A NUMBER-THEORETIC SPECTRAL INTERPOLATION ALGORITHM & ITS APPLICATION TO PROBLEM OF CONVENTIONAL BEAMFORMING

K J Jones

GEC-Marconi Naval Systems Ltd., Templecombe, Somerset.

1. INTRODUCTION

In many modern signal and image processing applications, the availability of limited data, whether of one-dimensional (1-D) or multi-dimensional (m-D) form, can make it extremely difficult to resolve the ambiguities inherent in the analysis of spectra produced by the discrete Fourier transform (DFT) algorithm, given in 1-D form by the expression

$$X[k] = \sum_{n=0}^{N-1} x[n]W_N^{nk} \qquad ...(1.1)$$

where

$$W_N = \exp\{-i2\pi / N\} \qquad ...(1.2)$$

the primitive N-th complex root of unity [6,14].

Ambiguities can arise in the analysis of conventional DFT-based spectra, such as via the periodogram [4], through the presence of narrowband signal components, for example, with centre frequencies lying between consecutive DFT outputs, as well as through the often 'rough' appearance of the spectrum resulting from the lack of sufficient spectral resolution from the available data. Two techniques have traditionally been used to deal with such problems : zero-padding of the DFT input [6], and the chirp Z-transform (CZT) algorithm [2].

The *zero-padding technique*, which is discussed in some detail in Section 2, involves interpolating the DFT output by appending a sufficient number of zero-valued samples to the input data sequence, with an increasing number of zero-valued samples producing a DFT spectrum with increasingly fine spectral resolution. The arithmetic complexity of this approach for the case of M samples and an interpolating factor of P, i.e. the appending of (P-1)×M zero-valued samples, is given by $O(PM\times\log_2\{PM\})$ complex operations, via the application of a fast Fourier transform (FFT) algorithm, although a number of *pruning* techniques, using an FFT algorithm which disregards redundant data paths in the flowgraph, have managed to reduce this figure to $O(PM\times\log_2\{M\})$, this being generally achieved at the expense of a loss of regularity of the algorithm structure.

With the CZT algorithm, which is a filtering process, the spectral output is produced by means of M complex pre-multiplications, PM complex post-multiplications, and in between, the application of one complex finite impulse response (FIR) filter, the impulse response of which is that of a chirp filter. The filtering operation can be carried out as a circular convolution of length PM, assuming as before an interpolating factor P, which can be efficiently carried out by means of an FFT algorithm to yield an arithmetic complexity figure of $O(PM\times\log_2\{PM\}$ complex operations. In certain applications its computational efficiency can be considerably improved, but it requires increased memory and remains a far more complex algorithm to implement than the conceptually simple FFT pruning approach.

A new approach to the spectral interpolation problem is discussed in this paper which carries out the computation of a zero-padded DFT via the prime factor algorithm (PFA) [2,14,17]. It is referred to as the *Sino spectral interpolation algorithm*, because of its reliance on the *Chinese remainder theorem* (CRT) [10] for the data reordering. The computation is carried out by means of a modified form of the *row-column method*, as discussed by the author in a previous paper [8], whereby the row-DFTs are processed coefficient-by-coefficient, rather than DFT-by-DFT, with each row-DFT possessing at most one non-zero input sample. The idea generalises very simply to the case of m-D interpolation, although this is not pursued in this paper.

The row-DFT processing decomposes into a number of simple *dual-node* processes, the output of which feeds directly into the column-DFT process, which comprises a set of radix-2 FFTs. The processing thus decomposes into a number of independent *Sino* *computing engines*, where each computing engine comprises one dual-node processor and two scrambled-output, radix-2 FFT routines. Unlike most other pruning techniques, the solution achieves a reduced arithmetic complexity figure, in this instance $O(PM \times \log_2\{M\})$, whilst at the same time achieving an elegant, regular computational structure, leading to a highly-efficient, parallel implementation. A performance analysis of the Sino algorithm is carried out, dealing with both the arithmetic complexity and communication complexity requirements, which emphasises the attraction of the Sino approach, with its inherent parallelism, when realised with very-large-scale integration (VLSI) technology.

Finally, it is seen how the Sino spectral interpolation algorithm can be applied to the problem of conventional beamforming [13], which lies at the heart of many high-performance signal processing systems. It is of particular relevance in the computationally demanding sonar signal processing field [15], where for the case of straight-line towing, a long towed array of hydrophones can be decomposed into a number of approximately-linear sub-apertures, each of which can be used to produce far-field frequency-domain beam patterns via the computationally-efficient Sino algorithm, prior to their coherent or incoherent combination.

