MVDR BEAMFORMING WITH INVERSE UPDATING

Marc Moonen
Katholieke Universiteit Leuven, E.E. Dept, K.Mercierlaan 94, 3001 Heverlee -Belgium
e-mail: marc.moonen@esat.kuleuven.ac.be
www: http://www.esat.kuleuven.ac.be/sista/members/moonen.html

Ian K. Proudler
DERA, St Andrews Road, Malvern, Worcs., WR14 3PS, UK
e-mail: proudler@signal.dera.gov.uk

ABSTRACT

A stable alternative is described for the 'standard' systolic MVDR beamforming algorithm of McWhirter and Shepherd [4], which is known to suffer from linear round-off error build-up. The algorithm combines a so-called inverse QR-updating algorithm, with (part of) the McWhirter and Shepherd procedure. Similar to the McWhirter and Shepherd algorithm, it is amenable to parallel (pipelined) implementation. Unlike the McWhirter and Shepherd algorithm, it is stable numerically, and hence does not need repeated re-initializations.

1 INTRODUCTION

The minimum variance distortionless response (MVDR) beamforming problem amounts to minimising, in a least squares sense, the combined output from an antenna array subject to K independent linear equality constraints, each of which corresponds to a given 'look direction'. By 'independent', we mean that the minimum array output is computed for each constraint in turn. In other words, K independent recursive least squares problems have to be solved at once. The aim is to derive efficient (parallel) algorithms for this.

In [3], a parallel solution is given for the linearly constrained recursive least squares problem, with a constraint *pre*-processor coupled to a Gentleman-Kung triangular array [1]. For MVDR beamforming, one would then need K such triangular arrays, which is inefficient. In [4], however, McWhirter and Shepherd have shown how the beamforming problem can be solved with only one triangular array, coupled to a constraint *post*-processor. This is commonly accepted as the 'standard' solution to the problem. However, in [6], it has been shown that this approach suffers from linear roud-off error build-up, and hence needs repeated re-intialization.

In [6], an alternative procedure has been developed, which is based on so-called inverse QR-updating (for which a pipelined implementation has been developed in [7]), combined with a certain data pre-transformation. Here, we improve upon those results by picking a particular transformation matrix which allows a much easier exposition, and which is such that the correspondence with the McWhirter and Shepherd procedure is much more clearly displayed. The resulting algorithm is fully stable and so there is no need for repeated re-initializations. Furthermore, as the inverse updating structure is preserved, it is readily amenable to parallel (pipelined) implementation.

In section 2 and 3, the McWhirter and Shepherd procedure and the inverse updating procedure are reviewed. In section 4, it is shown how, partly based on the results of [6], the two procedures can be combined into a convenient MVDR algorithm. Part of the exposition (and notation) here is borrowed from [4] and [6].

2 McWHIRTER AND SHEPHERD PROCEDURE

The minimum variance distortionless response (MVDR) beamforming problem amounts to minimising, in a least squares sense, the combined output from an antenna array subject to K independent linear equality constraints, each of which corresponds to a given 'look direction'. At each sample time t_n, the aim is to evaluate the a posteriori residuals

$$e^{(i)}(n) \quad = \quad \mathbf{u}^T(n) \cdot \mathbf{w}^{(i)}(n) \qquad i = 1, \ldots, K$$

where $\mathbf{u}(n)$ is the p-element vector of signal samples received by the array at time t_n, and $\mathbf{w}^{(i)}(n)$ is the vector of weights which minimises the quantity

$$\|\mathbf{e}^{(i)}(n)\| \quad = \quad \|\mathbf{U}(n) \cdot \mathbf{w}^{(i)}(n)\|$$

subject to

$$\mathbf{c}^{(i)T} \cdot \mathbf{w}^{(i)}(n) = 1.$$

Here, $\mathbf{U}(n)$ is the weighted matrix of all data received up to time t_n, *i.e.* (with β the exponential 'forget' factor)

$$\mathbf{U}(n) \quad = \quad \begin{bmatrix} \beta^{n-1}\mathbf{u}^T(1) \\ \beta^{n-2}\mathbf{u}^T(2) \\ \vdots \\ \beta^0\mathbf{u}^T(n) \end{bmatrix}.$$

