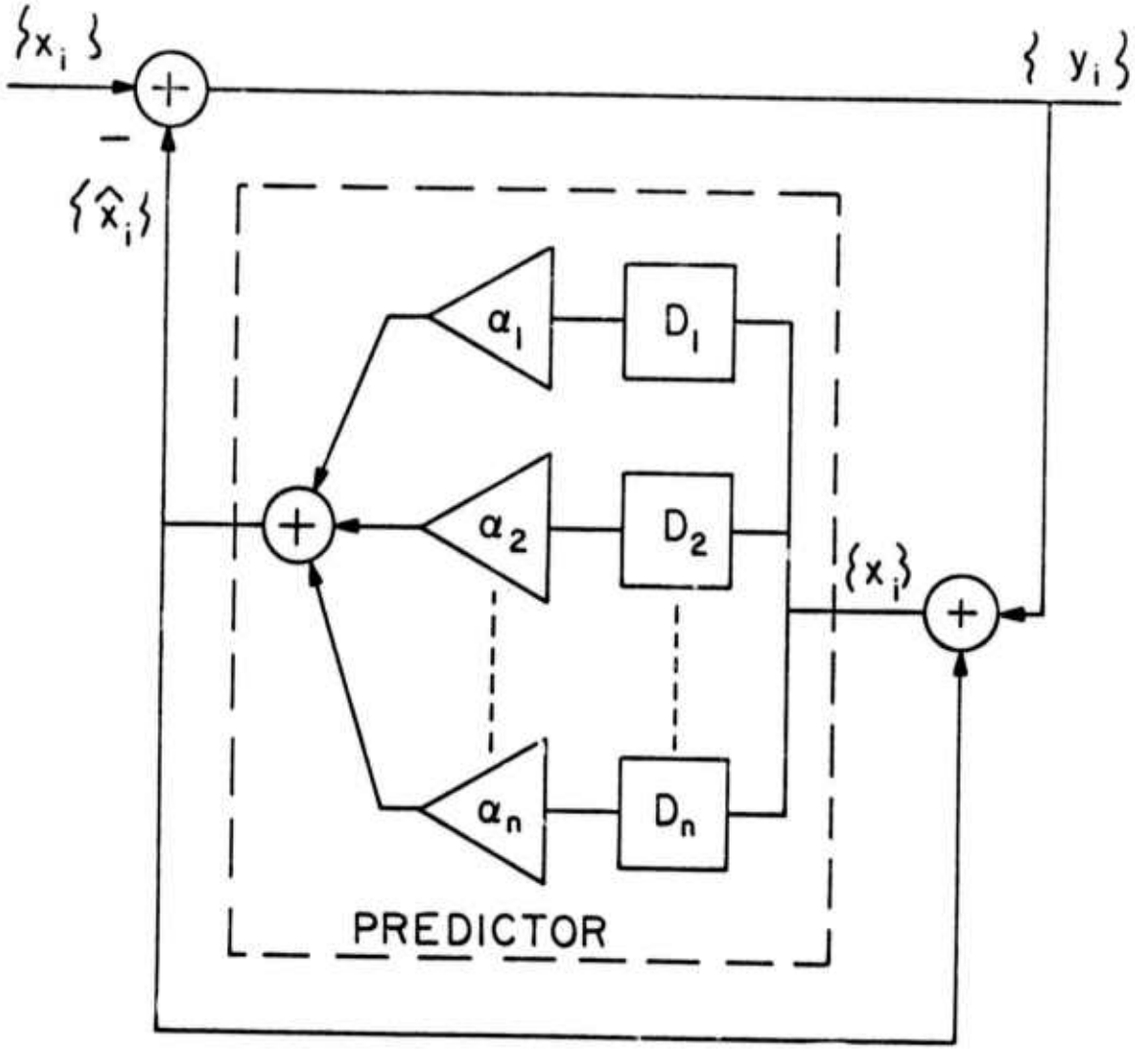


Fig 1 Block Diagram of the System Performing Transformation $Y = L_n X$

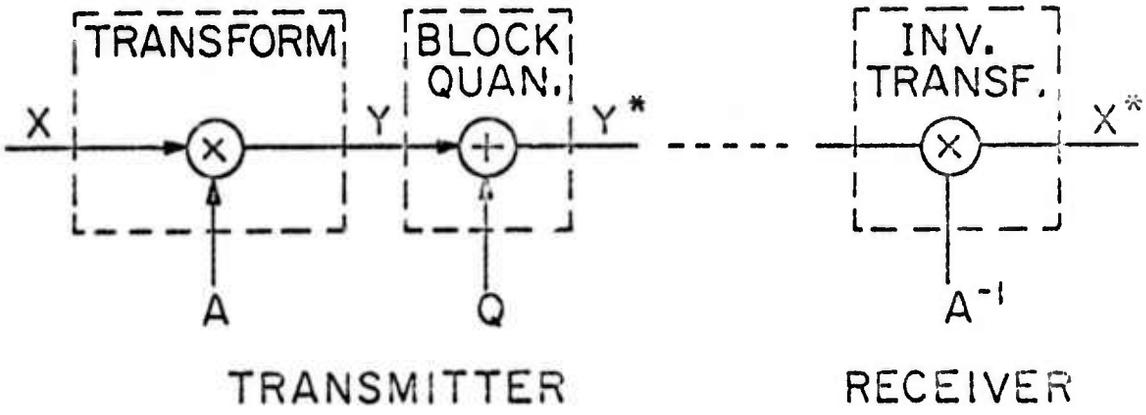


Fig 2. Block Diagram of The Transform Coding System.

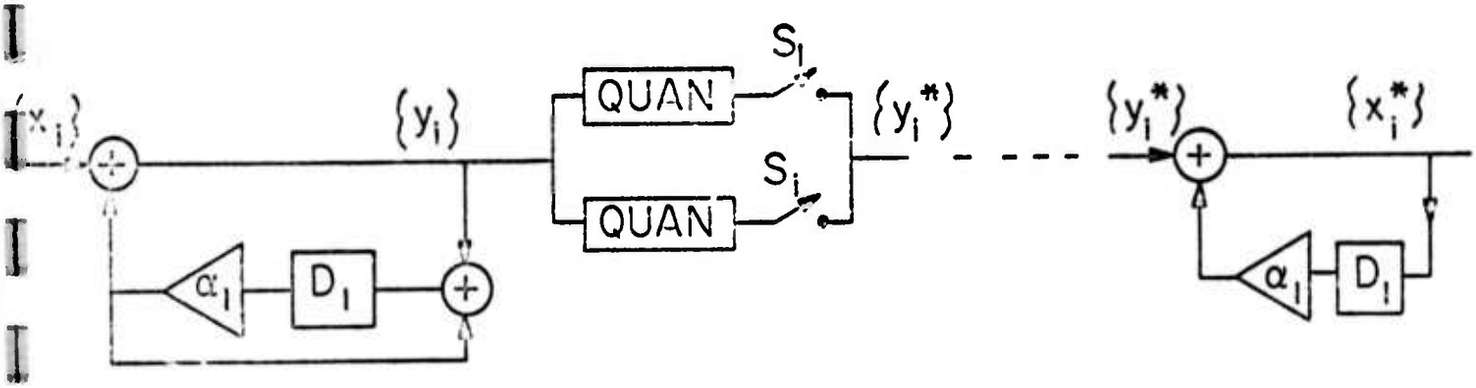


Fig 3a. Block Diagram of the Encoder Using L_1 Transformation and Block Quantization.

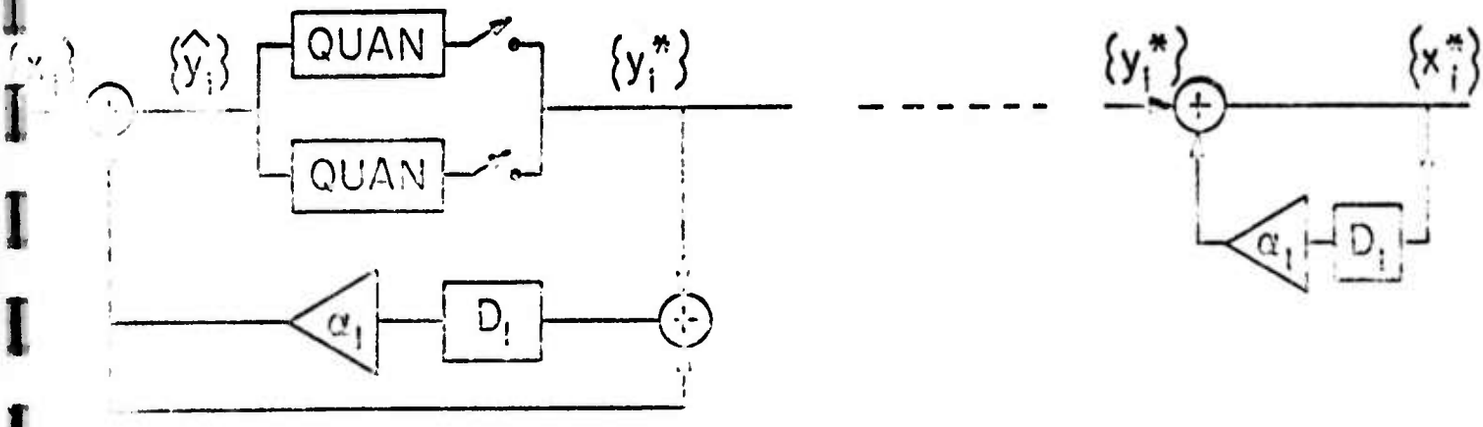


Fig 3b. Block Diagram of Modified DPCM Encoder.