2. EFFECT OF ZERO-PADDING ON DFT SPECTRUM

In this section it is seen how the DFT may be modified to produce interpolated transform values between the original N transform values by means of a process already referred to as zero-padding. Suppose that N zero-valued samples, x[N],x[N+1],...,x[2N-1], are appended to the available data samples x[0],x[1],...,x[N-1], i.e. an interpolating factor of two is to be used. Then the DFT of the 2N-point augmented data sequence is given by

$$X[k] = \sum_{n=0}^{2N-1} x[n]W_{2N}^{nk} \qquad ...(2.1)$$

$$= \sum_{n=0}^{N-1} x[n]W_{2N}^{nk} \qquad ...(2.2)$$

where the range of summation has now been adjusted to reflect the fact that the last N samples are zero-valued.

Putting k = 2m in equation 2.2 gives

$$X[m] = \sum_{n=0}^{N-1} x[n]W_{N}^{nm} \qquad ...(2.3)$$

for m=0,1,...,N-1, which represents the even values of X[k], so that the 2N-point DFT of the augmented data sequence reduces to that of the N-point DFT at the even-valued indices. The odd values of the index k represent the interpolated DFT values between the original N-point DFT values. This result generalises in an obvious fashion to interpolation by a factor P, where $P \geq 2$, simply by augmenting the input data sequence with a sufficient number of zero-valued samples, prior to taking the DFT.

There is a common misconception that the technique of zero-padding improves the inherent resolution in the spectrum because it increases the length of the sampling interval. This is not the case, however, as zero-padding simply provides an interpolated DFT spectrum with a smoother appearance, thereby facilitating greater accuracy in estimating the true frequency and amplitude of the spectral peaks and valleys.

It should be noted that the zero-padding technique can also be applied to the dual problem of interpolating the input data sequence, rather than that of its DFT spectrum. This is achieved by first placing zero-valued samples in the centre of the DFT spectrum, as required for preservation of the symmetry of the spectral waveform, prior to taking the inverse DFT of the augmented spectrum. The resulting data sequence will be a band-limited, interpolated version of the original data sequence, with the even-valued terms corresponding to the original data sequence, and the odd-valued terms representing the interpolated samples.

Finally, for computational efficiency, an FFT routine can be used for performing the spectral interpolation. The pruning of the associated FFT flowgraph is generally referred to in the literature as *input pruning*; for the dual problem of band-limited interpolation of the data sequence, the pruning of the associated inverse FFT flowgraph is generally referred to as *output pruning*.

A NUMBER-THEORETIC SPECTRAL INTERPOLATION ALGORITHM

The algorithm discussed in the following sections of this paper is an input pruned version of the PFA, although it does not actually require the input data to be physically padded out with zero-valued samples.

3. 2-D FORMULATION OF DFT VIA CHINESE REMAINDER THEOREM

The ubiquitous Chinese remainder theorem, so called because it was known to the ancient Chinese, plays an important role in many modern signal processing algorithms, and in particular the DFT. The theorem was first stated and proved in its general form by Chin Chiu Shao [10], in 1247, and briefly stated, it provides a unique solution to a set of simultaneous linear congruences [16]. It is thus able to generate 1-D to m-D and m-D to 1-D index mappings for the DFT, which are both unique and cyclic in every dimension.

Thus, application of the Chinese remainder theorem to DFT indexing ensures :

(1) the original DFT can be recovered in its correct form, and

(2) the correct periodicities are obtained for each of the small DFTs that constitute the m-D form,

and given the importance of the m-D formulation of the DFT for the results to follow, it is worth seeing at this point how, via such index mappings, the m-D formulation can be achieved.

Suppose firstly that the length N of a 1-D DFT can be written as

$$N = N_1 \times N_2 \qquad ...(3.1)$$

where N_1 and N_2 are relatively prime factors [16]. Then by applying the relatively prime modulo (RPM) and CRT index mappings [3,14]

$$n \equiv [N_2 n_1 + N_1 n_2] \bmod N \quad \& \quad k \equiv [N_2 t_2 k_1 + N_1 t_1 k_2] \bmod N \qquad ...(3.2)$$

to the input data and output data, respectively, for $n_1 = 0,1,...,N_1-1$ and $n_2 = 0,1,...,N_2-1$, where t_1 and t_2 are given by

$$N_2 t_2 \equiv 1 \bmod N_1 \quad \& \quad N_1 t_1 \equiv 1 \bmod N_2 \qquad ...(3.3)$$

the DFT can be written in a 2-D form as

$$X[k_2,k_2] = \sum_{n_1=0}^{N_1-1} \left[\sum_{n_2=0}^{N_2-1} x[n_1,n_2] W_{N_2}^{n_2 k_2} \right] W_{N_1}^{n_1 k_1} \qquad ...(3.4)$$

with W_{N_1} and W_{N_2} the primitive N_1-th and N_2-th complex roots of unity respectively.

