A Novel Systolic Array Processor for MVDR Beamforming

C.F. Tom Tang [1] *and K.J. Ray Liu*
Electrical Engineering Department and Systems Research Center
University of Maryland, College Park, MD 20740 USA

ABSTRACT

A fully-pipelined systolic array for computing the minimum variance distortionless response (MVDR) was first proposed by McWhirter and Shepherd. The fundamental concept is to fit the MVDR beamforming to the non-constrainted recursive least-squares (RLS) minimization. Until now, their systolic array processor is well-recognized as the most efficient design for MVDR beamforming. In this paper, we first propose a different approach for the MVDR beamforming which does not require the non-constrainted RLS minimization, then present a fully parallel and pipelined systolic array for the newly proposed algorithm, and finally describe the similarity and differences between both MVDR beamforming designs.

1 Introduction

McWhirter and Shepherd [1] proposed a single fully-pipelined systolic array processor (SAP) for minimum variance distortionless response (MVDR) beamforming by implementing Schreiber's algorithm without the need of an extra back substitution processor for computing the residual [2]. Up to now, McWhirter and Shepherds' systolic array processor is well-recognized as the most efficient design for MVDR beamforming. Their SAP for MVDR beamforming is designed by using the recursive least squares (RLS) algorithm proposed in McWhirter's earlier paper [3]. It is shown in [1, 4, 5, 6, 7] that most research works for MVDR beamforming have focused on how to fit MVDR to RLS minimization problem. In this paper, instead of fitting MVDR to RLS, we will show that the residual of MVDR beamforming can be obtained directly. The residual of the newly proposed MVDR beamformer is obtained by using the upper part of a unitary matrix, P, while the residual of the McWhirter-Shepherd's MVDR beamformer described in [3] is derived by employing the lower part of a unitary matrix, S.

This paper is organized as follows. In the second section, the new approach for MVDR beamforming is presented. In the third section, a single fully-pipelined MVDR systolic array processor is proposed and the operations for its processor elements are also described. In the fourth section, McWhirter and Shepherds' MVDR bamforming algorithm is briefly summarized and the similarity and differences of two different approaches are then discussed.

[1] with System Sciences Division, Computer Sciences Corporation, El Segundo, California

2 A Novel MVDR algorithm

The $n \times n$ unitary matrix $Q(n)$ used for recursive updating can be factorized into

$$Q(n) = \hat{Q}(n)\overline{Q}(n-1), \tag{1}$$

where $\overline{Q}(n-1)$ is

$$\overline{Q}(n-1) = \begin{bmatrix} P(n-1) & 0 \\ S(n-1) & 0 \\ 0 & 1 \end{bmatrix}, \tag{2}$$

where $P(n-1)$ is a $N \times (n-1)$ matrix, $S(n-1)$ is a $(n-1-N) \times (n-1)$ matrix.
$\hat{Q}(n)$ is

$$\hat{Q}(n) = \begin{bmatrix} A(n) & 0 & \underline{a}(n) \\ 0 & I & 0 \\ \underline{b}^T(n) & 0 & \gamma(n) \end{bmatrix}, \tag{3}$$

where

$$A(n) = \begin{bmatrix} c_1 & 0 & \cdots & 0 \\ -s_1 s_2^* & c_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -s_1 c_2 \cdots c_{n-1} s_n^* & -s_2 c_3 \cdots c_{n-1} s_n^* & \cdots & c_n \end{bmatrix}, \tag{4}$$

0-7803-0532-9/92 $3.00

$$\underline{a}^T(n) = [\ s_1^* \quad c_1 s_2^* \quad c_1 c_2 s_3^* \quad \cdots \quad c_1 \cdots c_{n-1} s_n^*\ ], \tag{5}$$

$$\underline{b}^T(n) = [\ -s_1 c_2 \cdots c_n \quad -s_2 c_3 \cdots c_n \quad \cdots \quad -s_n\ ], \tag{6}$$

and

$$\gamma(n) = c_1 c_2 \cdots c_{n-1} c_n. \tag{7}$$

Accordingly, the $n \times n$ unitary matrix $Q(n)$ can also be partitioned into two submatrices as in (2) given by

$$Q(n) = \begin{bmatrix} P(n) \\ S(n) \end{bmatrix}, \tag{8}$$

where $P(n)$ is a $N \times n$ matrix and $S(n)$ is a $(n - N) \times n$ matrix. From (3) and (2), we obtain

$$P(n) = [\ A(n)P(n-1) \quad \underline{a}(n)\ ]. \tag{9}$$

Applying the QR decomposition to $X(n)$, it follows that

$$Q(n)X(n) = \widehat{Q}(n) \begin{bmatrix} \beta R(n-1) \\ 0 \\ \underline{x}^T(t_n) \end{bmatrix} = \begin{bmatrix} R(n) \\ 0 \\ 0 \end{bmatrix}. \tag{10}$$