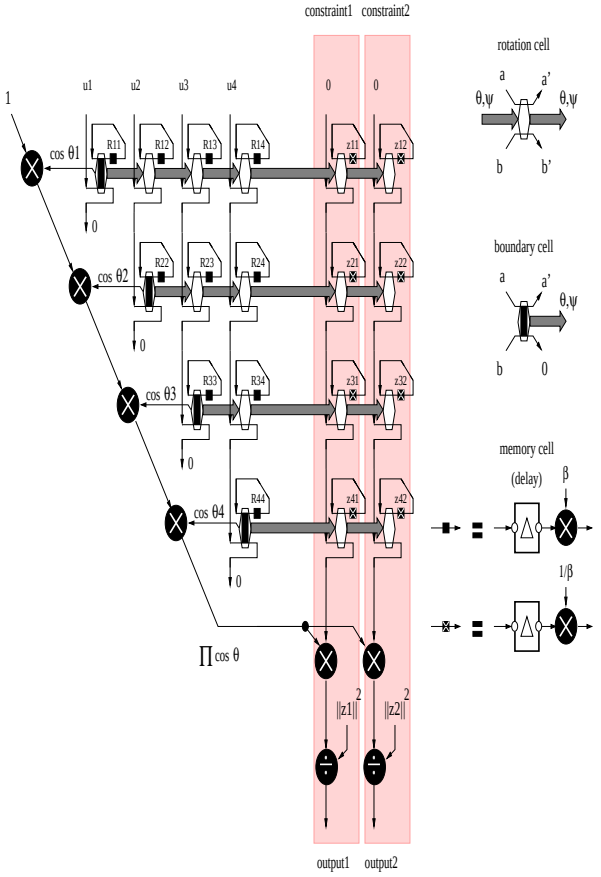

Figure 1: McWhirter and Shepherd algorithm

Assuming that a QR decomposition has been carried out on the data matrix $\mathbf{U}(n)$ so that

$$\mathbf{U}(n) \quad = \quad \mathbf{Q}(n) \cdot \left[\frac{\mathbf{R}(n)}{0} \right]$$

where $\mathbf{Q}(n)$ is unitary and $\mathbf{R}(n)$ is upper triangular, then it is easily shown that

$$e^{(i)}(n) \quad = \quad \mathbf{u}^T(n) \cdot \underbrace{\frac{1}{\|\mathbf{z}^{(i)}(n)\|^2} \mathbf{R}^{-1}(n)\mathbf{z}^{(i)}(n)}_{\mathbf{w}^{(i)}(n)}$$

where

$$\mathbf{z}^{(i)}(n) \quad = \quad \mathbf{R}^{-H}(n)\mathbf{c}^{(i)*}$$

In [4], McWhirter and Shepherd have shown how the beamforming problem can be solved with triangular updating (QR-updating) combined with constraint post-processor. A signal flow graph of this solution is shown in **Figure 1**. This commonly accepted as the 'standard' solution to the problem.

In Figure 1, the triangular array stores and updates the triangular matrix $\mathbf{R}(n)$, see [1]. The procedure is based on elementary unitary transformations (*cf.* 'rotation cells') of the form

$$\begin{bmatrix} a' \\ b' \end{bmatrix} = \begin{bmatrix} \cos\theta & e^{j\psi}\sin\theta \\ -e^{-j\psi}\sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} a \\ b \end{bmatrix}.$$

Transformations are computed on the diagonals (such that one output element is zeroed) and then applied to the other elements in the same row. (The angles θ and ψ are calculated implicitly and other parameters are in fact passed along the rows but for convenience we label the data with θ and ψ.)