Equation 3.4 thus reduces the N-point DFT, for the two-factor case of interest, to a true 2-D DFT, without twiddle factors [17], consisting of just two partial-DFT processes. Generalisation to the m-D case is a straightforward task, with the resulting expression being more commonly known as the PFA, as referred to in Section 1 above. The same result can be achieved by reversing the order of the RPM and CRT index mappings or even by applying the CRT index mapping to both the input and the output data, although it is not necessary to do so.

4. PARALLELISATION OF 2-D DFT VIA MODIFIED ROW-COLUMN METHOD

Returning to the 2-D DFT formulation of equation 3.4, it is now seen how this can be computed in a highly-parallel fashion to yield high-throughput solutions to the original DFT algorithm, and hence to the Sino spectral interpolation algorithm.

Firstly, by putting

$$y[n_1,k_2] = \sum_{n_2=0}^{N_2-1} x[n_1,n_2] W_{N_2}^{n_2 k_2} \qquad ...(4.1)$$

equation 3.4 can be written as

$$X[k_1,k_2] = \sum_{n_1=0}^{N_1-1} y[n_1,k_2] W_{N_1}^{n_1 k_1} \qquad ...(4.2)$$

This shows that the first partial-DFT, as given by equation 4.1, can be computed by means of N_1 independent N_2-point DFTs, referred to hereafter as the row-DFT process. Similarly, the second partial-DFT, as given by equation 4.2, can be computed by means of N_2 independent N_1-point DFTs, referred to hereafter as the column-DFT process. This decomposition of the DFT into row-DFT and column-DFT processes is known as the row-column method [17], with the 2-factor decomposition being as shown in Figure 4.1.

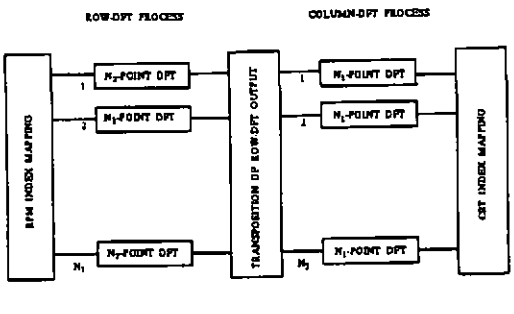

FIGURE 4.1 - ROW-COLUMN IMPLEMENTATION OF PRIME FACTOR DFT ALGORITHM FOR 2-FACTOR CASE

Given the independence of the small DFTs in the row-DFT and column-DFT processes, high-throughput solutions can be simply obtained for the 2-factor decomposition via parallel computation of the N_2-point DFTs in the row-DFT process, followed by parallel computation of the N_1-point DFTs in the column-DFT process. However, before the column-DFT process can be carried out, the matrix containing the row-DFT output data must first be transposed. An alternative approach [8] to implementing the row-column method is now discussed, which is particularly appropriate to the spectral interpolation problem, eliminating the need for matrix transposition of the row-DFT output.

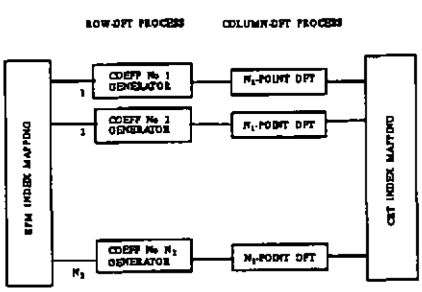

FIGURE 4.2 - MODIFIED ROW-COLUMN IMPLEMENTATION OF PRIME FACTOR DFT ALGORITHM FOR 2-FACTOR CASE

From the expression of equation 3.4, it is seen that each N_1-point column-DFT is carried out upon a set of N_1 outputs from the row-DFTs, where each N_1-point output set corresponds to the same N_2-point DFT coefficient. Therefore, the first column-DFT can be processed as soon as the individual N_2-point zero-frequency components are output from the row-DFT processor. The remaining N_1-point column-DFTs can be similarly processed as soon as the individual components of the corresponding DFT coefficient set are computed.

Thus, by processing the row-DFTs coefficient-by-coefficient, rather than DFT-by-DFT, as is conventionally done, the transposition of the row-DFT matrix output can be eliminated, as seen in Figure 4.2, with the output from each row-DFT coefficient generator feeding directly into a corresponding column-DFT. This idea is now applied to the spectral interpolation problem where particular simplifications resulting from the choice and ordering of the index mappings lead to an attractive, computationally-efficient solution.