According to the relation between the lower triangular matrix $R^H(n)$ and the parameter $\underline{z}^i(n)$ by definition, the same matrix $\widehat{Q}(n)$ can be used to update $R^H(n-1)$ and $\underline{z}^i(n-1)$ at the same time given by

$$\widehat{Q}(n) \begin{bmatrix} \frac{1}{\beta}\underline{z}^i(n-1) \\ \# \\ 0 \end{bmatrix} = \begin{bmatrix} \underline{z}^i(n) \\ \# \\ \# \end{bmatrix}, \tag{11}$$

with # denotes an arbitrary vector or scalar of no interest in mathematical and physical concept. The residual vector for MVDR beamforming is [3]

$$\underline{e}^i_{MVDR}(n) = \frac{r^i}{\|\underline{z}^i(n)\|^2} X(n) R^{-1}(n) \underline{z}^i(n). \tag{12}$$

For convenience, we define a vector $\widehat{\underline{e}}^i_{MVDR}(n)$ from (12) as

$$\widehat{\underline{e}}^i_{MVDR}(n) = X(n) R^{-1}(n) \underline{z}^i(n). \tag{13}$$

By substituting (10), (8), and (9) into (13), the updated residual vector $\widehat{\underline{e}}^i_{MVDR}(n)$ is obtained as

$$\begin{aligned} \widehat{\underline{e}}^i_{MVDR}(n) &= Q^H(n) \begin{bmatrix} R(n) \\ 0 \end{bmatrix} R^{-1}(n) \underline{z}^i(n) \\ &= \begin{bmatrix} P^H(n-1) A^H(n) \underline{z}^i(n) \\ \underline{a}^H(n) \underline{z}^i(n) \end{bmatrix}. \end{aligned} \tag{14}$$

1. Initialize Conditions at $n = 0$ by setting $R(0) = 0 \quad \underline{z}^i(0) = 0$
2. Initialization Procedure for $0 \le n \le N$:
   (a) $Q(N)X(N) = R(N)$ (Mode 1)
   (b) $\underline{z}^i(N) = R^{-H}(N)\underline{c}^i$ (Mode 2)
3. Recursive Procedure for $n > N$ (Mode 1 only):
   (a) $\widehat{Q}(n) \begin{bmatrix} \beta R(n-1) \\ 0 \\ \underline{x}^T(t_n) \end{bmatrix} = \begin{bmatrix} R(n) \\ 0 \\ 0 \end{bmatrix}$
   (b) $\widehat{Q}(n) \begin{bmatrix} \frac{1}{\beta}\underline{z}^i(n-1) \\ \# \\ 0 \end{bmatrix} = \begin{bmatrix} \underline{z}^i(n) \\ \# \\ \# \end{bmatrix}$
   (c) $e^i_{MVDR}(t_n) = \frac{r^i}{\|\underline{z}^i(n)\|^2} \underline{a}^H(n) \underline{z}^i(n)$

Table 1: Summary of Parallel/Pipelined QRD-MVDR Algorithm

Finally, the current updated residual of the MVDR beamforming is then given by

$$e^i_{MVDR}(t_n) = \frac{r^i}{\|\underline{z}^i(n)\|^2} \underline{a}^H(n) \underline{z}^i(n). \tag{15}$$

The newly developed MVDR algorithm is summarized in Table 1. First, the upper triangular matrix and a parameter vector are initialized by setting to be zero, i.e., $R(0) = 0$ and $\underline{z}(0) = \underline{0}$. Second, in the initialization for $0 \le n \le N$, when the observed input data matrix $X(N)$ is available, where N is the number of sensors, the initial upper triangular matrix $R(N)$ is generated by the QR decomposition and the initial parameter vector $\underline{z}^i(N)$ is computed by parallel multiplication instead of the forward substitution. Finally, for the recursive updating during the time period $n > N$, the upper triangular matrix $R(n-1)$ and the parameter matrix $\underline{z}^i(n-1)$ can be updated at the same time, and the current residual of MVDR beamforming is obtained simultaneously.

