Proceedings of The Institute of Acoustics

A SYSTOLIC ARRAY ADAPTIVE ANTENNA PROCESSOR

P.J. Hargrave

S.T.L. London Road, Harlow, Essex CM17 9NA

Abstract

A new technique for adaptive antenna beamforming is described which provides very rapid convergence. The technique avoids the explicit computation of the covariance matrix estimate thereby resulting in reduced sensitivity to limited precision arithmetic. The proposed algorithm may be implemented in an efficient architecture using a triangular systolic array.

Introduction

Adaptive beamforming provides a powerful means of enhancing the performance of a broad range of radar and communication systems in hostile electromagnetic environments. The technique operates by combining the signals from an array of antenna elements in an adaptive weighting network, the coefficients of which are automatically adjusted to null jamming waveforms and optimize the reception of a desired signal.

Optimal filter theory states that the weighting vector with which we should combine the element outputs is that which is a solution to the Wiener-Hopf equation $\underline{M}\,\underline{W} = \underline{C}$. Here $\underline{M}$ is a covariance matrix with components equal to the correlations between the outputs from all possible pairs of elements in the array, and $\underline{W}$ is the weighting vector we are seeking. For those cases where we are seeking to maximise the desired signal to noise ratio, $\underline{C}$ is a vector characterising the arrival direction of this signal. Often we have insufficient a priori information to achieve such a solution. Instead, we then remove the desired signal from the adaptation process, and seek the weighting vector which minimises the output power from the array, which then results solely from the undesired interference and jamming waveforms. We then need to apply some form of weight constraint to prevent the trivial solution of $\underline{W} = \underline{0}$ being obtained by our algorithm. This usually takes the form of a fixed weight being applied to one of the elements, and $\underline{C}$ is then a column vector which effectively defines this element.

Direct Solutions by Sample Matrix Inversion

At first sight it would appear that the best way to proceed is to average products of element outputs from successive measurement snap-shots to form a Maximum Likelihood estimate of $\underline{M}$. We may update our estimate of $\underline{M}$ each time a new snap-shot is available and then compute the resulting best estimate of $\underline{W}$ as $\underline{M}^{-1}\,\underline{C}$ (Figure 1). This is termed the Sample Matrix Inversion approach [1] and results in the optimum path of adaptation, requiring only minimal data to provide a null pattern which cancels jamming signals over a wide dynamic range of received power levels.

It is clear that we do not actually need to evaluate $\underline{M}^{-1}$. Rather, at each stage we simply require the solution to $\underline{M}\ \underline{W} = \underline{C}$, where $\underline{M}$ is the current Maximum Likelihood estimate of the covariance matrix. The classical way of solving such a linear equation set is to multiply both sides by a series of full rank matrix transformations until it is reduced to the form

$$\underline{R}\ \underline{W} = \hat{\underline{C}} ,$$

where $\underline{R}$ is an upper triangular matrix. We may then evaluate $\underline{W}$ by the process of back substitution. The full rank transformations are simply chosen to drive successive elements of $\underline{M}$ below the main diagonal to zero. One can, for example, use a set of transformations due to Givens, which correspond to rotations in the n dimensional complex space in which we are working. Such an approach has particularly good numerical properties. The resulting algorithm can be implemented using a systolic array as described by Kung and Gentleman[2]. The structure, which is illustrated in Figure 2, uses two sorts of processing nodes which we describe as boundary and internal cells. Although this approach gives the speed advantage of parallel processing for the matrix triangularisation portion of the calculation, we still have to compute an updated estimate of $\underline{M}$ each time a new snap-shot is available from the array output. In the same way that it was noted that we were not interested in $\underline{M}^{-1}$, we observe that we are not actually concerned with $\underline{W}$. It is therefore pertinent to examine whether we can evaluate $\underline{W}$ directly from the data snap-shots without the need to form $\underline{M}$ explicitly.