A NUMBER-THEORETIC SPECTRAL INTERPOLATION ALGORITHM

5. SINO SPECTRAL INTERPOLATION ALGORITHM

The algorithm described in this and the following sections is based on the direct DFT interpolation technique, discussed in Section 2, where the number of input samples to the interpolator is denoted by M and the interpolation factor by P. The number of spectral outputs, N, produced by the interpolator, referred to hereafter as the Sino algorithm, is given by

$$N = P \times K \qquad ...(5.1)$$

where P is taken to be an odd integer and K is an integral power-of-two such that $M \leq K$, i.e. P and K are relatively prime.

By choosing the interpolation parameters in this way, it is possible to carry out the N-point DFT required by the interpolation algorithm as a row-DFT process, comprising the computation of K independent P-point DFTs, followed by a column-DFT process, comprising the computation of P independent K-point DFTs. If, in addition, the input index mapping is taken as the RPM mapping and the row-DFTs are processed coefficient-by-coefficient, rather than DFT-by-DFT, as discussed in Section 4 above, then the resulting algorithm can be represented by the flowgraph of Figure 5.1, where each P-point row-DFT possesses at most one non-zero input sample, i.e. M row-DFTs possess one non-zero input sample and K-M row-DFTs possess no non-zero input samples.

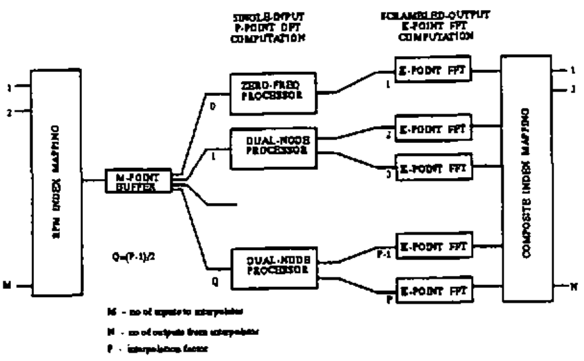

FIGURE 5.1 - FLOWGRAPH FOR SINO SPECTRAL INTERPOLATION ALGORITHM

Thus, the row-DFT process can be decomposed into a single zero-frequency process and ½(P-1) *dual-node* processes, as carried out by the dual-node processor of Figure 5.2, where each dual-node processor computes two P-point DFT coefficients from the same non-zero input sample via the relationship

$$W_P^p = (W_P^{P-p})^* \qquad ...(5.2)$$

for p = 0,1,...,P-1, where W_P is the primitive P-th complex root of unity and where "*" stands for complex conjugation.

The output from the zero-frequency and dual-node processes can be fed directly into the K-point DFTs of the column-DFT process, which can, if required, be carried out by any conventional radix-2 FFT algorithm [2,6,17]. A further simplfication can be achieved, however, by selecting a K-point radix-2 FFT algorithm with naturally ordered input and bit-reversed [6] output, i.e. a decimation-in-frequency (DIF) algorithm [6], as the bit-reversal index mapping can be removed from the FFT algorithm and combined with that of the CRT index mapping, as required by the PFA for reordering of the output data, to yield a composite index mapping for the reordering of the interpolated spectral output.

Thus, by regarding the zero-frequency process as a special case of the dual-node process, the N-point DFT reduces to a very simple, elegant computational structure, comprising ½(P+1) identical, independent processes, with each process being carried out by the Sino computing engine of Figure 5.3, which comprises one dual-node processor and two scrambled output, K-point FFT modules. It should be noted that for the case where M is sufficiently small, i.e. M < 32 say, the value of K could be chosen such that the K-point DFTs of the column-DFT stage could be carried out by means of a theoretically optimal (in the sense of reduced arithmetic operations) small-DFT module, based upon Winograd's complexity theory results [18], thus yielding a reduced-complexity solution. For $M \geq 32$, however, a computationally-efficient implementation of a radix-2 FFT algorithm, such as that based upon the split-radix algorithm [5], would prove a better approach, enabling the regularity of the FFT structure to be fully exploited.