For each 'look direction' (constraint), a right-hand side column is added that stores and updates $\mathbf{z}^{(i)}(n) = \mathbf{R}^{-H}(n) \cdot \mathbf{c}^{(i)*}$ In [4], it is proved that the right-hand side columns can be updated by means of the unitary transformations that are computed in the triangular array (with a 0-input at the top of each right-hand side column, as indicated in the graph). Furthermore, the beamformer outputs can be computed as indicated at the bottom of the Figure, *i.e.* by multiplying the output of each right-hand side column with a factor γ (equal to the product of the cosines of all the rotation angles) and dividing the result by the squared norm of the column (the computation of these norms is left out in the graph, but can easily be added as a top-to-bottom accumulation). Note that the triangular part has the usual weighting with β, whereas the right-hand side columns have a weighting with $\frac{1}{\beta}$ (because of the $\mathbf{R}^{-H}(n)$ in the formula for $\mathbf{z}^{(i)}(n)$). We assume that the reader is familiar with this procedure, and refer to [4] for further details.

In [6], it has been shown that this approach suffers from linear round-off error build-up. In particular, the equation $\mathbf{z}^{(i)}(n) = \mathbf{R}^{-H}(n) \cdot \mathbf{c}^{(i)*}$ indicates that the procedure has some redundancy (as $\mathbf{c}^{(i)*}$ is a fixed vector, and the $\mathbf{R}^{-H}(n)$ is already available from the stored $\mathbf{R}(n)$.). In a finite wordlength implementation, the above equation will be satisfied only up to a round-off error, which is observed to grow in each iteration. As a result, the McWhirter and Shepherd procedure needs repeated re-intialization, which is undesirable. Here, an alternative scheme is presented, which does not need re-intialization.

3 INVERSE QR-UPDATING

The inverse updating procedure [5] is shown in **Figure 2**. Instead of $\mathbf{R}(n)$, we now store and update $\mathbf{R}^{-H}(n)$ (denoted here as $\mathbf{S}(n)$). It has been shown that the updating transformations are the same as those given in Figure 1, but now derived in a different fashion (by means of the matrix-vector product $\mathbf{v} = \mathbf{R}^{-H}(n) \cdot \mathbf{u}(n)$, as indicated). A mathematical description is as follows :

Algorithm Inverse Updating [5]