Data Domain Algorithms

We may write successive snap-shots from the elements with variable weights as the rows of a matrix $\underline{X}$, and those from the element with fixed weight as the components of a column vector $\underline{y}$ (Figure 3). We then seek the weighting vector $\underline{W}$ that minimises the sum of the squares of the magnitude of the elements of the vector $\underline{e}$ defined by

$$\underline{e} = \underline{X}\ \underline{W} - \underline{y}$$

The obvious tendency is to "square both sides", but this leads to a solution in terms of the covariance formulation. Instead, we multiply both sides by a series of the Givens rotations we considered above. A rotation does not alter the length of the vector $\underline{e}$, and hence leaves unaffected the quantity we are trying to minimise. We thereby annihilate successive elements of $\underline{X}$ until the equation is reduced to the form

$$\underline{Q}\ \underline{e} = \begin{bmatrix} \underline{R} \\ \underline{O} \end{bmatrix} \underline{W} - \begin{bmatrix} \underline{b}_1 \\ \underline{b}_2 \end{bmatrix}$$

Here, $\underline{Q}$ is the resultant effect of all the rotations which individually operate on pairs of rows at a time, and $\underline{R}$ is an upper triangular matrix. The $\underline{b}_1$ and $\underline{b}_2$ are no more than a partitioned representation of $\underline{Q}\ \underline{y}$. It is now clear that minimum length of $\underline{Q}\ \underline{e}$, and hence of $\underline{e}$, occurs when $\underline{R}\ \underline{W} = \underline{b}_1$. $\underline{W}$ then follows by back substitution.

Once again, we may employ a systolic processing structure, but this time we may proceed all the way from the data snap-shots from the array to the input to the back substitution processor.[3] The resulting structure is illustrated in Figure 4. Each horizontal array of processing nodes performs a rotational transform with the rotation parameters computed at the circular (or boundary) node and the rotation applied at the square (or internal) nodes.

Input data in vector form is applied to the systolic array in time skewed sequence and then flows through the network of processing nodes on the application of successive clock pulses to the array. Of particular interest is the significance of the output from the lowest internal cell after a full update of the triangular system. Rather surprisingly it represents a scaled version of the desired beamformed output that would be obtained had a weight vector been calculated by back substitution and then applied to the data vector. The scaling factor follows from parameters characterising the individual rotations applied to the data vector. The links between the boundary cells in the structure provide access to this scaling factor.

The triangular systolic array thus acts as a funnel which collects the raw data at the top and delivers the beamformed residual at the final stage.

We have thus reformulated the solution to the adaptive antenna problem in terms of $\underline{X}$, which simplistically is the square root of $\underline{M}$. As well as increasing the speed of computation by exploiting a parallel processing architecture, we have also gained the advantage of reduced susceptibility to instabilities that result from the use of finite precision arithmetic.

Practical Consideration

Wide bandwith military adaptive antenna applications will place stringent demands on the throughput rate of the processing nodes for the systolic array implementation. Furthermore, due to the arithmetic requirements, the systolic array nodes are, in themselves, complex at the system level involving floating point computation and flexible I/O.

In view of the processing node complexity, a building block design approach is considered to be inevitable for future systolic array networks, where, with processing elements integrated as single VLSI devices, the systolic array configuration becomes one of board interconnection and layout. Cascading complex components of this type in a structured manner thus allows extremely complicated signal processing systems to be designed more efficiently and offers the potential for very high computational throughput rates matched to the requirements of advanced radar and communication receivers.

Ref: [1] Reed I.S., Mallett J.D. and Brennan L.E. - "Rapid Convergence Rate in Adaptive Arrays". Institute of Electrical and Electronics Engineers Transactions, Vol. AES-10, No 6, November 1974, pp 853-863.

[2] Kung, H.T. and Gentleman, W.M. - Matrix Triangularisation by Systolic Arrays", Proc. SPIE, 1981, 298, Real-Time Signal Processing IV.

[3] Ward,C.R., Robson, A.J., Hargrave, P.J., and McWhirter, J.G., - "Application of a Systolic Array to Adaptive Beamforming": Institution of Electrical Engineers Proceedings, Vol.131, Pt.F., No 6, October 1984.

Acknowledgement: The author thanks the Directors of Standard Telecommunication Laboratories Ltd. for permission to publish this paper. The work has been carried out with the support of the Procurement Executive, Ministry of Defence and sponsored from the Royal Signals & Radar Establishment.