3 A Novel MVDR Systolic Array Processor

In Figure 1, a single fully-pipelined MVDR systolic array processor for four sensors and two constraints to receive the observed input vector and steering

$PE1$	$PE2$
$d \leftarrow (\beta^2 r^2 + \lvert x\rvert^2)^{\frac{1}{2}}$ $c \leftarrow \frac{\beta r}{d}$ $s \leftarrow \frac{x}{d}$ $r \leftarrow d$	$y \leftarrow -s\beta r + cx$ $r \leftarrow c\beta r + s^* x$
$PE3$	$PE4$
$x_{out} \leftarrow -s\frac{1}{\beta}r + cx_{in}$ $r \leftarrow c\frac{1}{\beta}r + s^* x_{in}$ $\eta_{out} \leftarrow \lvert r\rvert^2 + \eta_{in}$ $\delta_{out} \leftarrow c\delta_{in}$ $e_{out} \leftarrow s\delta_{in} r + e_{in}$	$e_{out} \leftarrow \frac{r e_{in}\beta^2}{\eta}$

Table 2: The Operation for Mode 1

$PE1$	$PE2$	$PE3$	$PE4$
$s \leftarrow \frac{x}{r}$ $c \leftarrow 1$	$y \leftarrow cx - sr$	$x_{out} \leftarrow x_{in}$ *if* $x_{in} \leftarrow 1$ *then* $r \leftarrow s*$	$e_{out} \leftarrow e_{in}$

Table 3: The Operation for Mode 2

vector, and then to instantaneously generate the updated residuals in parallel is given. The system needs two procedures which are the initialization and recursive updating. The initialization is divided into two parts: the QR decomposition that is called the mode 1 operation and the parallel multiplication performed as the forward substitution that is called the mode 2 operation. In the initialization, the mode 1 and mode 2 operations are required, while in recursive updating, only mode 1 is needed.

Figure 2 shows McWhirter and Shepherds' MVDR systolic array. As we can see the boundary cells send parameters into the internal cells as well as the next boundary cell. While the boundary cells in Figure 1 only send parameters to internal cells. The four processor elements are depicted in Figure 3. The mode 1 operation for the MVDR systolic array for each processor element is described in Table 2, and the mode 2 operation is presented in Table 3. The mode 1 operation is used to carry out the QR decomposition and parallel multiplication in both initialization and recursive updating. Under the mode 1 operation, the PE1 generates rotation coefficients c and s when zeroing out the observed input data. The PE2 performs the rotation of the received input data according to the rotation coefficients. The PE3 not only performs the loading operation but also carries out the parallel multiplication and accumulation operation to compute the non-normalized residuals. The PE4 performs the normalization to generate the residuals. In the mode 2 operation, the PE1 is used to generate parameters while the PE2 is employed to carry out the parallel multiplication and accumulation operation. The PE3 is simply used to store the matrix.

4 Comparisons

Since the MVDR algorithm in [1] is based on McWhirter's famous RLS algorithm [3], the current residual for MVDR beamforming by using that of the RLS algorithm can be obtained by

$$\hat{e}^i_{MVDR}(t_n) = -\gamma(n)\alpha_{MVDR}(n) = -\frac{1}{\beta}\gamma(n)\underline{b}^T(n)\underline{z}^i(n-1). \tag{16}$$

Now for the newly proposed MVDR beamforming algorithm, the current residual for Tang and Lius' MVDR beamforming algorithm has the form

$$\hat{e}^i_{MVDR}(t_n) = \underline{a}^H(n)\underline{z}^i(n) = \frac{1}{\beta}\underline{a}^H(n)A(n)\underline{z}^i(n-1). \tag{17}$$

Recall from (3) that $\hat{Q}(n)$, a unitary matrix, is given by

$$\hat{Q}(n) = \begin{bmatrix} A(n) & 0 & \underline{a}(n) \\ 0 & I & 0 \\ \underline{b}^T(n) & 0 & \gamma(n) \end{bmatrix}. \tag{18}$$

According to the property of the unitary matrix, we have

$$\begin{bmatrix} A^H(n) & 0 & \underline{b}^*(n) \\ 0 & I & 0 \\ \underline{a}^H(n) & 0 & \gamma(n) \end{bmatrix} \begin{bmatrix} A(n) & 0 & \underline{a}(n) \\ 0 & I & 0 \\ \underline{b}^T(n) & 0 & \gamma(n) \end{bmatrix} = I. \tag{19}$$

From (19),

$$\underline{a}^H(n)A(n) + \gamma(n)\underline{b}^T(n) = \underline{0}. \tag{20}$$

Therefore, $\underline{a}^H(n)A(n) = -\gamma(n)\underline{b}^T(n)$, and it is readily seen that (17) and (16) are equivalent.