A NUMBER-THEORETIC SPECTRAL INTERPOLATION ALGORITHM

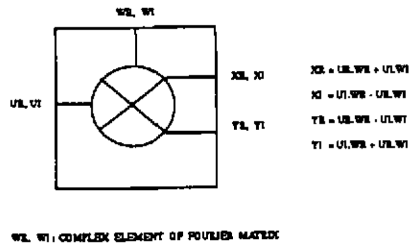

FIGURE 5.2 - DUAL-NODE PROCESSING UNIT

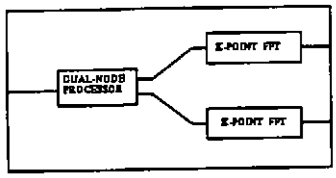

FIGURE 5.3 - SINO COMPUTING ENGINE FOR 1-D INTERPOLATION

Finally, the Sino algorithm provides a very flexible approach to the spectral interpolation problem, as the same computational building block, namely the Sino computing engine, can be used to construct spectral interpolators of arbitrary precision, with only the input and output index mappings needing to be recomputed, together with the Fourier matrix elements required by each dual-node processor. Also, the Sino algorithm does not require the input data to be physically padded out with zero-valued samples, so that the memory requirement, for both input data and input index mapping, is kept to a minimum, whilst the memory requirement for the output data buffer is proportional to the number of interpolated spectral outputs to be produced, i.e. upon the interpolation accuracy required.

6. PERFORMANCE ANALYSIS OF SINO SPECTRAL INTERPOLATION ALGORITHM

To properly assess the performance of the Sino spectral interpolation algorithm, it is necessary to consider the arithmetic complexity, communication complexity and structure of the algorithm for the particular implementation concerned, whether it be on a von Neumann uni-processor machine [1,11], or a high-throughput, parallel implementation using an array of processing elements to pipeline the computation, as for example with a systolic array [12].

The performance, in terms of throughput, of a particular implementation can be expressed in terms of its time-complexity [1], which comprises components corresponding to :

(1) its arithmetic complexity, i.e. total number of arithmetic operations, and
(2) its communication complexity, i.e. total number and length of data routing operations,

with each being expressed as a number of time units, where one *arithmetic time unit* corresponds to the time taken to carry out a basic arithmetic operation, and one *routing time unit* corresponds to the time taken to move a sample of data one distance unit, where one distance unit is the distance between consecutive processing elements.

6.1. Arithmetic Complexity of Sino Algorithm

As far as the arithmetic complexity is concerned, which would be the prime performance metric for a uni-processor implementation, an appropriate measure is given by the total number of complex multiplication/accumulation operations required, as this operation forms the major computational load of any technique based upon the FFT algorithm. One arithmetic time unit therefore corresponds to the time taken to carry out one complex multiplication/accumulation operation.

For a data sequence of length M, with a corresponding PFA factor K, and an interpolating factor P, the arithmetic complexity, S_A, of the Sino algorithm can be expressed as

$$S_A = \tfrac{1}{2}(P-1)M + \tfrac{1}{2}PK\times\log_2\{K\}$$
$$\sim \tfrac{1}{2}PK\times\log_2\{K\} \qquad ...(6.1)$$

arithmetic time units.

For the direct DFT solution, using an FFT algorithm but without pruning, the arithmetic complexity, D_A, involves

A NUMBER-THEORETIC SPECTRAL INTERPOLATION ALGORITHM

$$D_A = \tfrac{1}{2}PK \times \log_2\{PK\}$$
$$= \tfrac{1}{2}PK \times \log_2\{P\} + \tfrac{1}{2}PK \times \log_2\{K\} \qquad \text{...(6.2)}$$

arithmetic time units, which implies that the Sino algorithm involves

$$\sim \tfrac{1}{2}PK \times \log_2\{P\} \qquad \text{...(6.3)}$$

fewer time units than that of the direct DFT solution. Using an input pruned FFT algorithm, however, the arithmetic complexity, P_A, reduces to just

$$P_A \sim \tfrac{1}{2}PK \times \log_2\{K\} \qquad \text{...(6.4)}$$

arithmetic time units, which is of the same order as achieved by the Sino algorithm.

Similarly, for an efficient CZT solution, i.e. one exploiting a pre-computed FFT for use in the circular convolution operation, the arithmetic compclexity, C_A, can be written as

$$C_A = 2PK + M + PK \times \log_2\{PK\}$$
$$\sim 2PK + PK \times \log_2\{P\} + PK \times \log_2\{K\} \qquad \text{...(6.5)}$$

arithmetic time units, which implies that the Sino algorithm involves

$$\sim PK \times (2 + \log_2\{P\sqrt{K}\}) \qquad \text{...(6.6)}$$

fewer time units than that of the CZT solution.

Thus, from these figures, it is evident that in terms of the arithmetic complexity alone, the Sino algorithm looks an attractive candidate for performing spectral interpolation, when compared to both the direct DFT and the CZT solutions, with its relative computational efficiency improving as the interpolation factor P is increased. Its arithmetic complexity is similar to that achieved by other FFT pruning techniques, but it has the additional attraction of possessing an elegant, regular structure, making it an ideal candidate for VLSI implementation, as now discussed.