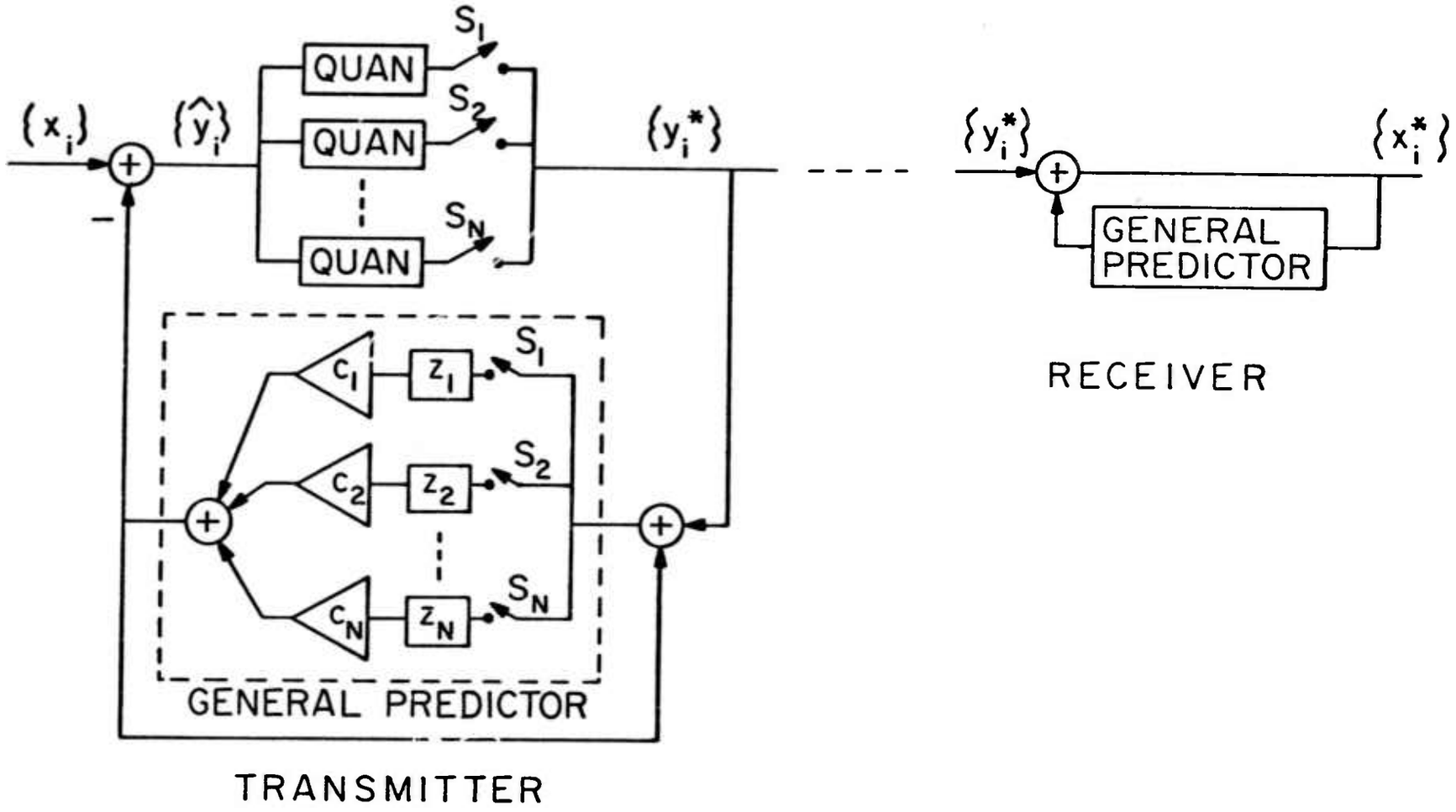


Fig 4 Block Diagram of the Ceneralized DPCM

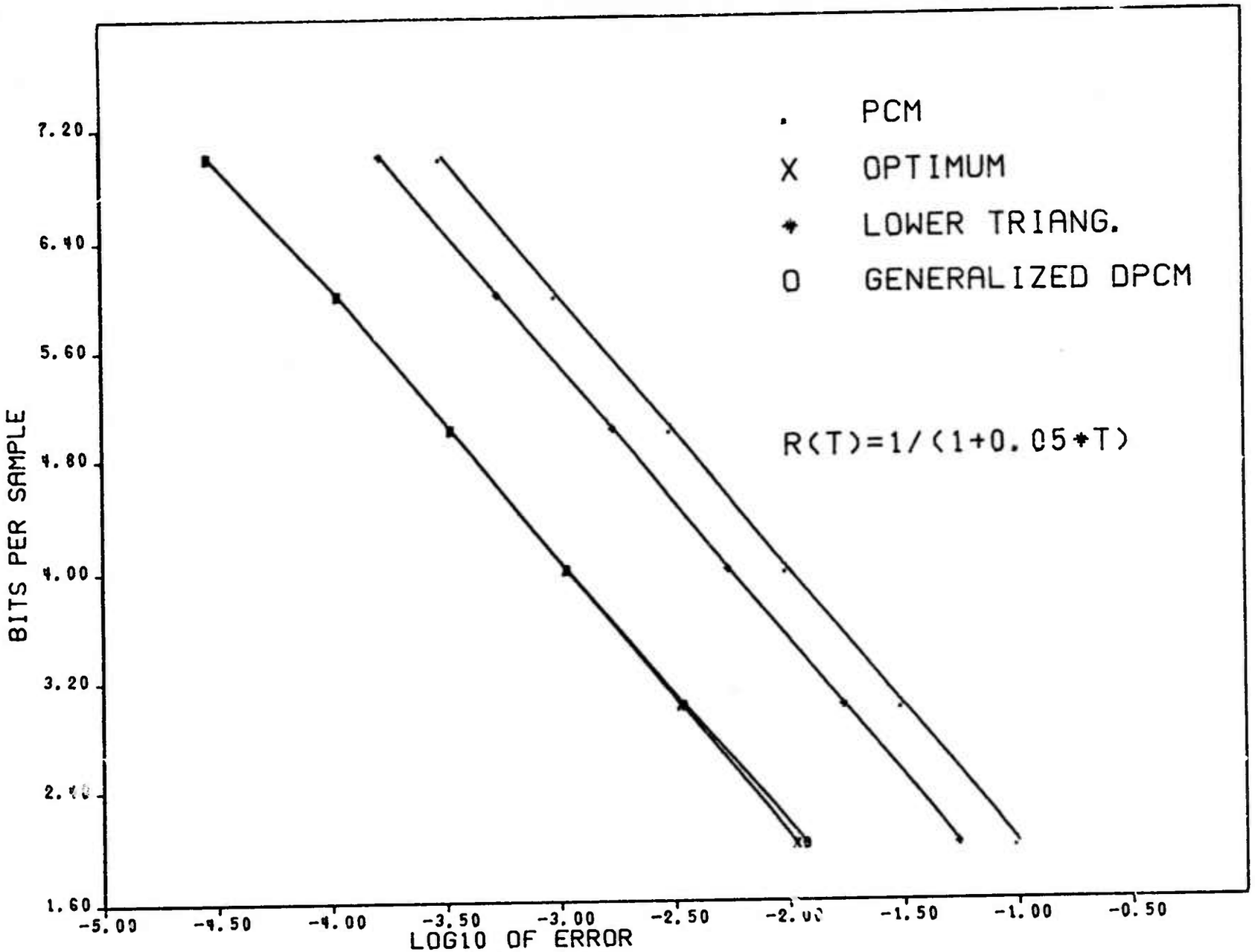


Figure 5. Bit rate versus the coding error of various encoders. Optimum refers to coding by the method of principal components (K. L. Trans.) and block quantization.

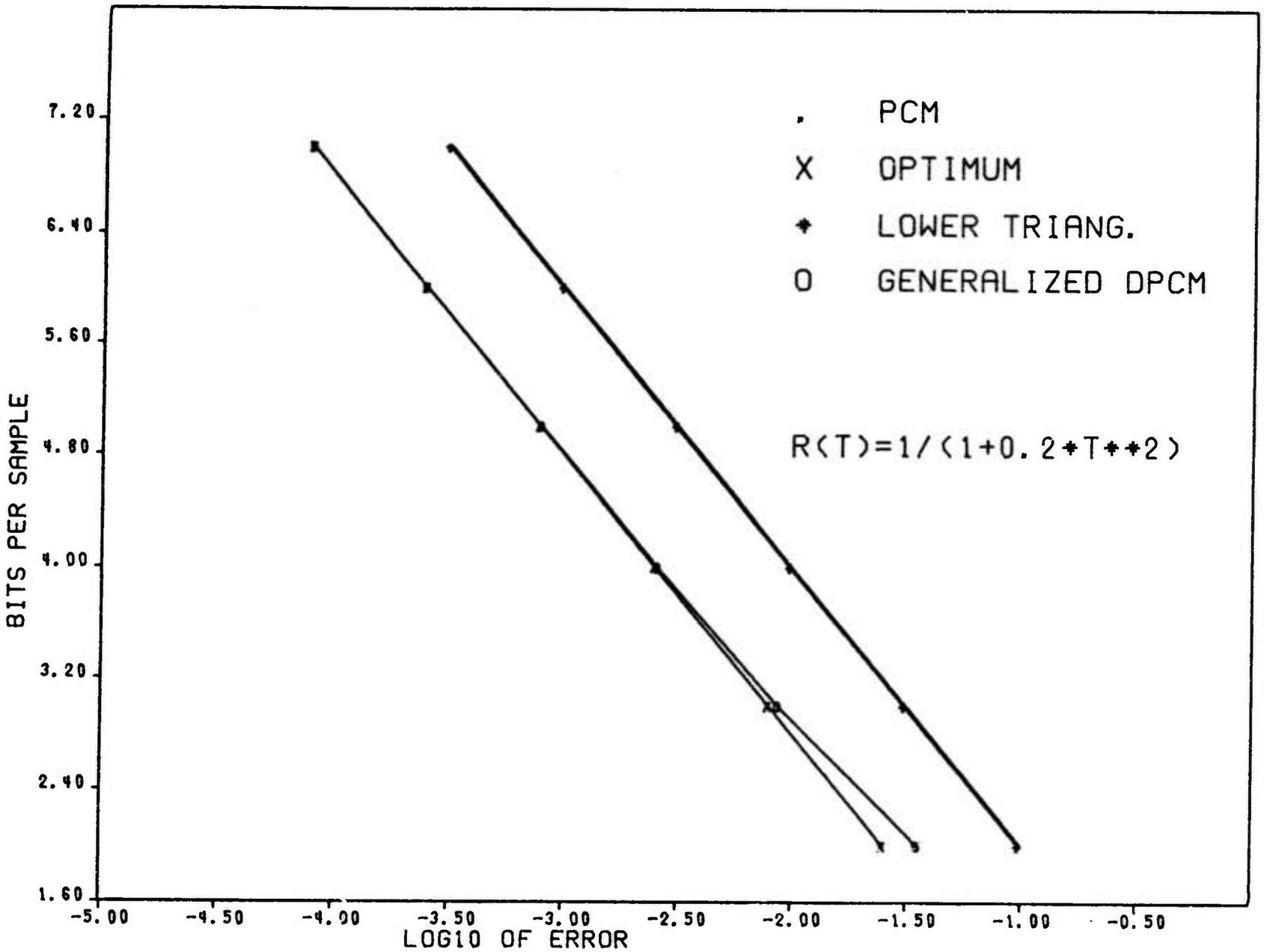


Figure 5. Bit rate versus the coding error of various encoders. Optimum refers to coding by the method of principal components (K. L. Trans.) and block quantization.

APPENDIX B

NAVAL UNDERSEA CENTER

San Diego, California 92132

SIGNAL PROCESSING INTERPRETER

PRELIMINARY DESCRIPTION AND USER'S GUIDE

by

J. M. SPEISER

Processor Development Branch

25 June 1973

CONTENTS

ABSTRACT

This report describes a signal processing interpreter which was developed as part of the ARPA sponsored program for image transmission via spread spectrum links. The signal processing interpreter SPIN3 is an interactive program for use at a time-sharing demand terminal. It provides the user with the equivalent of a calculator designed to perform signal processing operations and provides the software equivalent of a large number of modules for breadboarding a complete signal processing system.

This interpreter is meant to be used by engineers and scientists who are familiar with signal processing, but who may have no knowledge of programming.

The signal processing interpreter is particularly useful for the rapid investigation of systems whose complexity precludes a complete analytic study, and whose utilization of new components may make hardware breadboarding undesirable because of cost and procurement time limitations.

INTRODUCTION

The signal processing interpreter is an interactive program for use at a time-sharing demand terminal. It provides the user with the equivalent of a calculator designed to perform signal processing operations - the software equivalent of a large collection of modules for breadboarding a complete signal processing system. Such a system might include an information source, a source encoder, a channel encoder or modulator, a channel with noise, jamming, and multipath, a channel decoder or demodulator, and an information decoder. Typically, one would wish to compare several such systems operating through each of several channels.

The complexity and variety of the systems and channels will frequently preclude a complete analytic comparison, while cost and procurement time limit the flexibility of a hardware breadboard of a system utilizing new components.

The two critical requirements for such a signal processing interpreter are: (a) It must include the most frequently needed signal processing operations, with a provision for readily adding new operations as the need arises; and (b) It must be easy to use by engineers and scientists who are familiar with signal processing, but who may have no knowledge of programming.

The signal processing interpreter SPIN3 which will be described here was developed as part of the ARPA sponsored program for image transmission via spread spectrum links. The command set is meant to be general enough for a variety of signal processing problems. In addition, special commands are available for examining and processing portions of pictures in order to facilitate the comparison of proposed image transmission systems.

HOW TO USE THE SIGNAL PROCESSING INTERPRETER

The signal processing interpreter is structured around an operational stack. Such stacks are used in many compilers, in the Burroughs B5000 computer, and in the HP-35 calculator [1,2,3]. The stack structure used in the signal processing interpreter is in fact identical to that used in the HP-35 except for the differences necessitated by dealing with complex vectors rather than real scalars.

The stack consists principally of the complex vectors X,Y,Z, and T. There is also a storage vector S which is effected only by the command STORE, and which effects the stack only through the command RECALL. Associated with each of these vector registers is an integer NX, NY, NZ, NT, NS which indicates the current dimension of the associated vector.

After the user performs the start up procedure described in Appendix B, and also after the successful completion of any command, the interpreter will solicit a command by displaying the message "PLEASE ENTER NEXT COMMAND". The user may then enter any command (up to 25 characters enclosed in single quotes) and then depress the carriage control key to send the command to the computer. One of the five following outcomes will then result:

a) The command will be recognized and performed, and a new command will be solicited.

b) The command will be recognized, and the program will request the user to enter additional data from the keyboard.

c) The command will be recognized, but not yet implemented. In this case, the user will be so informed, and a new command will be solicited. In many cases, the desired command may be replaced by a short sequence of commands which have already been implemented.

d) The command will not be recognized; the user will be so informed; and a new command will be solicited.

e) The attempted command is not in the form of an expression of zero to twenty-five characters enclosed in single quotes. This will produce an error

exit from the signal processing interpreter and an Algol error message will be displayed. At this point one can use the procedure described in Appendix B to get back to the signal processing interpreter, but all of the stack's vector registers will be reinitialized. (It is planned to modify the program in the near future so that read errors will only result in an attempt to reread.)

The following types of commands are available:

a) operations on a single vector
b) operations on a pair of vectors
c) stack manipulations
d) exiting from the signal processing interpreter.

If the operation has a vector argument, then the X vector is used as the argument. Similarly, if the operation has a single vector result, the result is placed in X. If a vector function of a vector argument is evaluated, the previous value of the X vector (i.e. the argument) is destroyed unless some special action is taken to save it, such as storing it in S. Figure 1 shows the stack movements for this case.

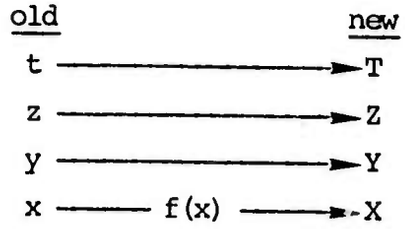


Figure 1. Stack movements for a vector function of a vector argument

If the operation is an output operation, such as printing or plotting, the X vector is used, and the entire stack is left unchanged. If the operation is a call on a function generator such as TONE or CHIRP, or a command to accept a vector input from the keyboard, the vector is placed in the X vector register, destroying the previous contents of X.