Given $\mathbf{R}^{-H}(n-1)$

Input $\mathbf{u}(n)$

Step 1. Form the matrix-vector product

$$\mathbf{v} = -\frac{1}{\beta}\mathbf{R}^{-H}(n-1) \cdot \mathbf{u}(n)$$

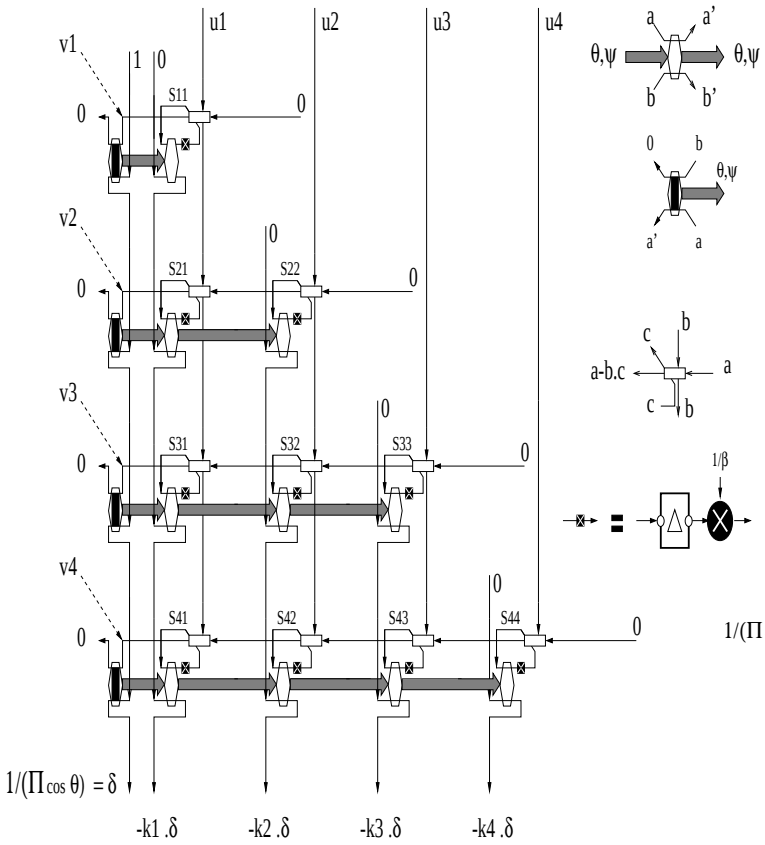

Figure 2: Inverse updating algorithm

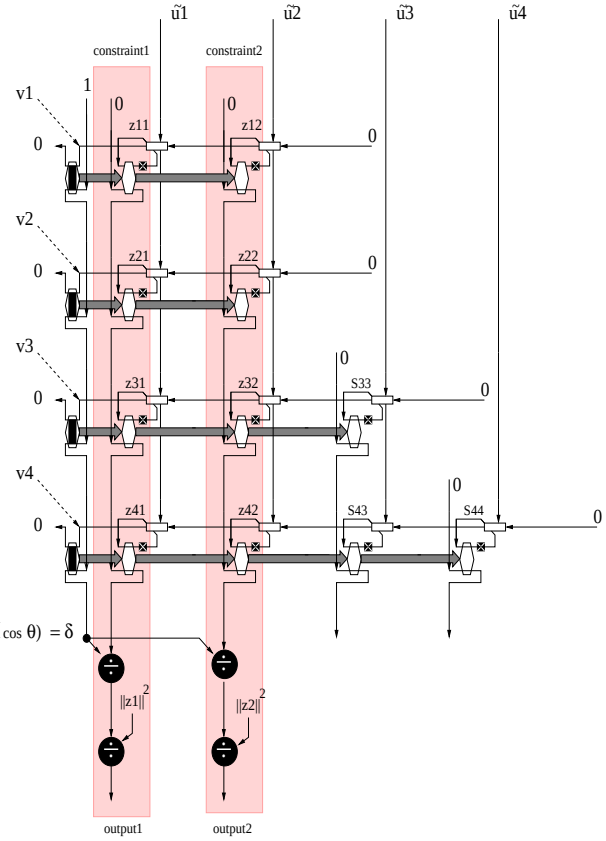

Figure 3: MVDR algorithm

Step 2. For $i = 1, \ldots, p$ determine unitary transformations $\mathbf{Q}_i$ so that

$$\begin{bmatrix} \delta \\ \hline 0 \end{bmatrix} \Leftarrow \mathbf{Q}_p \mathbf{Q}_{p-1} \ldots \mathbf{Q}_1 \cdot \begin{bmatrix} 1 \\ \hline \mathbf{v} \end{bmatrix}$$

Step 3. Update $\mathbf{R}^{-H}$

$$\begin{bmatrix} * \\ \hline \mathbf{R}^{-H}(n) \end{bmatrix} \Leftarrow \mathbf{Q}_p \mathbf{Q}_{p-1} \ldots \mathbf{Q}_1 \cdot \begin{bmatrix} 0 \\ \hline \frac{1}{\beta}\mathbf{R}^{-H}(n-1) \end{bmatrix}$$

In *Step 2*, $\mathbf{Q}_i$ is a plane transformation acting upon the first and the $(i+1)$-st component of $\mathbf{Q}_{i-1} \ldots \mathbf{Q}_1 \cdot \begin{bmatrix} 1 \\ \hline \mathbf{v} \end{bmatrix}$, such that the $(i+1)$-st component is zeroed. For details, we refer to [5].