6.2. Communication Complexity and Structure of Sino Algorithm

The Sino spectral interpolation algorithm decomposes into an input index mapping, a number of independent and identical processing modules, and a composite output index mapping. Such a decomposition lends itself naturally to a parallel implementation, where the parallelism can be achieved in one of two ways :

(a) assigning up to Q+1 computing engines, where $Q = \tfrac{1}{2}(P-1)$, for carrying out the processing, these operating in parallel; and

(b) building parallelism into the design of the computing engine, with the two column-DFTs, for example, being carried out in parallel,

with maximum throughput being achieved by assigning Q+1 parallel implementations of the Sino computing engine to the processing, although in practise a trade-off would have to be made between hardware size, cost and performance.

Given the independence of the Sino processing modules, the communication complexity of the Sino implementation is limited to that of an individual computing engine, which is dependent upon how each of the K-point, scrambled-output FFT modules is to be implemented. For a parallel, radix-2 solution, comprising ½K processing elements, for example, where the processing element is a complex butterfly operator [6], the maximum distance that data samples must travel to reach their required processing element is ¼K distance units, whereas for the conventional input pruned FFT solution, comprising ½PK processing elements, the maximum distance is given by ¼PK distance units.

Since the time-complexity associated with the routing of data over d distance units is $\sqrt{d}$ routing time units, the time-complexity of the routing operations for the input pruned FFT algorithm, P_C, is given by

$$P_C \sim O(\sqrt{PK}) \qquad \text{...(6.7)}$$

routing time units, whilst that for the Sino algorithm, S_C, is given by

$$S_C \sim O(\sqrt{K}) \qquad \text{...(6.8)}$$

routing time units, with the result that the communication complexity of a parallel implementation of the input pruned FFT algorithm, i.e. one involving ½PK complex butterfly units, is approximately $\sqrt{P}$ times greater than that of a parallel implementation of the Sino algorithm, i.e. one involving ½K complex butterfly units for each K-point FFT.

For the case where M is sufficiently small, i.e. $M \leq 32$, the value of K can be set to M and each K-point DFT carried out in a fully-pipelined, systolic fashion by means of a linear array of K processing elements [8], where each processing element carries out the same operation of complex multiplication/accumulation. The row-DFT output from the dual-node processor can be pipelined into the two length-K linear arrays at a constant rate, to give a time-complexity for the Sino computing engine, and hence for the Sino spectral interpolation algorithm, of O(K), with each computing engine producing all 2K outputs in just K time units.

Thus, the Sino spectral interpolation algorithm possesses a number of properties, not found in more conventional solutions, which make it an ideal candidate for VLSI implementation. The regularity of its structure, i.e. the repetitive use of a single building block - the Sino computing engine, makes it very easy to design spectral interpolators of arbitrary precision. This, combined with the relatively simple communication requirements, which are dependent purely upon the implementational complexity of the computing engine itself, lead to a cost-effective solution when implemented with VLSI technology.

7. APPLICATION TO CONVENTIONAL BEAMFORMING WITH ARRAY OF HYDROPHONES

Modern towed array sonar systems are being built with an increasing number of channels in order to achieve the desired acoustic aperture length. Such systems, which can comprise up to a thousand channels, possess tremendous throughput requirements which have to be met by the available processing equipment. Thus, there is an ever-present need for improved processing algorithms and processing strategies [15], not only to fully exploit the size of the aperture, but also to overcome the limitations imposed by the excessive computational demands of many conventional solutions.

A sub-aperture beamforming strategy, for example, can offer a number of advantages over a single aperture scheme, both in terms of processing gain and computational efficiency, e.g. ease of partitioning. Where the spatial coherence of the incoming signal is limited by the environment, a number of sub-apertures may each be coherently processed and then appropriately combined to yield the full beamformer output. If the prevailing propagation conditions permit the use of a large aperture, then the sub-aperture beamformer outputs may be coherently combined to yield the optimum gain from the array. If conditions do not permit this, however, they may be incoherently combined to yield the best possible result under the prevailing conditions.

For towing conditions where the sub-apertures may be assumed to be approximately linear, the beamforming may be efficiently carried out via frequency-domain techniques, as they offer a number of computational advantages over the more conventional time-domain approach, in terms of both reduced sampling rates and, in certain applications, increased computational efficiency [13]. This is particularly so when the number of hydrophones and the number of beams to be generated is large, or when only a fairly limited region of the frequency-bearing space is of interest, as for example with a trials system where one has control over the target parameters of interest. Adaptive processing algorithms and acoustic array shape estimation algorithms, which are currently receiving some attention [7], might also dictate a preference for highly-directional frequency-domain beamformer output.