If the operation evaluates a vector function of a pair of vectors, then X and Y are used as the arguments. The result is placed in X, and the stack is lowered. The previous contents of X and Y are lost. These actions are shown in Figure 2.

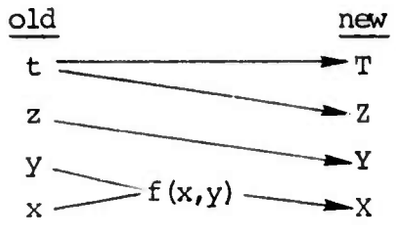


Figure 2. Stack movements for a vector function of two vector arguments

For most functions of two vector arguments, the dimensions of the two argument vectors are first automatically made equal by extending the shorter one with zeros to the dimension of the longer one. The only exception at present is the command APPEND X TO Y, which leaves a resultant of dimension NX + NY in X.

The stack operation commands are identical to those of the HP-35, except for ROLL UP, which is an exact inverse to ROLL DOWN. Their effects are shown in Figures 3-8.

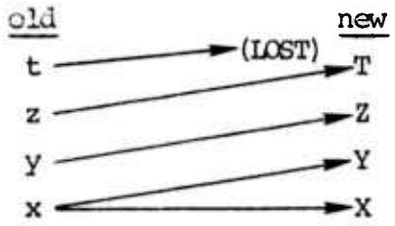


Figure 3. Stack movements for the command 'enter'

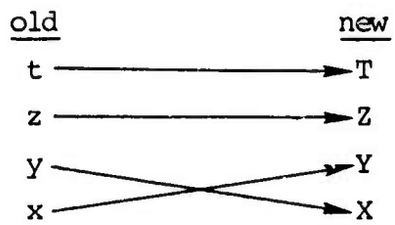


Figure 4. Stack movements for the command 'xy interchange'

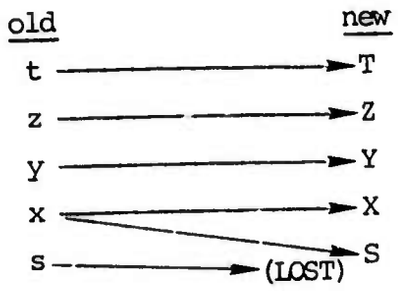


Figure 5. Stack movements for the command 'store'

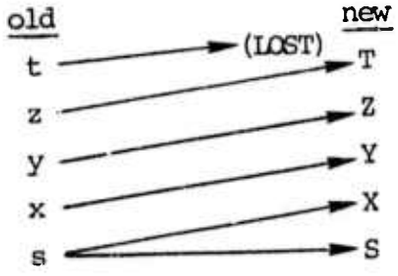


Figure 6. Stack movements for the command 'recall'

The commands 'clear x' and 'clear' respectively set either the x register or all registers to zero. These two commands are not yet implemented.

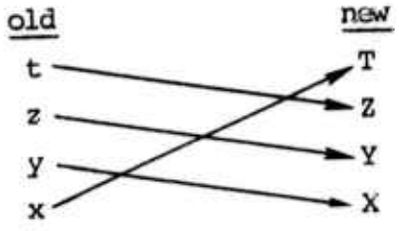


Figure 7. Stack movements for the command 'roll down' (not yet implemented)

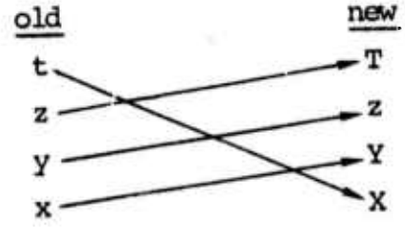


Figure 8. Stack movements for the command 'roll up' (not yet implemented)

PLANNED EXTENSIONS

Several extensions of the signal processing interpreter are likely in the near future. In order of probable implementation they are:

a) Increasing the maximum vector dimension.

b) Implementing the commands which are currently recognized but not yet implemented.

c) Providing a re-read when read errors occur.

d) Providing additional addressable vectors with addressable store and recall commands.

e) Providing for user-definable compound commands and synonyms for present commands.

f) Providing for ambiguity function calculation and the use of more sophisticated plot programs.

BIBLIOGRAPHY

1. Katzan, Harry, Advanced Programming, Van Nostrand Reinhold, 1970 (Chapter 4, Basic Compiler Methods).

2. Wegner, P., An Introduction to Stack Compilation Techniques, Introduction to System Programming, Chapter 7, pp. 101-121, edited by Peter Wegner, Academic Press, New York, 1964.

3. HP-35 Operating Manual, Hewlett Packard, Cupertino, California.

4. Bornstein, I., Demand Terminal User's Guide, NUC, Pasadena.

5. Uhrich, M. L., Fast Fourier Transform Without Sorting, IEEE Transactions on Audio and Electroacoustics, Vol. AU-17, June 1969, pp. 170-172.

6. Rabiner, L. R., R. W. Schafer, and C. M. Rader, The Chirp Z-Transform Algorithm, IEEE Transactions on Audio and Electroacoustics, Vol. AU-17, 1969, pp. 86-92.

APPENDIX A
COMMAND LIST

This table lists all commands which are recognized by the signal processing interpreter, even if the command is not yet implemented. Commands which are recognized but not yet implemented are followed by (N).

OPERATIONS ON A SINGLE VECTOR:

- fft
- reversal
- chirp generator
- tone generator
- czt
- conjugation
- input vector
- print vector
- plot complex vector
- picture line
- set x dimension
- function generator
- mirror
- sum absolute values
- pointwise square magnitude
- constant function
- raise to the power
- absolute value
- plot(N)
- noise generator(N)
- quantizer(N)
- histogram(N)
- correlation parameter(N)
- hadamard transform(N)

cosine transform(N)
kl transform(N)
rectangular to polar(N)
threshold(N)
binary plot(N)
shift(N)
print max(N)
scale(N)
autoscale(N)
exponential(N)
modified log(N)

OPERATIONS ON A PAIR OF VECTORS:

multiplication
periodic convolution
subtract
append x to y
periodic correlation(N)
correlation(N)
golay sequence generator(N)
add(N)

STACK OPERATIONS:

enter
store
recall
xy interchange
clear(N)
clear x(N)
roll down(N)
roll up(N)

APPENDIX B

START UP AND SIGN OFF PROCEDURES

This section assumes that the signal processing interpreter is to be used from a demand terminal on a Univac 1108 operating under EXEC VIII. In the examples below, user entries are underlined to distinguish them from the computer's replies.

A) START UP PROCEDURE:

UD108A (enter the terminal's SITEID)

UNIVAC 1108 TIME/SHARING EXEC VERS 27.20.164 12 (the system identifies itself to the user)

@RUN ABCD (enter a short run card)

@ASG,AZ 60*ARPA. (assign the program file)

READY (the system responds when the file is available)

@ADD,L 60*ARPA.BEGIN (bring in the remaining needed executive control commands automatically)

@ASG,AZ 60*PICSET1. (the file containing the USC master picture set 1 is assigned)

READY

@USE 10.,60*PICSET1. (the name 10 is associated with the picture set file)

READY

@XQT 60*ARPA.SPIN3 (the signal processing interpreter is called)

B) SIGN OFF PROCEDURE:

'EXIT' (the user exits from the signal processing interpreter)

EXIT FROM SIGNAL PROCESSING INTERPRETER

EXIT ALGOL-SIMULA LIBRARY

EXECUTION TIME 2.2 SECONDS - LIBRARY OF SEPT 19,1969

@FIN (the user terminates the run)

(accounting information is then received by the user)

C) RESTART AFTER AN ERROR EXIT:

```
@XQT 60*ARPA.SPIN3
SIGNAL PROCESSING INTERPRETER,VERSION 1
NX=  1
PLEASE ENTER NEXT COMMAND
```

APPENDIX C
DETAILED COMMAND DESCRIPTION

This appendix only describes the commands which are currently implemented. Commands which are currently recognized but not yet implemented will be documented when available.

FFT replaces X by its FFT of length NX. The previous contents of X is destroyed. The current X dimension, NX, must be a power of two.

REVERSAL X is replaced by X(0), X(NX-1), X(NX-2), . . . X(1), i.e. each index is replaced by its negative modulo NX. This is the same type of reversal which is produced by two forward discrete Fourier transforms in succession.

CHIRP GENERATOR generates a complex chirp of length NX, i.e. $X(k)=e^{i\pi k^2/NX}$ for k=0, 1, . . . NX-1. The previous contents of the X vector is destroyed.

TONE GENERATOR generates uniformly spaced samples of a sine wave. The number of samples, frequency, and sampling increment are requested of the user. The resulting sinusoid is placed in the real part of the X vector, and zero is placed in the imaginary part of the X vector. The previous contents of the X vector is destroyed.

CONJUGATION replaces the X vector by its complex conjugate.

INPUT VECTOR allows the user to manually input a complex vector. First he is asked for the dimension of the vector he would like to input. He then enters each term in the form REAL,IMAGINARY followed by a carriage return. If the number to be entered is purely real, the imaginary part can be replaced by an extra carriage return. The previous value of the X vector is destroyed, and the entered vector is placed in X. The real and imaginary parts may have up to 5 digits preceding the decimal point and up to 3 digits following the decimal point.

PRINT VECTOR prints the X vector.

PLOT COMPLEX VECTOR normalizes the X vector to a maximum absolute value of 1, prints the maximum absolute value prior to normalization, prints the normalized

X vector, and simultaneously plots the real and imaginary parts of the normalized X vector.

PICTURE LINE replaces the X vector by a section of a line of picture from the USC Image Coding Master Set No.1. The user is asked to specify the picture number (1-9) and the line number (1-256). The length of the section written into X is the smaller or 256 or NX.

SET X DIMENSION allows the user to reset NX. Immediately following the command, he enters an integer (1-512). If the integer entered is smaller than the previous value of NX, it replaces the previous value and the current dimension of the X vector is reduced. If the integer entered is larger than the previous value of NX, then the X vector is filled out with zeros to the new length, and the entered integer is the new value of the dimension, NX.

FUNCTION GENERATOR at present only generates a linear ramp, i.e. X(k)=k for k=0, 1, . . . NX-1. The previous contents of the X vector is destroyed.

MIRROR replaces NX by 2(NX)-1. The first NX entries of X (i.e. X(0), . . . X(NX-1)) are left unchanged. For k=1, . . . NX-1 X(NX-1+k) is set equal to X(NX-k). In other words, the sequence is extended to length 2(NX)-1 so as to have mirror symmetry (with indices modulo 2(NX-1)) about the zero index point.

SUM ABSOLUTE VALUES prints out the sum of the absolute values of X(0), . . . X(NX-1).

POINTWISE SQUARE MAGNITUDE replaces each of X(0), . . . X(NX-1) by the square of its absolute value. The previous contents of the X vector is destroyed.

CONSTANT FUNCTION lets the user enter a complex number in the format REAL IMAGINARY, and each of X(0), . . . X(NX-1) is set equal to the entered value. Note that a space is used here as the delimiter between the real and imaginary parts. The previous contents of the X vector is destroyed. Other than the required space, the input is free format.

RAISE TO THE POWER permits the user to enter a real number (with decimal point), and each component of the X vector is raised to that power. The previous contents of the X vector is destroyed.

ABSOLUTE VALUE replaces each element of the X vector by its absolute value, destroying previous contents of the X vector.

MULTIPLICATION extends the shorter of the X and Y vectors to the dimension of the longer one. Then X is replaced by the pointwise product of the two vectors, and the stack is lowered. The original values of the X and Y vectors are destroyed.

PERIODIC CONVOLUTION assumes that at least the larger of NX,NY is a power of two. The shorter vector is automatically padded out with zeros to the larger dimension, and X is replaced by the periodic convolution. The stack is lowered, and the original values of the X and Y vectors are destroyed.

SUBTRACT pads out the shorter of the X and Y vectors to the larger of the two dimensions, and X is replaced by Y-X. The stack is lowered, and the original values of the X and Y vectors are destroyed.

APPEND X TO Y the X vector is replaced by X appended to Y, i.e. by Y(0), . . . Y(NY-1), X(0), . . . X(NX-1), and the new value of NX is the sum of the previous X and Y dimensions. The stack is lowered, and the previous values of the X and Y vectors are lost.

APPENDIX D
SOFTWARE DESCRIPTION

The software associated with the signal processing interpreter is contained in two UNIVAC 1108 Fastrand files; a program file called 60*ARPA., and a data file called 60*PICSET 1.

The data file contains the USC Image Coding Master Set No.1, comprising nine pictures, each 256 lines long. Each line consists of 256 words, and each word is an integer in the range 0-255.

The program file contains symbolic, relocatable, and absolute versions of the signal processing interpreter SPIN3 and its associated external Algol procedures and Fortran subroutines. The program file also contains assorted test programs and file manipulation programs which were used in the course of developing the software.

The Algol programs are all in UNIVAC 1108 Algol, and the Fortran subroutines are all in UNIVAC 1108 Fortran V. The signal processing interpreter and the external procedures it uses are shown in Table 1.