It can easily be proved that the δ, resulting from *Step 2*, satisfies

$$\delta = \frac{1}{\prod_{i=1}^{p} \cos\theta_i}$$

which will be useful in the MVDR procedure.

In [8], it is shown that this procedure is numerically stable (despite the exponential weighting with $\frac{1}{\beta}$). In [7], it is shown that this algorithm can be accomodated for parallel (fully pipelined) implementation.

4 INVERSE QR-UPDATING BASED MVDR BEAMFORMING

In **Figure 3**, it is shown how Figure 1 and 2 can be combined into a convenient MVDR algorithm. Now, instead of $\mathbf{R}(n)^{-H}$, we store and update

$$\tilde{\mathbf{R}}(n)^{-H} = \mathbf{R}(n)^{-H} \cdot \left[\begin{array}{c|c} C & \begin{matrix} 0 \\ I \end{matrix} \end{array}\right]$$

where

$$C = \left[\begin{array}{c|c|c} \mathbf{c}^{(1)*} & \ldots & \mathbf{c}^{(K)*} \end{array}\right].$$

The stored matrix thus contains the $\mathbf{z}^{(i)}(n)$ -columns of Figure 1 (left-hand part) as well as the (remaining) right-hand part of $\mathbf{R}(n)^{-H}$ (Figure 2).

To obtain consistent results, the input vectors have to be pre-transformed, *i.e.* the vectors fed into the array are

$$\tilde{\mathbf{u}}(n) = \left[\begin{array}{c|c} C & \begin{matrix} 0 \\ I \end{matrix} \end{array}\right]^{-1} \cdot \mathbf{u}(n).$$

(Note that the transformation matrix is a fixed matrix.) For this to be possible, we assume that the topmost K rows of C constitute an invertible $K \times K$ matrix. Whenever the look directions are linearly independent, a permutation can be applied such that this is indeed the case.

It is readily checked that, because of the pre-transformation, the computed $\mathbf{v}$ is unchanged

$$\mathbf{v} = -\frac{1}{\beta}\tilde{\mathbf{R}}(n)^{-H} \cdot \tilde{\mathbf{u}}(n) = -\frac{1}{\beta}\mathbf{R}(n)^{-H} \cdot \mathbf{u}(n),$$

so that the transformations $Q_1, \ldots, Q_p$ are also unchanged. ¿From this it follows that Figure 3 is indeed a valid combination of Figures 1 and 2. A mathematical algorithm description is as follows :

Algorithm MVDR [5]

Given $\tilde{\mathbf{R}}^{-H}(n-1)$

Input $\mathbf{u}(n)$

Step 1. Pre-transformation

$$\tilde{\mathbf{u}}(n) = \left[\begin{array}{c|c} C & \frac{0}{I} \end{array}\right]^{-1} \cdot \mathbf{u}(n)$$

Step 2. Form the matrix-vector product

$$\mathbf{v} = -\frac{1}{\beta}\tilde{\mathbf{R}}^{-H}(n-1) \cdot \tilde{\mathbf{u}}(n)$$

Step 3. For $i = 1, \ldots, p$ determine unitary transformations $\mathbf{Q}_i$ so that

$$\left[\frac{\delta}{0}\right] \Leftarrow \mathbf{Q}_p\mathbf{Q}_{p-1}\ldots\mathbf{Q}_1 \cdot \left[\frac{1}{\mathbf{v}}\right]$$

Step 4. Update $\tilde{\mathbf{R}}^{-H}$

$$\left[\frac{\mathbf{k}^T}{\tilde{\mathbf{R}}^{-H}(n)}\right] \Leftarrow \mathbf{Q}_p\mathbf{Q}_{p-1}\ldots\mathbf{Q}_1 \cdot \left[\frac{0}{\frac{1}{\beta}\tilde{\mathbf{R}}^{-H}(n-1)}\right]$$

Step 5. Compute outputs $i = 1, \ldots, K$

$$e^{(i)}(n) = \mathbf{k}^{(i)} \cdot \frac{1}{\delta \cdot ||\mathbf{z}^{(i)}(n)||^2}$$

where $\mathbf{z}^{(i)}(n)$ is the i-th column of $\tilde{\mathbf{R}}^{-H}(n)$, and $\mathbf{k}^{(i)}$ is the i-th component of $\mathbf{k}$.