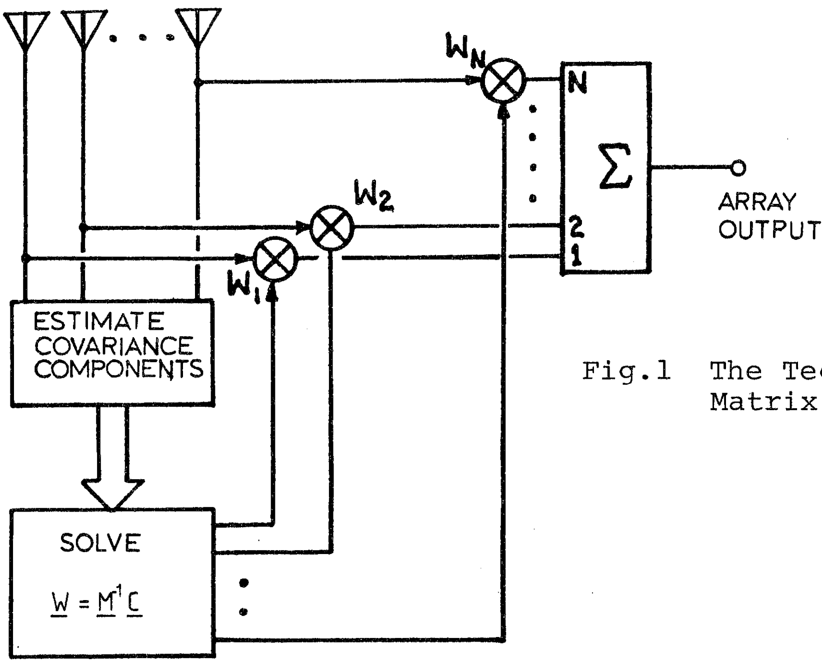

Fig.1 The Technique of Sample Matrix Inversion.

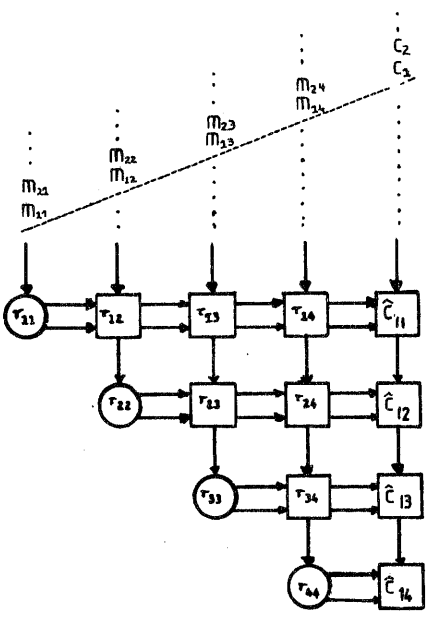

Fig.2 A Systolic Array for Orthogonal Triangularisation.