PROGRAM NAME	TYPE	FUNCTION PERFORMED
SPIN3	Algol	signal processing interpreter, main program
ATAB	Algol	procedure for command recognition via table lookup
FFTSP	Fortran	complex FFT with results in normal order
MULT	Fortran	subroutine for pointwise multiplication of two complex vectors
REV	Fortran	subroutine for periodic reversal of a complex vector
CHIRP	Fortran	subroutine to generate a complex chirp
TONE	Fortran	subroutine to generate uniformly spaced samples of a sinusoid
STAR	Fortran	subroutine to replace a vector by its complex conjugate
PCONV	Fortran	subroutine to perform the periodic convolution of two complex vectors
READV	Fortran	subroutine to let user manually enter a complex vector

CPLOT	Fortran	subroutine to normalize, print, and plot a complex vector
LINE	Fortran	subroutine to read a picture line from a suitably structured mass storage file
CZT	Algol	procedure to perform a discrete Fourier transform via the Chirp Z-Transform algorithm
BEGIN	ELT	element to add control commands to the run stream to get ready to use the signal processing interpreter together with the picture data file

DESCRIPTION OF THE MAJOR PROGRAMS:

SPIN3

SPIN3 is the signal processing interpreter main program. Upon first being entered, the program prints an identification message. It then jumps to the statement labeled START. The successful completion of an action by the main program (possibly including a call to an external procedure) also results in a jump to START.

When START is reached, the current dimension of the X vector is printed, together with a request to the user to enter the next command. At this point, control is transferred to the external Algol procedure ATAB, which reads a string variable (up to 25 characters enclosed in single quotes) and returns the variables ICOM and CHECK to the main program. The boolean variable CHECK is set equal to true if the string is recognized by ATAB, otherwise it is set equal to false. If CHECK is false, then the message "COMMAND NOT RECOGNIZED" is printed, and the program again jumps to START. If CHECK is true, then the integer ICOM is used to determine to which of the labels L1, L2, etc. control is to be transferred. If the command is recognized by ATAB, but not yet implemented in SPIN3, a jump is performed to the statement labelled SOON; the user is notified that the command is not yet implemented; and the program jumps to START.

When one of the statements labelled L1, L2, etc. is reached, the requested action is performed by the main program or an external procedure or both. In general, stack operations and short computations are performed by the main

program, while relatively complicated operations are performed by external procedures. In all cases, the external procedures return information to the main program only through their argument lists. While this method pays some penalty in address computation time, it greatly facilitates the addition of new procedures, changing the maximum array dimensions, and other modifications as additional requirements are placed on the signal processing interpreter. The vectors, X, Y, Z, T, S currently have maximum dimension 512 complex, but this parameter could be easily increased if desired.

ATAB

ATAB is an Algol procedure which reads and recognizes signal processing commands. A string variable S is read, and is compared with each of the variables in the string array T. If a match is found, the integer ICOM is set equal to the corresponding index in T, and the boolean variable CHECK is set equal to true. If no match is found, CHECK is set equal to false.

FFTSP

FFTSP is a Fortran subroutine to perform an FFT on a complex vector having a dimension which is a power of two. It is a minor modification of Uhrich's FFT algorithm [5]. The main changes from Uhrich's algorithm are that communication is only through the argument list, and that $N=2^m$ is specified in the call rather than m.

CZT

CZT is a straightforward implementation of the Chirp Z-transform algorithm described by Rabiner et al [6]. It may be used to perform the discrete Fourier transform of a vector of arbitrary dimension (up to the maximum specified in the main program and the FFT subroutine).

APPENDIX E

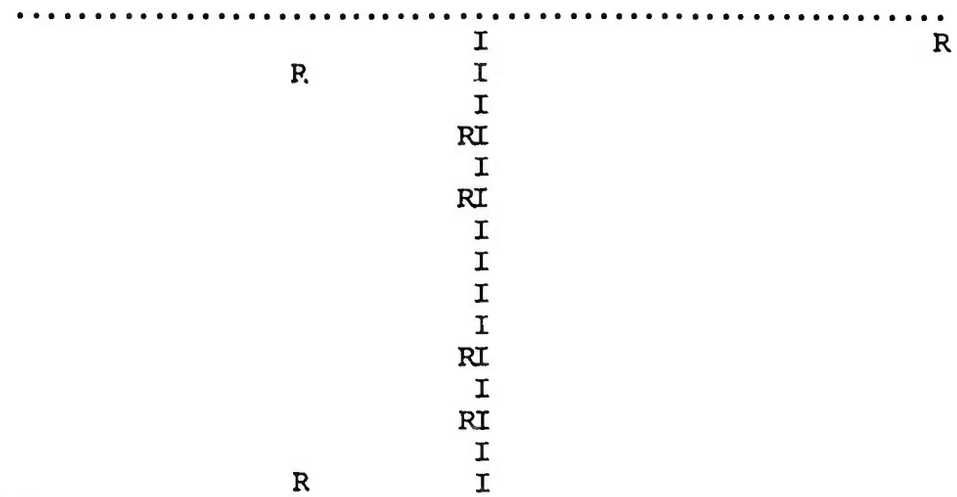
SAMPLE RUN

(USER ENTRIES ARE UNDERLINED, COMMENTS ARE IN PARENTHESES)

```
UD108A  (user enters his terminal's siteid)
UNIVAC 1108 TIME/SHARING EXEC  VERS 27.20.225 01
@RUN ABCD  (user enters a short run card)
DATE: 062273     TIME: 142232
@ASG,AZ 60*ARPA.  (user assigns the program file)
READY
@ADD,L 60*ARPA.BEGIN  (this command adds a prestored set of control commands
                       to the run stream to complete the initialization)
@ASG,AZ 60*PICSET1.  (the data file containing USC Master Picture Set 1 is
READY                 assigned)
@USE 10,60*PICSET1.  (the internal filename 10. is identified with the external
READY                 filename 60*PICSET1.)
@XQT 60*ARPA.SPIN3  (the signal processing interpreter is called)
SIGNAL PROCESSING INTERPRETER,VERSION 1
NX=
          1
PLEASE ENTER NEXT COMMAND
'FUNCTION GENERATOR'  (requests generation of a linear ramp)
ENTER DESIRED FUNCTION LENGTH
8
NX=
          8
PLEASE ENTER NEXT COMMAND
'PRINT VECTOR'  (prints the contents of the X vector)
 INDEX      COMPLEX VALUE
    0       .000       .000
    1      1.000       .000
    2      2.000       .000
    3      3.000       .000
    4      4.000       .000
    5      5.000       .000
    6      6.000       .000
    7      7.000       .000
 FINISHED PRINTING VECTOR
NX=
          8
```

```
PLEASE ENTER NEXT COMMAND
'STORE'  (stores the contents of the X vector in the S vector register)
NX=

           8
PLEASE ENTER NEXT COMMAND
'FFT'  (X is replaced by its discrete Fourier transform)
NX=

           8
PLEASE ENTER NEXT COMMAND
'PRINT VECTOR'  (prints the contents of the X vector)
 INDEX      COMPLEX VALUE
    0     28.000       .000
    1     -4.000      9.657
    2     -4.000      4.000
    3     -4.000      1.657
    4     -4.000       .000
    5     -4.000     -1.657
    6     -4.000     -4.000
    7     -4.000     -9.657
 FINISHED PRINTING VECTOR
NX=

           8
PLEASE ENTER NEXT COMMAND
'RECALL'  (recalls S into X, so X now contains the linear ramp)
NX=

           8
PLEASE ENTER NEXT COMMAND
'MIRROR'  (generates a symmetric function of length 2(NX)-1 by appending
NX=        to X its mirror image about the zero index point)

          15
PLEASE ENTER NEXT COMMAND
'PRINT VECTOR'  (prints the contents of the X vector)
 INDEX      COMPLEX VALUE
    0      .000       .000
    1     1.000       .000
    2     2.000       .000
    3     3.000       .000
    4     4.000       .000
    5     5.000       .000
    6     6.000       .000
    7     7.000       .000
    8     7.000       .000
    9     6.000       .000
   10     5.000       .000
   11     4.000       .000
   12     3.000       .000
   13     2.000       .000
   14     1.000       .000
 FINISHED PRINTING VECTOR
NX=
```

```
          15
PLEASE ENTER NEXT COMMAND
'CZT'  (replace X by its discrete Fourier transform, computed via the
NX=     Chirp Z-Transform Algorithm)

          15
PLEASE ENTER NEXT COMMAND
'PRINT VECTOR'  (print the contents of the X vector)
 INDEX      COMPLEX VALUE
    0    56.000      .000
    1   -22.881      .000
    2     -.261      .000
    3    -2.618      .000
    4     -.300      .000
    5    -1.000      .000
    6     -.382      .000
    7     -.558      .000
    8     -.558      .000
    9     -.382      .000
   10    -1.000      .000
   11     -.300      .000
   12    -2.618      .000
   13     -.261      .000
   14   -22.880      .001
 FINISHED PRINTING VECTOR
NX=

          15
PLEASE ENTER NEXT COMMAND
'PLOT COMPLEX VECTOR'  (normalize X to a maximum absolute value of 1.0, print and plot)
REAL PLOT USES R,IMAGINARY PLOT USES I AS CHARACTER
 MAX ABS VALUE IN ARRAY=    56.00000
     1          1.00000          .00000
     2          -.40859          .00000
     3          -.00467          .00000
     4          -.04675          .00000
     5          -.00535          .00000
     6          -.01786          .00000
     7          -.00682          .00000
     8          -.00997          .00000
     9          -.00997          .00000
    10          -.00682          .00000
    11          -.01786          .00000
    12          -.00535          .00000
    13          -.04675          .00000
    14          -.00467          .00000
    15          -.40858          .00002
```



```
NX=
          15
PLEASE ENTER NEXT COMMAND
'EXIT'  (exit from the signal processing interpreter)
EXIT FROM SIGNAL PROCESSING INTERPRETER
EXIT ALGOL-SIMULA LIBRARY
EXECUTION TIME 000000.918 SECONDS - LIBRARY OF   SEPT 19, 1969
@FIN  (terminate the run and call the accounting routines)


    RUNID:  ABCD     ACCOUNT:  XXXXXXXX       PROJECT:  XXXXXXXXXXXX

CPU TIME    00:00:0;.102 @ $.300/SEC    = $    .33

CORE SECS       81       @ $.300/CBS    = $    .24

I/O REFS        51       @ $.300/CRF    = $    .15

WDS XFD(E2)    902       @ $.000/WRD    = $    .00

CARDS IN        21       @ $.001/CRD    = $    .02

PAGES            5       @ $.040/PGE    = $    .20

TOTAL RUN COST (TENTATIVE)              = $    .94

    INITIATION TIME:   14:22:31-JUN 22,1973      VERSION:   27.20.225 01


                    ****LINE INACTIVE****
```

APPENDIX C

APPLICATION OF MAXIMUM ENTROPY ESTIMATION TO IMAGE TRANSMISSION

The intensity of an image line may be represented by a nonnegative function I(x), on the interval [0, L]. The Fourier transform of I is nonnegative definite, and hence may be regarded as an autocorrelation function. Reconstruction of the image line from a truncated version of the Fourier transform of the image is thus mathematically equivalent to the determination of a spectral density function from a truncated autocorrelation function. If desired, I(x) may be replaced by its mirror symmetrized version g(x) = I(|x|). The Fourier transform of g is a real autocorrelation function.

For wide-sense stationary random processes with all-pole spectra, the method maximum entropy provides an accurate means to extrapolate a truncated autocorrelation function and to use the extrapolated autocorrelation function to form a spectral estimator having much finer resolution than that of conventional (Blackman-Tukey type) estimators that use the truncated autocorrelation function.

We propose to address three problems in order to determine the applicability of maximum entropy estimation to reduced redundancy image transmission:

a. Applicability of the all-pole spectrum model.

b. Feasibility of a real-time computational algorithm.

c. Accuracy requirements for hardware to implement a real time computational algorithm.

APPENDIX D

NAVAL UNDERSEA CENTER

San Diego, California 92132

HIGH SPEED SERIAL ACCESS LINEAR TRANSFORM IMPLEMENTATIONS

by

H. J. WHITEHOUSE, J. M. SPEISER, and R. W. MEANS

January 1973

Paper presented at
All Applications Digital Computer (AADC) Symposium
Orlando, Florida

HIGH SPEED SERIAL ACCESS LINEAR TRANSFORM IMPLEMENTATIONS

H. J. Whitehouse, J. M. Speiser, and R. W. Means
Naval Undersea Center
San Diego, CA 92132

INTRODUCTION

The time required to perform signal processing computations is determined largely by the time required to perform linear transformations on data vectors. The choice of algorithm and implementation structure strongly affects both the processing time and the implementation complexity via the required number of multiplication times and via the data storage and transfer structure.

Any linear transformation that takes a sequence of N data points into a sequence of N transform points may be regarded as a multiplication of a vector by an N × N matrix. A direct implementation that uses a single multiplier requires N^2 multiplication times and N^2 words of storage. The Fast Fourier Transform (FFT) requires a number of multiplications proportional to $N \log_2 N$.