Compared to McWhirter and Shepherds' MVDR systolic array processor, both PE1 and PE4 of our MVDR systolic array processor require one less multiplication, while PE3 needs three more multiplications and one more addition. Besides, our systolic array processor saves N transmission lines in the boundary cells. The throughput for both MVDR systolic array processors is the same when parallel processing is available in these processor elements.

5 Conclusions

A novel approach to design the systolic array for MVDR beamforming without incorporating the RLS is proposed. Same as McWhirter-Shepherd's systolic array, the proposed architecture is modular and fully-pipelined. The similarity and differences of both MVDR beamformers are also studied. In this paper, we show a new insight into MVDR beamforming algorithm developments and present a new approach to design the MVDR systolic array processor. Comparing to the conventional method using RLS in MVDR beamforming [1], we introduce a direct approach for MVDR systolic array processor design to obtain the same result.

References

[1] J. G. McWhirter and T. J. Shepherd, "Systolic Array Processor for MVDR Beamforming," *IEE proceedings*, Vol 136, No. 2, 1989, pp. 75-80.

[2] R. Schreiber, "Implementation of Adaptive Array Algorithms," *IEEE Trans. ASSP*, vol. ASSP-34, NO. 5, Oct. 1986, pp. 1038-1045.

[3] J. G. McWhirter, "Recursive Least-Squares Minimization Using a Systolic Array," *Proc. SPIE*, **431**, 2983, 1983, pp. 105-112.

[4] B. Yang and J.F. Bohme, "Systolic Implementation of A General Adaptive Array Processing Algorithm," *Proc. IEEE ICASSP*, 1988, pp. 2785-2788.

[5] A. W. Bojanczyk and F. T. Luk, "A Novel MVDR Beamforming Algorithm," *Proc. SPIE*, 1987, **826**, pp. 12-16.

[6] M. Moonen and J. Vandewalle, "A Square Root Covariance Algorithm for Constrained Recursive Least Squares Estimation," Submitted to *Journal of VLSI Signal Processing.*

[7] T.C. Liu and B. Van Veen, "A Modular Structure for Implementation of Linearly Constrained Minimum Variance Beamformers," *IEEE Trans. ASSP*, **39**, 10, Oct.1991, pp. 2343-2346.

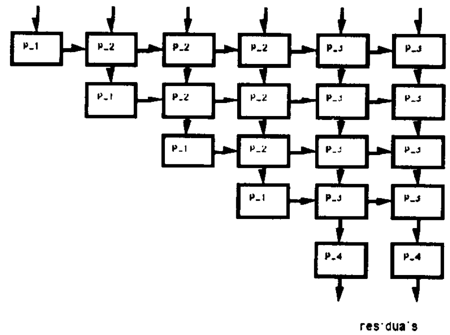

Figure 1: Novel MVDR Systolic Array Processor

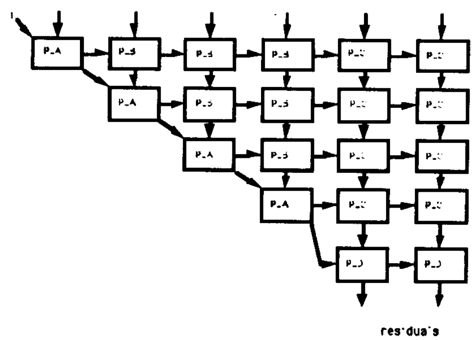

Figure 2: McWhirter's MVDR Systolic Array Processors

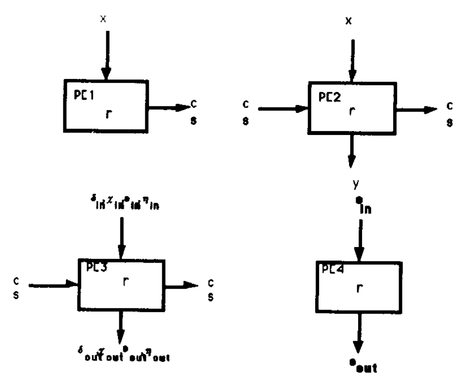

Figure 3: Processor Elements of Novel MVDR Systolic Array