By applying the Sino spectral interpolation algorithm to the sub-aperture hydrophone data at a given temporal frequency (i.e. following transformation of the hydrophone data from the time domain to the frequency domain via a real-to-complex FFT algorithm), with an appropriate choice of interpolating factor P, the far-field beamformer outputs can be generated sufficiently close together in frequency-bearing space as to ensure that one of the interpolated outputs corresponds, to a specified accuracy, to any desired beam steering angle. Wideband time-series data, if required, can be reconstructed from the frequency-domain beamformer output by passing the corresponding beam set through an inverse, complex-to-real FFT algorithm, as shown in the towed array trials system of Figure 7.1. This must be done, however, in accordance with either the overlap-add or overlap-save technique [17], as the ends of the data segments need to be correctly treated to account for edge effects due to cyclic wrap-around of the input time-series resulting from channel phasing [15].

For each sub-aperture, an index mapping can be defined which maps the required subset of Sino outputs from that sub-aperture to the set of beam steering angles. This mapping is periodically updated to take account of the changing orientation of the sub-aperture with respect to the coordinate axes in which the array position is defined, or to cater for changes to the speed of sound in the water. The mapping may be combined with the Sino output index mapping, however, so that changes may be simply catered for by modifying the composite output index mapping, the other components of the beamformer remaining unchanged.

A NUMBER-THEORETIC SPECTRAL INTERPOLATION ALGORITHM

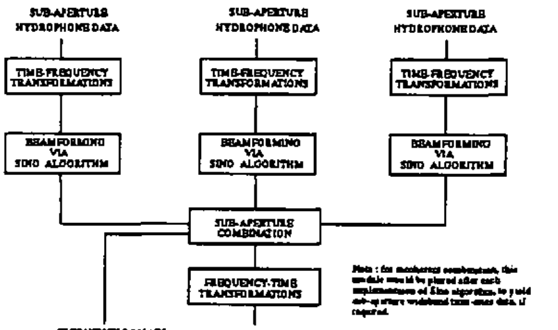

FIGURE 7.1 - SUB-APERTURE BEAMFORMING STRATEGY VIA SINO SPECTRAL INTERPOLATION ALGORITHM

For applications where the steered beams are equally spaced in the sine of the steering angle, the CZT algorithm could equally well be used, as far as arithmetic complexity requirements alone are concerned, as its arithmetic complexity, for the generation of N beams from N hydrophones, reduces to just

$$C_A = 2N(3 + \log_2\{N\}) \qquad ...(7.1)$$

arithmetic time units, with the length of the associated circular convolution being only of order 2N. This compares well with the Sino complexity figure of

$$S_A = ½(P-1)N + ½PN\times\log_2\{N\} \qquad ...(7.2)$$

arithmetic time units. In fact, for parameter values of N = 256 and P = 5 (i.e. corresponding to hydrophone data from a 128λ array, with a spectral interpolation factor of 5), the arithmetic complexity figures of the CZT and Sino algorithms are identical. However, it should be stated that the CZT only achieves this reduced-complexity figure at the expense of an increased memory requirement, for the storage of pre-computed coefficients, and a considerably more complex algorithm structure.

Finally, another fast solution to the conventional beamforming problem, which is little known in the sonar signal processing community, is via the Arbitrary Frequency Fourier Transform (AFFT) algorithm [9], as used by the author some years ago for the design of a compact, frequency-domain implementation of the BAe Active Towed Array Sonar (ATAS) system. This solution, which involves an approximation to the Z-transform over an arc of the unit circle, requires less arithmetic operations than the CZT approach, but like the CZT, is somewhat complex, relying heavily on the generation and storage of pre-computed coefficients for each temporal frequency of interest.

8. Summary and Conclusions

A novel signal processing algorithm has been described in this paper for the efficient computation of an interpolated DFT spectrum, and hence of a spatial spectrum, as required by a conventional beamformer. The algorithm, which is based on the conventional zero-padding approach, involves a modified row-column implementation of the prime factor DFT algorithm, and reduces to a number of independent processing modules which can be implemented in parallel to yield computationally-efficient, high-throughput solutions to the spectral interpolation problem, and to its dual, the problem of band-limited interpolation.