This paper discusses a number of transform implementations that have a simple serial access data flow and a computation time proportional to N. These transform implementations which consist of two complex multipliers and a complex convolver as a module, provide an effective method by which the equivalent of a large number of multipliers can be used in a processing architecture with a simple data flow. The transform structure discussed here may be used for different types of transforms including the Discrete Fourier Transform (DFT) by a change of the constants provided in a local store. If a crossconvolver is used in this structure, it may be used also to perform direct high speed linear filtering and crosscorrelation.

STRUCTURE OF TRANSFORMATIONS

Several types of elementary linear transformations on data vectors may be viewed as building blocks for the implementation of more general linear transformations. Among these elementary transformations are multiplication by a diagonal matrix (Fig. 1), convolution (Fig. 2), and circular convolution (Fig. 3). A transformation of particular importance for signal processing is the discrete Fourier transform. The chirp Z-transform (CZT) algorithm [1] decomposes the DFT into a premultiplication by a discrete chirp, a convolution with a discrete chirp, and a postmultiplication by a discrete chirp (Fig. 4), since

$$G_n = \sum_{m=0}^{N-1} e^{-i2\pi nm} g_m = e^{-i\pi n^2/N} \sum_{m=0}^{N-1} e^{i\pi(n-m)^2/N} \left(e^{-i\pi m^2/N} g_m \right)$$

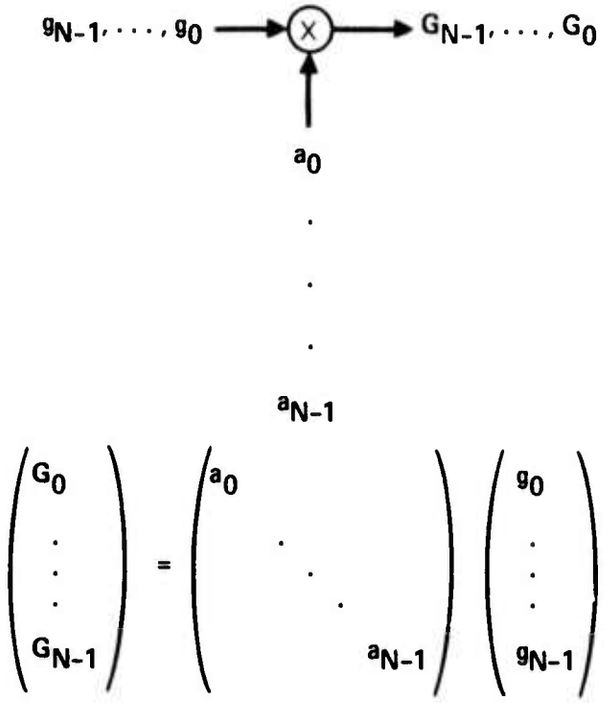


Fig. 1. Multiplication by a diagonal matrix

If the reference functions used in the convolver and the multipliers are varied, the same architecture used to implement the CZT may be used to implement a wider class of linear transforms (Fig. 5).

A linear transform may be implemented by this architecture if and only if its kernel decomposes appropriately. For a continuous transform

$$G(u) = \int_{-\infty}^{\infty} k(u, t)\, g(t)\, dt$$

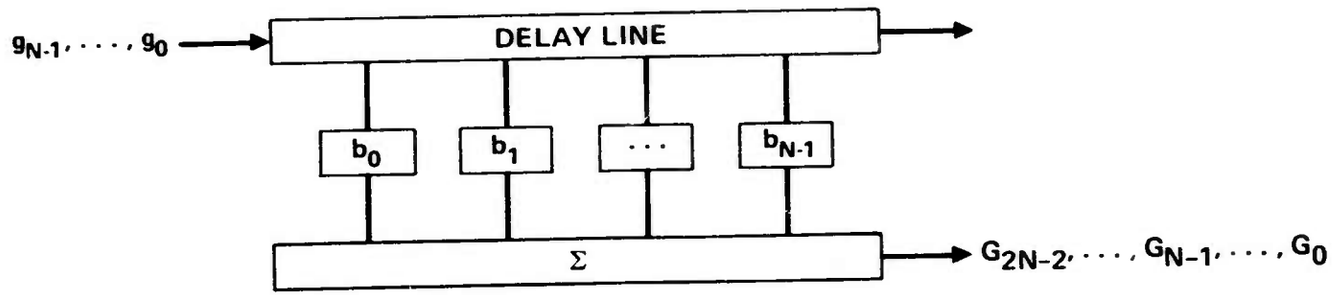
the required decomposition is

$$k(u, t) = a(t)\, b(u-t)\, c(u)$$

if a convolution is to be used as the intermediate operation. Similarly, if a correlation is to be used as the intermediate operation, the required decomposition is $k(u, t) = a(t)\, b(t + u)\, c(u)$.

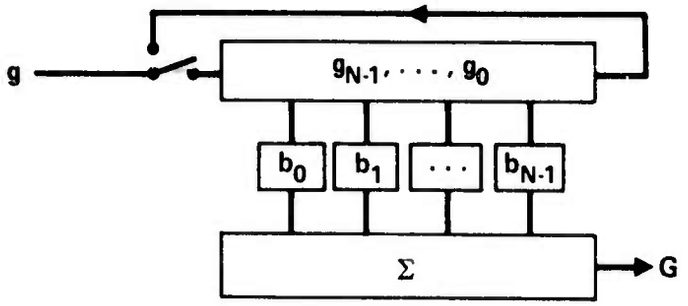
If the kernel is smooth and nonvanishing, a necessary and sufficient condition for decomposition is

$$\frac{\partial \log k}{\partial t} \pm \frac{\partial \log k}{\partial u} \equiv \frac{1}{k}\left(\frac{\partial k}{\partial t} \pm \frac{\partial k}{\partial u}\right) = \text{Any function of t plus any function of u}$$



$$\begin{pmatrix} G_0 \\ G_1 \\ \vdots \\ G_{N-1} \\ \vdots \\ G_{2N-2} \end{pmatrix} = \begin{pmatrix} b_0 & & & \\ b_1 & b_0 & & \\ \vdots & \vdots & & \\ b_{N-1} & b_{N-2} \cdots & b_1, & b_0 \\ \vdots & \vdots & & \\ 0, & 0 \ldots & 0, & b_{N-1}, \ldots, b_0 \end{pmatrix} \begin{pmatrix} g_0 \\ g_1 \\ \vdots \\ g_{N-1} \\ \vdots \\ 0 \end{pmatrix}$$

Fig. 2. Convolution



$$\begin{pmatrix} G_0 \\ G_1 \\ \vdots \\ G_{N-1} \end{pmatrix} = \begin{pmatrix} b_{N-1} & b_{N-2} & \cdots & b_0 \\ b_0 & b_{N-1} & \cdots & b_1 \\ \vdots & & & \\ b_{N-2} & b_{N-3} & \cdots & b_{N-1} \end{pmatrix} \begin{pmatrix} g_0 \\ g_1 \\ \vdots \\ g_{N-1} \end{pmatrix}$$

Fig. 3. Circular Convolution

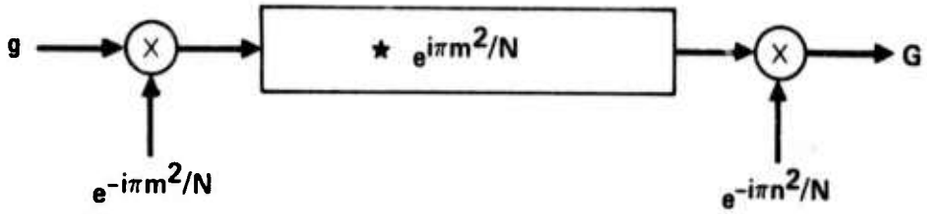
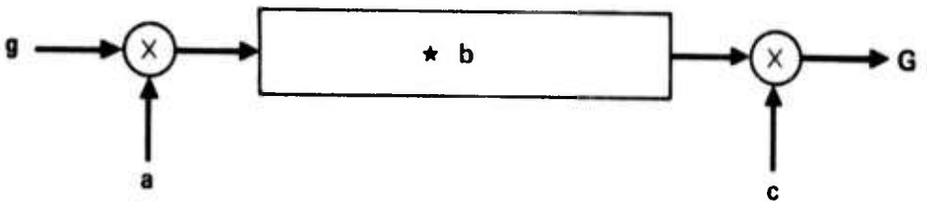


Fig. 4. Chirp Z-transform Implementation of the DFT



$$\underline{G} = \underline{\underline{C}}\,\underline{\underline{B}}\,\underline{\underline{A}}\,\underline{g}$$

★ DENOTES EITHER CONVOLUTION OR CIRCULAR CONVOLUTION

Fig. 5. Architecture Composed of a Compound of Elementary Transformations

An alternative necessary and sufficient condition is

$$\frac{\partial^3 \log k}{\partial t^2 \partial u} \pm \frac{\partial^3 \log k}{\partial t \partial u^2} = 0$$

where the plus sign corresponds to the convolution decomposition and the minus sign corresponds to the correlation decomposition.

For a discrete transformation of the form

$$G_m = \Sigma k_{m,n} g_n$$

the corresponding required factorization for a convolution decomposition is

$$k_{m,n} = a_n b_{m-n} c_m$$

where the necessary and sufficient condition for factorization (assuming that the kernel is nonvanishing) is

$$\bar{k}_{m+1,n} - \bar{k}_{m,n-1} = \text{any function of m plus any function of n}$$

where

$$\bar{k}_{m,n} = \log k_{m,n}$$

An alternative necessary and sufficient condition is

$$w_{m,n} = 0$$

where

$$\bar{k}_{m,n} = \log k_{m,n}$$

$$u_{m,n} = \bar{k}_{m+1,n} - \bar{k}_{m,n-1}$$

$$v_{m,n} = u_{m+1,n} - u_{m,n}$$

$$w_{m,n} = v_{m,n+1} - v_{m,n}$$

If the discrete transform is to be applied to a data vector of length N, then all subscripts are to be interpreted modulo N.

A listing of some of the more useful decomposable transforms is shown on the following page.

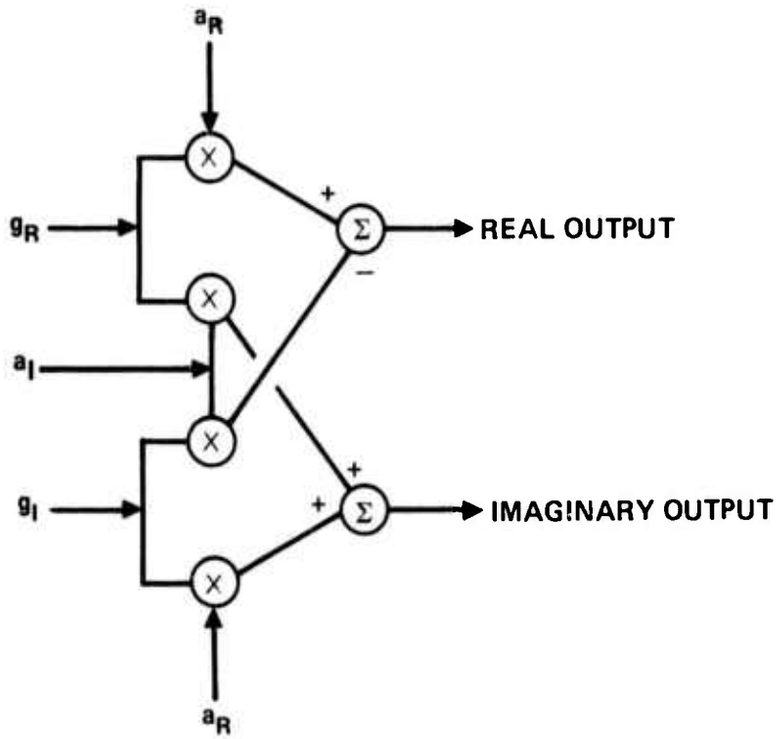
Decomposable Transforms

TRANSFORM	KERNEL
FOURIER	e^{-itu}
LAPLACE	e^{-tu}
HILBERT	$\frac{1}{t-u}$
FOURIER COSINE*	cos tu
FOURIER SINE*	sin tu

*NOT STRICTLY DECOMPOSABLE BUT REALIZABLE WITH SAME HARDWARE AS DECOMPOSABLE TRANSFORMS

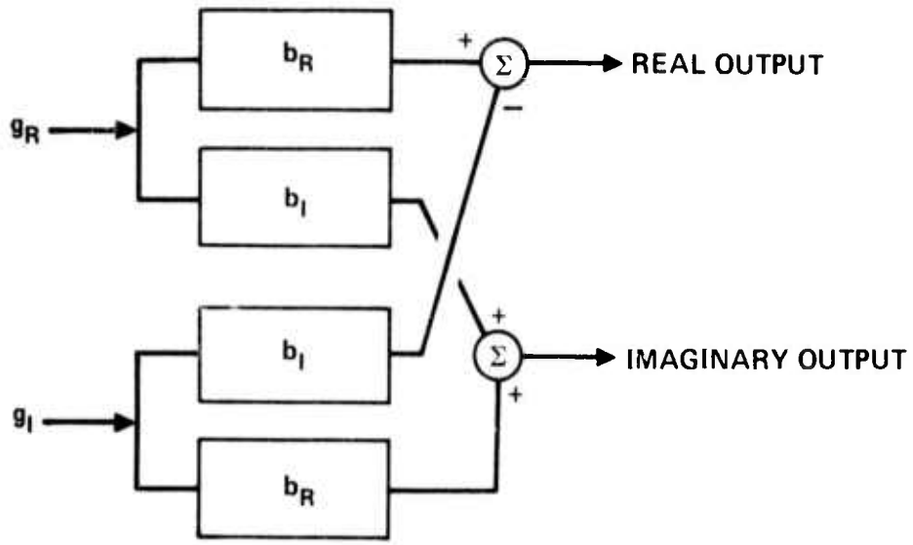
COMPLEX ARITHMETIC

Complex arithmetic operations applied to sampled data may be implemented in parallel or interleaved form. The parallel implementations (Figs. 6 and 7) follow directly



$$g \cdot a = (g_R + ig_I)(a_R + ia_I) = (g_R a_R - g_I a_I) + i(g_R a_I + g_I a_R)$$

Fig. 6. Complex Arithmetic Multiplier (parallel implementation)



$$g \star b = (g_R + i g_I) \star (b_R + i b_I) = (g_R \star b_R - g_I \star b_I) + i\,(g_R \star b_I + g_I \star b_R)$$

Fig. 7. Complex Arithmetic Convolver (parallel implementation)

from the definition of a complex multiplication and a complex convolution. The operation of the complex convolver with interleaved arithmetic (Fig. 8) may be understood by an examination of the operations on the data samples in the even and odd positions in the data vector.