The resulting algorithm is fully stable, which follows from the proved stability of the inverse QR-updating procedure [8], and so there is no need for repeated re-initializations. Furthermore, as the inverse updating structure is preserved, it is readily amenable to parallel (pipelined) implementation, see [7].

Suffice it to say that the MVDR procedure of [6] works with a more general data pre-transformation matrix T. A more extensive mathematical derivation is then needed to prove that the algorithm actually works. Also, there is no obvious connection between the resulting algorithm and the McWhirter and Shepherd procedure. Here, with $T = \left[\begin{array}{c|c} C & \frac{0}{I} \end{array}\right]$, it is readily verified that the algorithm (figure 3) is a valid combination of the McWhirter and Shepherd procedure (figure 1) and the inverse QR-updating procedure (figure 2), and the connection between the resulting algorithm and the McWhirter and Shepherd procedure is very apparent.

5 CONCLUSIONS

A stable alternative is described for the MVDR beamforming algorithm of McWhirter and Shepherd. This algorithm employs transformed data in an inverse QR-updating scheme. This algorithm is stable numerically as well as amenable to parallel (pipelined) implementation.

6 ACKNOWLEDGEMENT

Marc Moonen is a Research Associate with the F.W.O. (Fund for Scientific Research- Flanders). This research work was carried out in the frame of the Belgian State, Prime Minister's Office (F.D.W.T.C), Interuniversity Poles of Attraction Programme *IUAP P4-02*, the *Concerted Research Action MIPS* ('Model-based Information Processing Systems') of the Flemish Government, *Research Project FWO nr. G.0295.97* ('Design and implementation of adaptive digital signal processing algorithms for broadband applications'). The scientific responsibility is assumed by its authors.

REFERENCES

[1] Gentleman, W.M., and Kung, H.T. : 'Matrix triangularization by systolic arrays', *Proc. SPIE*, Real-Time Signal Processing IV, 1982, Vol. 298, pp 19-26.

[2] Griffiths, L.J., and Jim, C.W. : 'An alternative approach to linearly constrained adaptive beamforming', *IEEE Trans.* AP-30, 1982, pp 27-34.

[3] McWhirter, J.G., and Shepherd, T.J. : 'A systolic array for linearly constrained least-squares problems', *Proc. SPIE*, Advanced Algorithms and Architectures for Signal Processing, 1986, Vol. 696, pp 80-87.

[4] McWhirter, J.G., and Shepherd, T.J. : 'Systolic array processor for MVDR beamforming', *IEE Proceedings*, 1989, Vol. 136, Pt. F, pp 75-80.

[5] Pan, C.T., and Plemmons, R.J. : 'Least squares modifications with inverse factorization: parallel implications', *J. Computational and Applied Mathematics*, 1989, Vol. 27, pp. 109-127.

[6] Moonen, M. : 'Systolic MVDR beamforming with inverse updating', *IEE Proceedings-F, Radar and Signal Processing*, Vol. 140 (1993), No. 3, pp 175-178.

[7] Moonen, M., and McWhirter, J.G. : 'A systolic array for recursive least squares by inverse updating', *Electronics Letters*, Vol. 29 (1993), No. 13, pp 1217-1218.

[8] Verhaegen, M.H. : 'Round-off error propagation in four generally applicable Recursive least-squares estimation schemes', *Automatica*, 1989, Vol. 25, pp 437-444.