The computation is carried out by means of a modified form of the row-column method, whereby the row-DFTs are processed coefficient-by-coefficient, rather than DFT-by-DFT, as is conventionally done, with each row-DFT possessing at most one non-zero input sample. This enables the row-DFT computation to be decomposed into a number of simple dual-node processes, where each such process involves the computation of two P-point DFT coefficients, where P is the interpolation factor, from the same non-zero input sample.

The outputs from the dual-node processes feed directly into the column-DFT process, comprising a set of DFT modules, which, given sufficient input data, are best implemented as radix-2, decimation-in-frequency FFTs. The processing thus decomposes into a number of independent Sino computing engines, where each computing engine comprises one dual-node processor and two radix-2 FFT routines, the output of which can be left in scrambled form, as the bit-reversal mapping can be simply combined with the CRT index mapping, as required by the PFA for reordering of the Sino output data.

The Sino algorithm provides a very flexible approach to the spectral interpolation problem, as the same computational building block, namely the Sino computing engine, can be used to construct spectral interpolators of arbitrary precision, with only the input and output index mappings needing to be recomputed, together with the Fourier matrix elements required by each dual-node processor. Also, the Sino algorithm does not require the input data to be physically padded out with zero-valued samples, so that the memory requirement, for both input data and input index mapping, is kept to a minimum.

Finally, the Sino algorithm achieves the same reduced arithmetic complexity figures as other computationally-efficient pruning techniques based upon the FFT algorithm. However, unlike other approaches, it does so whilst maintaining an elegant, regular computational structure, with minimal communication requirements, as the communication is local within each of the Sino computing engines, which operate independently of each other to yield highly-efficient, parallel implementations, particularly with VLSI technology in mind.

Acknowledgement - The author wishes to acknowledge the support of Colin Richardson and his Towed Array Sonar Systems group at DRA Winfrith Technology Centre, Dorset, for whom this work was carried out.

9. References

[1] S.G. AKL, "The Design and Analysis of Parallel Algorithms", Prentice-Hall, 1989.

[2] R.E. BLAHUT, "Fast Algorithms for Digital Signal Processing", Addison-Wesley, 1985.

[3] C.S. BURRUS, "Index Mappings for Multi-Dimensional Formulation of the DFT and Convolution", IEEE Trans. on Acoustics, Speech and Signal Processing, 1977, ASSP-25, (3), pp. 239-242.

[4] D.G. CHILDERS (Ed.), "Modern Spectrum Analysis", IEEE Press, 1978.

[5] P. DUHAMEL, "Implementation of Split-Radix FFT Algorithms for Complex, Real & Real-Symmetric Data", IEEE Trans. on Acoustics, Speech and Signal Processing, 1986, ASSP-34, (2), pp. 285-295.

[6] D.F. ELLIOTT & K.R. RAO, "Fast Transforms: Algorithms, Analyses, Applications", Academic Press, 1982.

[7] B.G. FERGUSON, "Remedying the Effects of Array Shape Distortion on the Spatial Filtering of Acoustic Data from a Line Array of Hydrophones", IEEE J. Oc. Eng., 18, (4), 1993, pp. 565-571.

[8] K.J. JONES, "High-Throughput, Reduced-Hardware Systolic Solution to Prime Factor Discrete Fourier Transform Algorithm", IEE Proc. on Computers and Digital Techniques, Pt. E, 1990, Vol-137, (3), pp. 191-197.

[9] K.J. JONES, "Fast Conventional Beamforming via Arbitrary Frequency Fourier Transform", unpublished technical note.

[10] D.E. KNUTH, "The Art of Computer Programming", Vol. 2, Addison-Wesley, 1981.

[11] L. KRONSJO, "Computational Complexity of Sequential and Parallel Algorithms", John Wiley & Sons, 1985.

[12] H.T. KUNG & C.E. LEISERSON, "Systolic Arrays (for VLSI)", Sparse Matrix Symposium, SIAM,1978, pp. 256-282.

[13] B. MARANDA, "Efficient Digital Beamforming in the Frequency Domain", J. Acoust. Soc. Am., 86, (5), 1989, pp. 1813-1819.

[14] J.H. McCLELLAN & C.M. RADER, "Number Theory in Digital Signal Processing", Prentice-Hall, 1979.

[15] R.O. NIELSEN, "Sonar Signal Processing", Artech-House, 1991.

[16] I. NIVAN & H.S. ZUCKERMAN, "An introduction to the Theory of Numbers", John Wiley & Sons, 1980.

[17] H.S. NUSSBAUMER, "Fast Fourier Transform and Convolution Algorithms", Springer-Verlag, 1981.

[18] S. WINOGRAD, "Arithmetic Complexity of Computations", SIAM Monograph, Vol-33, 1980.