We wish to evaluate

$$g \star b = (g_R + i\,g_I) \star (b_R + i\,b_I) = (g_R \star b_R - g_I \star b_I) + i\,(g_R \star b_I + g_I \star b_R)$$

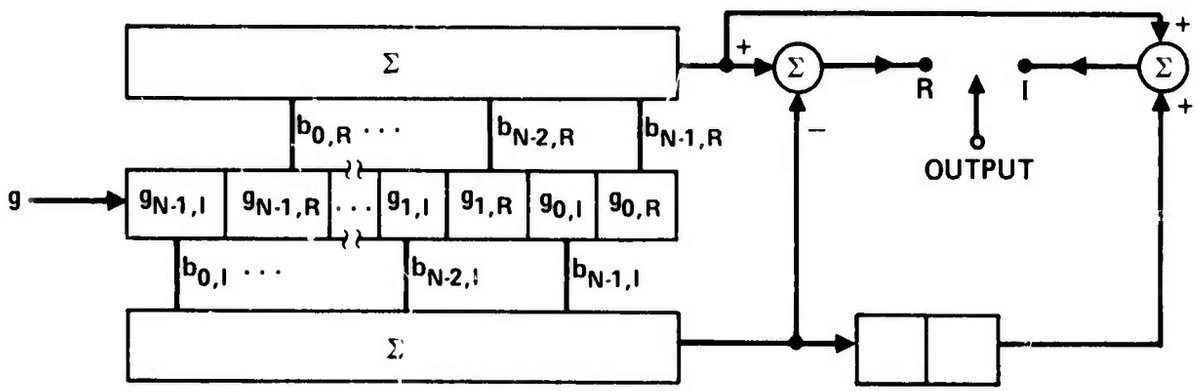


Fig. 8. Complex Convolver with Interleaved Arithmetic

so the desired real term is

$$\sum_n b_{n,R}\, g_{m-n,R} - \sum_n b_{n,I}\, g_{m-n,I}$$

and the desired imaginary term is

$$\sum_n b_{n,I}\, g_{m-n,R} + \sum_n b_{n,R}\, g_{m-n,I}$$

Let

$$x_{2k} = g_{k,R}$$

$$x_{2k+1} = g_{k,I}$$

$$y_{2k} = b_{k,I}$$

$$y_{2k+1} = b_{k,R}$$

At an even shift position, say $m = 2p$,

$$\sum_{k\ \text{even}} x_k\, y_{2p-k} = \sum_s x_{2s}\, y_{2p-2s} = \sum_s g_{s,R}\, b_{p-s,I}$$

$$\sum_{k\ \text{odd}} x_k\, y_{2p-k} = \sum_s x_{2s+1}\, y_{2p-2s-1} = \sum_s g_{s,I}\, b_{p-s-1,R}$$

and the two terms must be combined with a relative delay. At an odd shift position, say $m = 2p+1$

$$\sum_{k\ \text{even}} x_k\, y_{2p+1-k} = \sum_s x_{2s}\, y_{2p+1-2s} = \sum_s g_{s,R}\, b_{p-s,R}$$

$$\sum_{k\ \text{odd}} x_k\, y_{2p+1-k} = \sum_s x_{2s+1}\, y_{2p+1-(2s+1)} = \sum_s x_{2s+1}\, y_{2(p-s)}$$

$$= \sum_s g_{s,I}\, b_{p-s,I}$$

and the two terms must be combined with zero relative delay.

A DISCRETE FOURIER TRANSFORM IMPLEMENTATION

Let us examine the discrete Fourier transform decomposed via the CZT algorithm, and implemented with the parallel form of the complex arithmetic.

$$G_m = \sum_{n=0}^{N-1} e^{-i2\pi mn/N} g_n = e^{-i\pi m^2/N} \sum_{n=0}^{N-1} e^{+i\pi(m-n)^2/N} \left(e^{-i\pi n^2/N} g_n \right)$$

The complex operations that we wish to implement are shown in Fig. 5 where

$$a_n = c_n = e^{-i\pi n^2/N} \qquad \text{for } n = 0, \ldots N-1$$

$$b_m = e^{+i\pi m^2/N} \qquad \text{for } \begin{cases} m = 0, \ldots 2N-2 \text{ if correlation is used} \\ m = 0, \ldots N-1 \text{ if circular correlation is used} \end{cases}$$

When the parallel implementation of the complex arithmetic is used, the DFT architecture of Fig. 9 results. If the data vector dimension N is even, then $b_{m+N} = b_m$, and a circular convolution of length N may be used in place of the convolution of length 2N-1.

CONVOLUTION IMPLEMENTATION WITH TRANSVERSAL FILTERS

The convolution operation may be implemented in highly parallel form as a transversal filter. Such a filter is shown in Fig. 3 with the output of the filter connected back to the input so that it may be used for circular convolution. Various acoustic transversal filters [2] have been described that are suitable for use in a fixed transform implementation if the signal processing can proceed without interruption during each block of the input signal. If interruption of the signal processing is required, nonacoustic transversal filters may be employed [3]. The three types of filters which will be examined here employ, respectively, acoustic surface waves, charge transfer phenomena, and signal propagation in digital shift registers.

Acoustic Surface Waves

For signal processing at sample rates above 10 MHz, acoustic surface waves are attractive. Figure 10 shows a possible surface wave electrode configuration. Aluminum electrodes are usually deposited on a piezoelectric material such as ST quartz, and they act to launch and receive the acoustic waves. Each transducer is composed of a pair of interdigitated electrodes and can act as a tap the weight of which is proportional to the length of the interdigitation which intercepts the acoustic wave. Since the launch transducers are usually many wavelengths long, the resulting waves are well collimated and several filters may be implemented simultaneously on a single substrate. Since both the registration of the multiple filters and the weight of the filter taps are photo-lithographically determined, good uniformity and repeatability are possible.