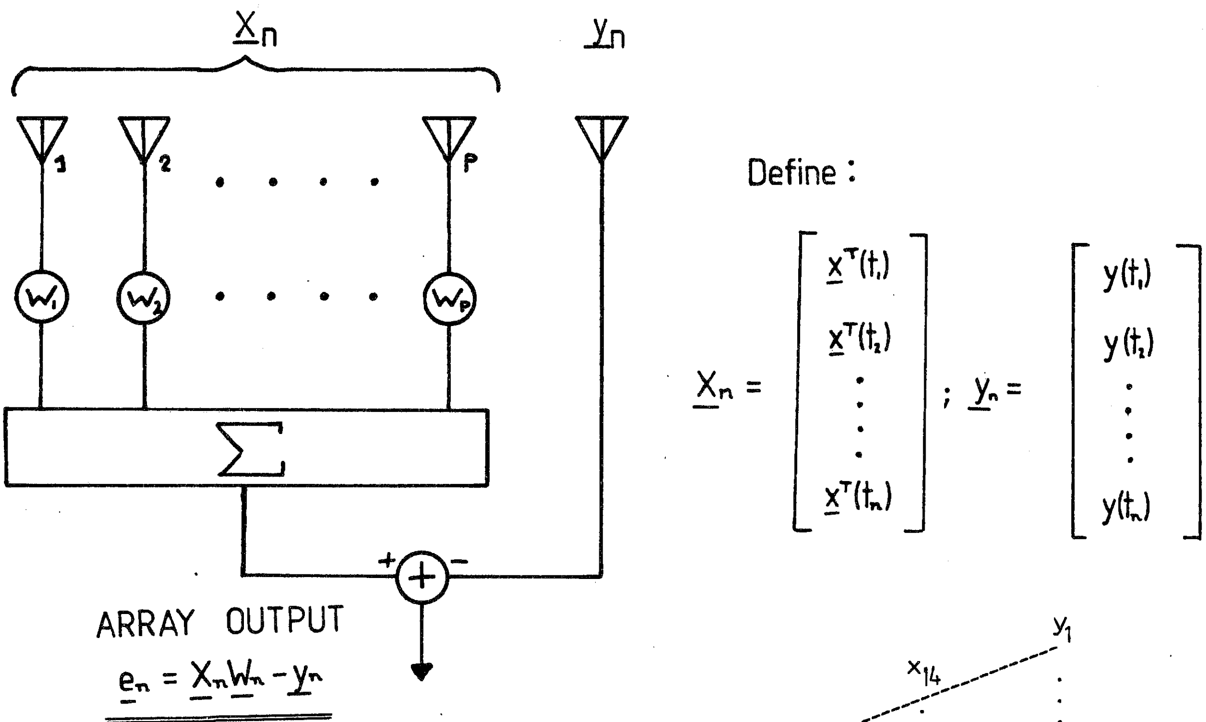

Fig.3 Matrix Notation

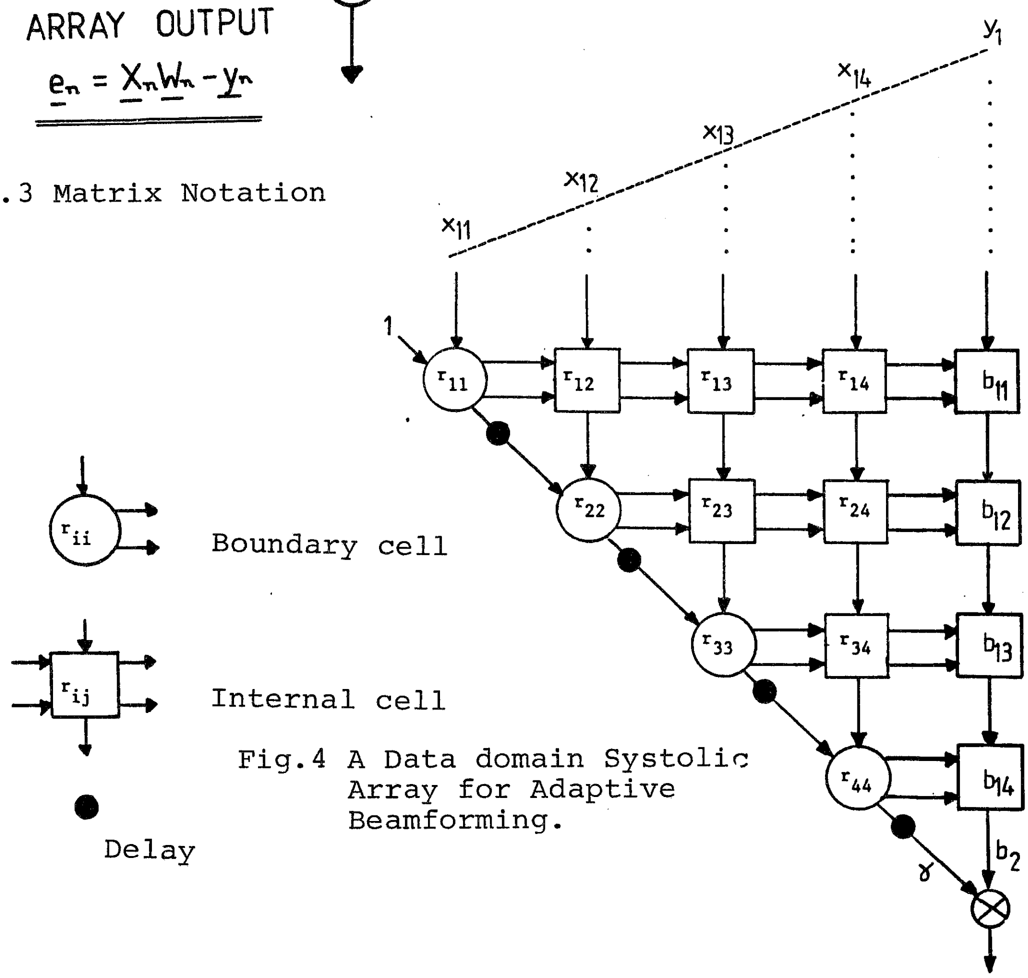

Fig.4 A Data domain Systolic Array for Adaptive Beamforming.