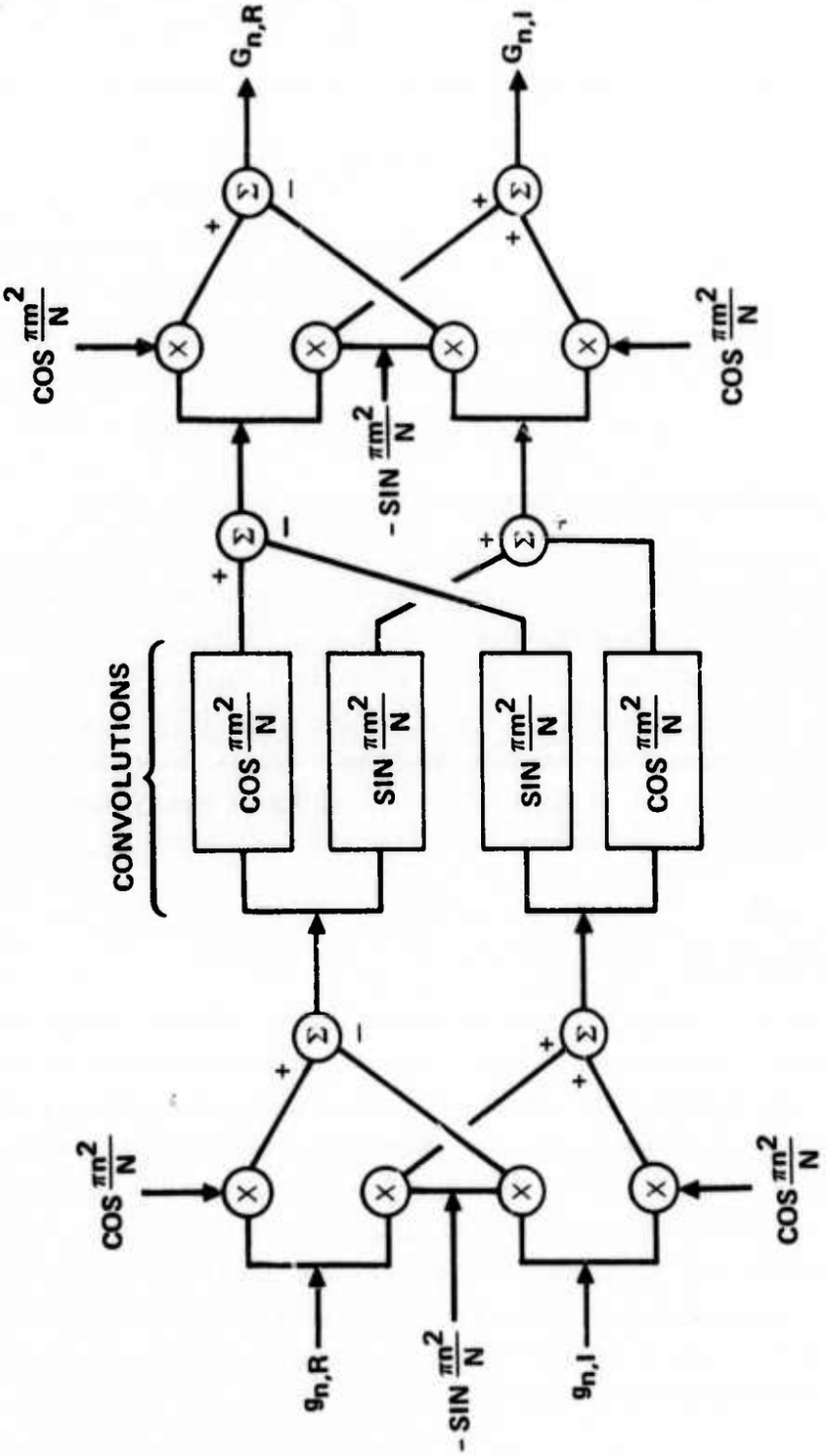


Fig. 9. DFT via CZT Algorithm with Parallel Implementation of Complex Arithmetic

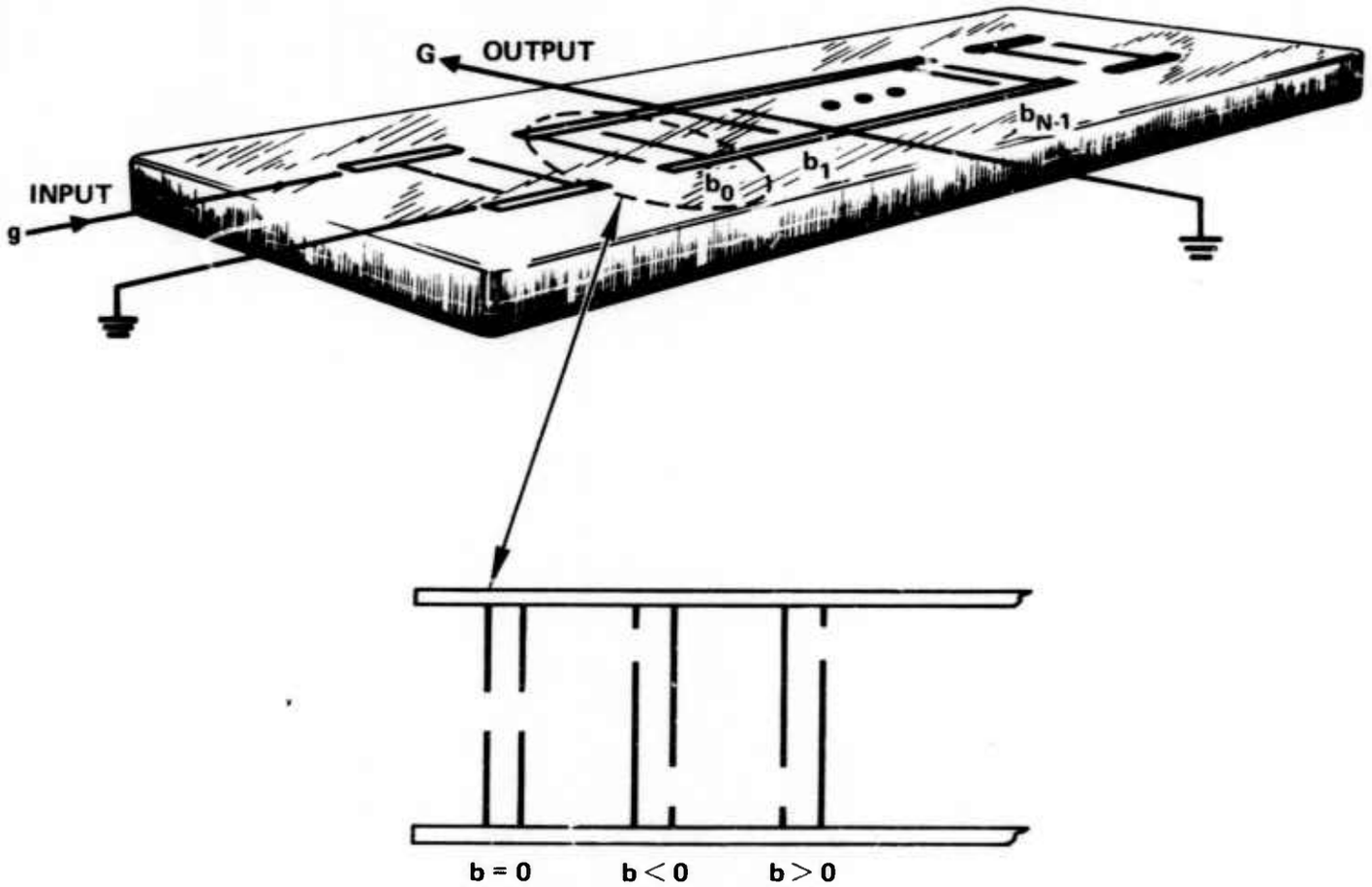


Fig. 10. Surface Wave Transversal Filter

When it is desirable to implement more than one transform with the same hardware, then programmable filters are required. These filters must have the weighting function specified in advance after which they operate as ordinary transversal filters with one exception – at the present time only binary tap weighting is possible so that several programmable filters must be used simultaneously with a binary decomposition of the weighting function.

Surface wave filters with up to 128 digitally switched taps have been built. [4, 5]. A representative programmable filter is shown in Figure 11. It consists of a 128 tap transversal filter with binary taps that are switched under the control of a shift register. Each tap may be connected to either of two busses. The output signal is the difference of the two summed bus signals and is provided by a differential transformer. The signal may be applied at either of transducers A or B, and the weight vector enters at the terminal labeled "Code in."

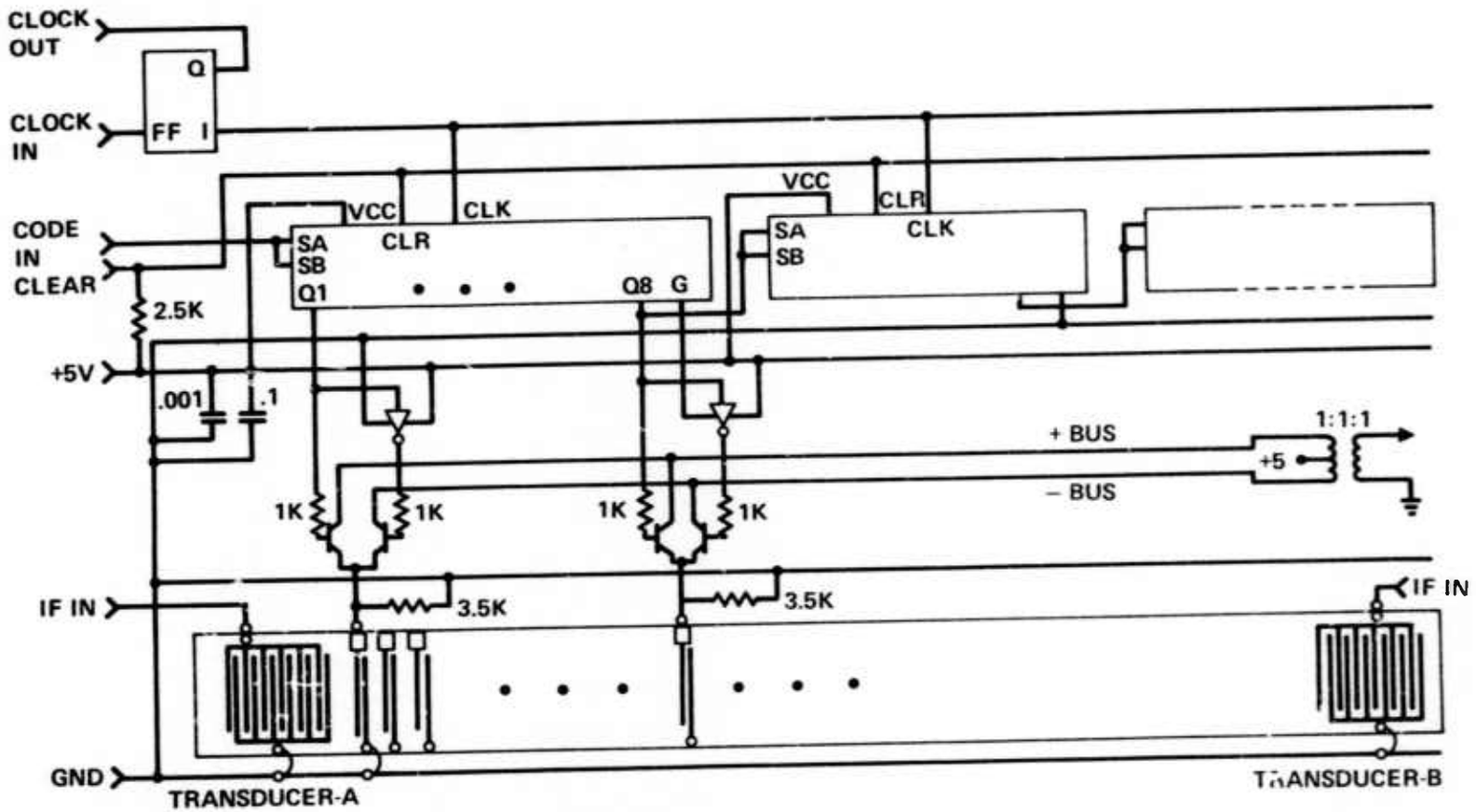
Charge Transfer Devices

For signal processing at sample rates up to 10 MHz, when short duration interruption of the signal processing may occur, charge transfer devices such as charge coupled devices (CCD) and bucket brigade devices offer many advantages. These devices are fabricated by MOS technology from silicon and are small in size. There are, however, two limitations: there is charge transfer inefficiency and thermally generated charge combines with the signal so that the storage time of the signal is less than a second at room temperature. Since the carriers representing the signal are moved through the device by clock voltages applied to an overlying set of electrodes, the propagation of the data signal may be interrupted by stopping the clock. Figure 12 adapted from [6] shows a three phase CCD transversal filter and the method of sensing the charge. Figure 13 shows a 21 sample chirp filter [7].

A programmable CCD filter is currently under development by General Electric for the Naval Undersea Center. This filter will be built using a module size of 64 samples and will have a 4 MHz sample rate. Its construction is such that either the binary or the analog signal may be considered as the input and the other signal considered as the weight. Unlike the fixed reference CCD filter previously discussed, this filter has provision for cascading of modules without degradation of the signal by charge transfer inefficiency.

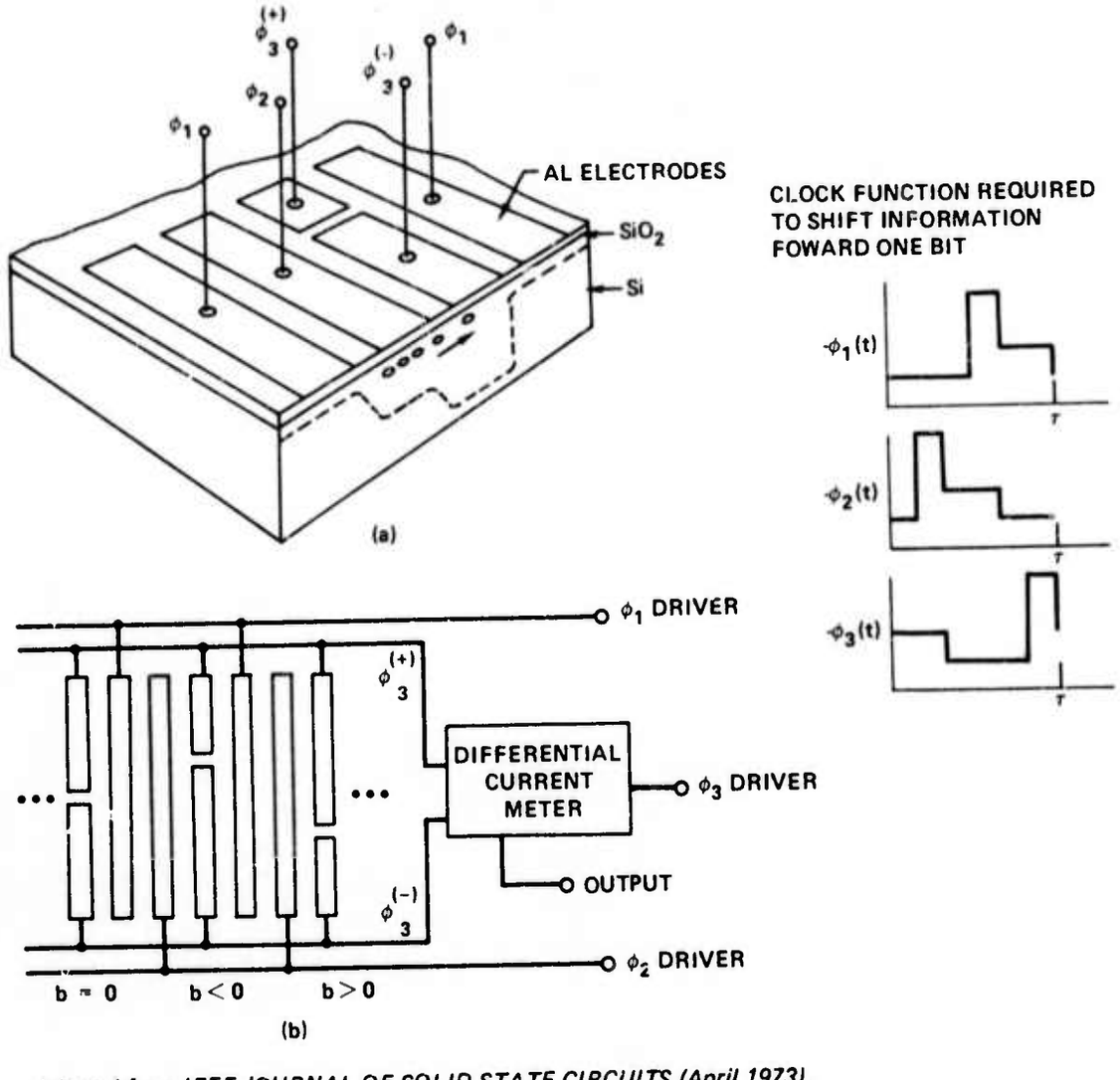
Binary Shift Register Crossconvolvers

When signal processing must be interrupted indefinitely, semiconductor shift register equivalents of programmable transversal filters are most suitable. Such a convolver module (Fig. 14), has been developed by TRW for NavElex under the name of LSI Correlator [8]. This device may be considered as a binary crossconvolver and its use is similar to the



Reprinted from ROME AIR DEVELOPMENT CENTER Report RADC TR-72 (April 1972), p. 17.

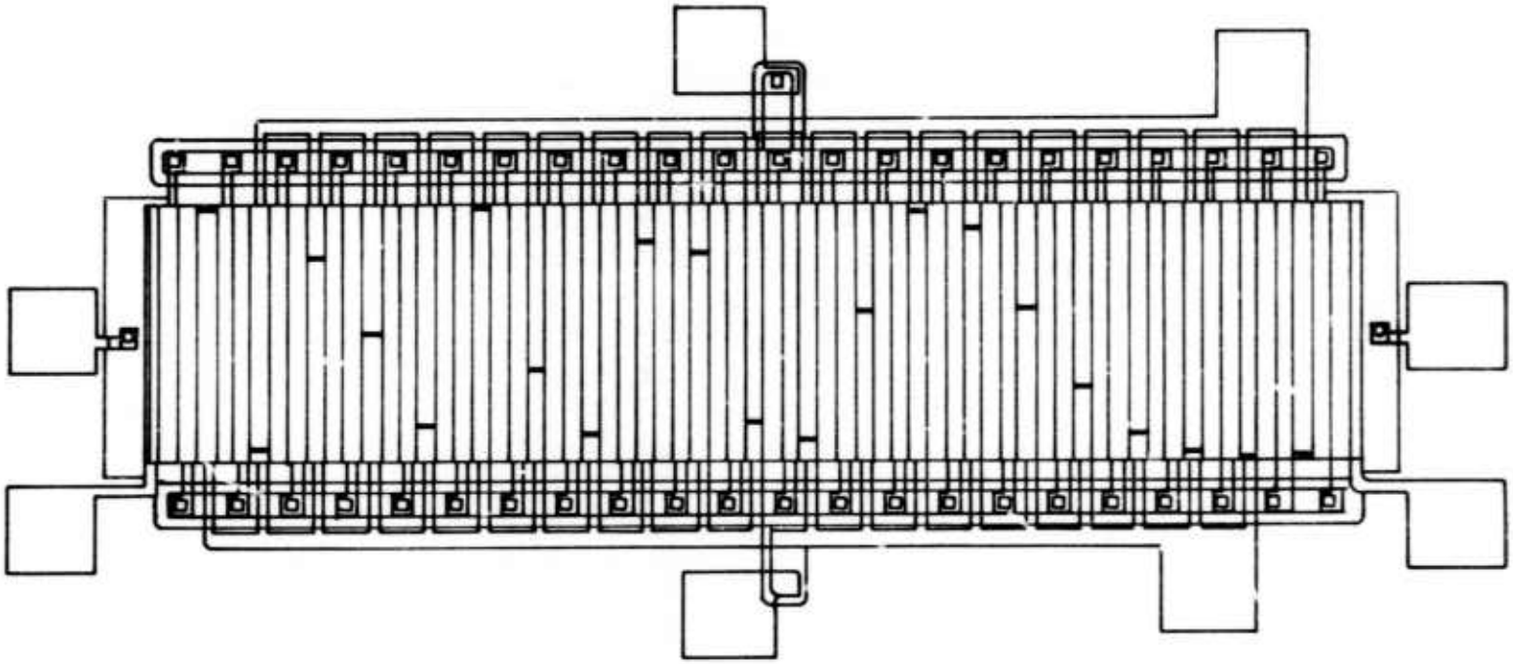
Fig. 11. Surface Wave Programmable Filter



Adapted from IEEE JOURNAL OF SOLID STATE CIRCUITS (April 1973), in press.

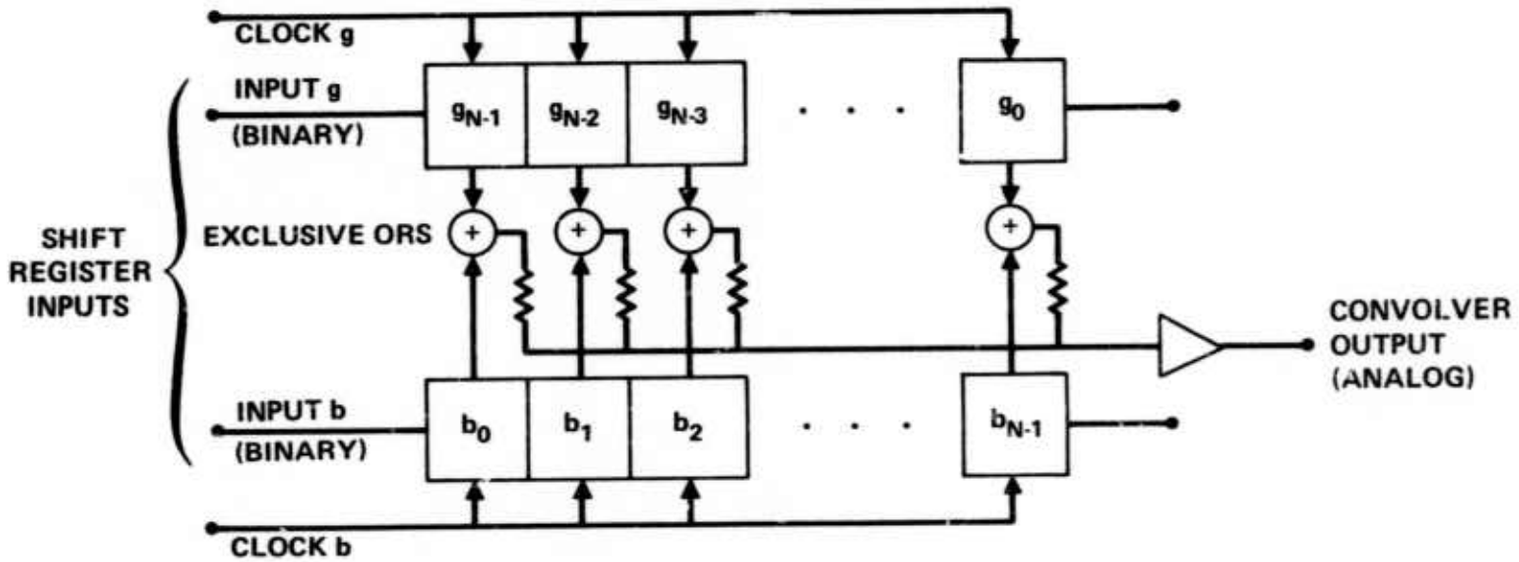
Fig. 12. Charge-Coupled Transversal Filter

programmable transversal filters previously described. However, it is restricted to binary signals and binary weights. The use of multiple binary correlation modules to attain multiple level quantization is described in [3]. Figure 15 shows a configuration suitable for multilevel operation of one of the channels. The LSI correlator is available in either a module size of 64 bits at a 20 MHz clock rate or a module size of 8 bits at a 100 MHz clock rate.



Reprinted from ELECTRONICS LETTERS 8 no. 13 (29 June 1972), p. 329.

Fig. 13. Charge-Coupled Chirp Filter



Adapted from J. L. Buie and D. R. Breuer, "A Large-Scale Integrated Correlator," IEEE JOURNAL OF SOLID STATE CIRCUITS SC-7, no. 5 (October 1972), p. 358.

Fig. 14. Binary Crossconvolver

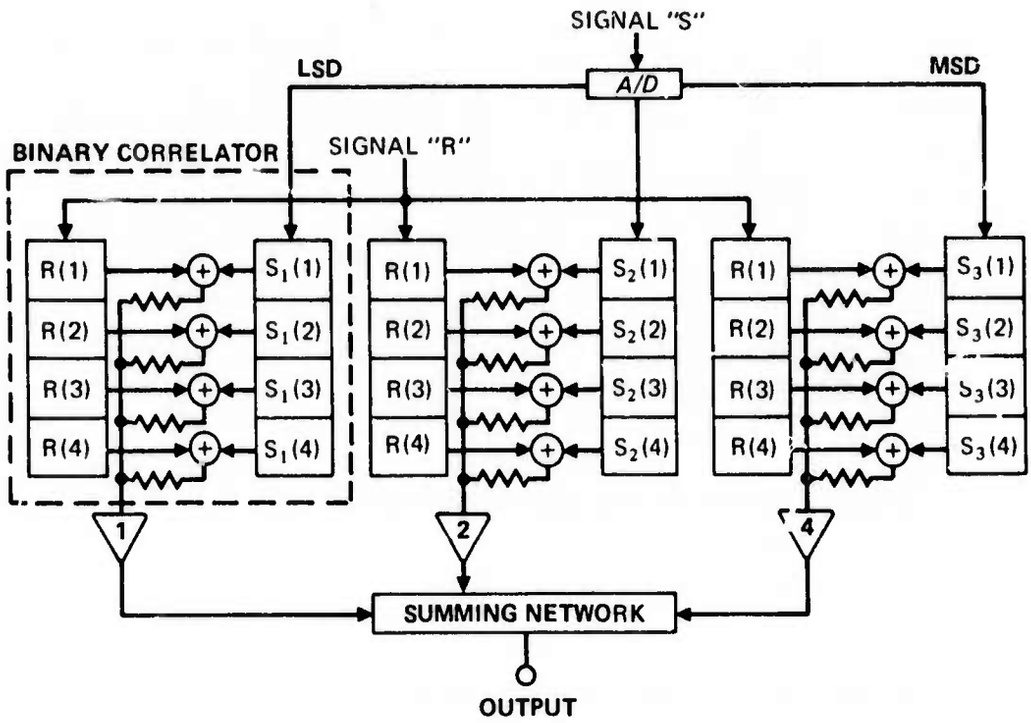
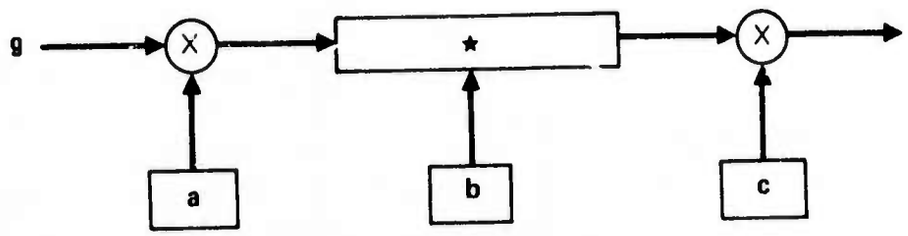


Fig. 15. Correlator Configuration for 3-Bit Signal vs Binary Reference

LINEAR TRANSFORM PROGRAMMABLE SIGNAL PROCESSING ARITHMETIC UNIT

The linear and bilinear operations most needed for signal processing are: matched filtering or replica correlation, the discrete Fourier transform, and crosscorrelation. These transforms represent an excessive computational load for a general purpose digital computer and a heavy computational load even for a digital computer structured for signal processing. The decomposable transform architecture may be used to provide a single module which can perform the above operations in a time proportional to the data block length. Such modules may be configured under computer control to permit not only changes in task assignment but also an exchange among data block length, accuracy, and speed. Figure 16 shows such a module and the contents of the local stores for matched filtering, crosscorrelation, and the discrete Fourier transform.

Serial access delay line memories may be used in the modular building blocks for high speed non-digital implementations. If interruptions of the processing are infrequent, then ultrasonic delay lines are preferred, since they offer large storage capacity, convenient tapping, and low power dissipation. Where frequent processing interrupts are required, charge coupled devices with their controllable clock rate may be used. The corresponding modular building block for digital implementations is a contiguously tapped binary shift register crossconvolver.



FOR LINEAR TRANSFORMS
a, b, c ARE LOCAL STORES

FOR MATCH FILTERS AND CROSS CORRELATION

$a = 1 \qquad b = \{b_j\} \qquad c = 1$

$0 \leq j \leq N\text{-}1$

FOR THE DFT VIA THE CHIRP-Z ALGORITHM

$a = e^{-i\pi n^2/N} \qquad b = e^{i\pi n^2/N} \qquad c = e^{-i\pi n^2/N}$

$0 \leq n \leq N\text{-}1 \qquad 0 \leq n \leq 2N\text{-}2 \qquad 0 \leq n \leq N\text{-}1$

★ denotes convolution

Fig. 16. Linear Transform Programmable SPAU Module

REFERENCES

[1] Rabiner, L. R., R. W. Schafer, and C. M. Rader, "The Chirp-Z Transform Algorithm," *IEEE Transactions on Audio and Electroacoustics* AU-17, no. 2 (June 1969), pp. 86–92.

[2] Squire, W. D., H. J. Whitehouse, and J. M. Alsup "Linear Signal Processing and Ultrasonic Transversal Filters," *IEEE Transactions on Microwave Theory and Techniques* MTT-17, 1020–1040 (Nov. 1969)

[3] Byram, G. W., J. M. Alsup, J. M. Speiser, and H. J. Whitehouse, "Signal Processing Device Technology," *Proceedings of the NATO Advanced Study Institute on Signal Processing*, held at University of Technology Loughborough, U.K. (21 August – 1 September 1972), pp. IV.10-1 – IV.10.13.

[4] Fifer, W. C., R. LaRosa, and J. F. Crush, *Switchable Acoustic Matched Filter*, Rome Air Development Center Technical Report RADC-TR-72 (April 1972), p. 17.

[5] Wrigley, C. Y., P. J. Hagon, and R. N. Seymour, Abstract of "Programmable SAW Matched Filters for Phase Coded Waveforms," *IEEE Transactions on Sonics and Ultrasonics* SU-20 (January 1973), p. 58.

[6] Buss, D. D., D. R. Collins, W. H. Bailey, and C. R. Reeves, "Transversal Filtering Using Charge Transfer Devices," *IEEE Journal of Solid State Circuits* (April 1973), in press.

[7] Collins, D. R., W. H. Bailey, W. M. Gosney, and D. D. Buss, "Charge-Coupled-Device Analogue Matched Filters," *Electronic Letters* 8, no. 13 (29 June 1972), p. 329.

[8] Buie, J. L. and D. R. Breuer, "A Large-Scale Integrated Correlator," *IEEE Journal of Solid State Circuits* SC-7, no. 5 (October 1972), p. 358.