QR-RLS BASED MINIMUM VARIANCE DISTORTIONLESS RESPONSES BEAMFORMER

Zhu Liang Yu, Wee Ser

Center for Signal Processing
Nanyang Technological University
Singapore, 639798

Susanto Rahadja

Institute for Infocomm Research,
21 Heng Mui Keng Terrace,
Singapore, 119613

ABSTRACT

In this paper, a QR-RLS based Minimum Variance Distortionless Responses (MVDR) method, and its systolic array processor, are proposed. The QR-RLS based MVDR has many advantages, such as numerical stability, computational efficiency and pipelined structure in implementation. We also point out that the conventional method, MVDR using QR-RLS method by directly forcing the desired signal to zero is not correct. Numerical experiments are carried out to illustrate the effectiveness of the proposed method.

1. INTRODUCTION

Minimum variance distortionless response (MVDR) beamformer [1] is a key algorithm in array signal processing. In the last decades, research on its implementing using QR decomposition has been very active [2–5]. The QR based method can easily be transformed to systolic implementation [6], which is very attractive for the demand for sophisticated system with high throughput rate and superior numerical accuracy. The potential advantages of QR based method include numerical stability, computational efficiency and fully pipelined structure.

A systolic array processor for MVDR is proposed in [4]. This method does not require to compute the weight vector for computing the residual signal. By comparing the recursive least square (RLS) algorithm with MVDR, it seems easy to obtain the residual of the MVDR from that of the RLS algorithm by forcing the desired input data to be zero [4, 7]. However, in this paper, we indicate that the directly mapping RLS to MVDR by forcing desired signal to be zero has some problems. When the forgetting factor of the RLS method is selected to be less than 1, the resulting method will produce zero tap-weight vector eventually. The beamformer fails to work. After a modification of QR-RLS MVDR [7], a new method is proposed in this paper. The theoretical and numerical studies show that the proposed method produces correct output and demonstrates high numerical performance. Moreover, its systolic implementation is also shown briefly.

This paper is organized as following. In Section 2, we briefly compare the RLS and MVDR method. The mapping between these two methods is also discussed. In Section 3, we analysis the mapping in Section 2 and indicate that it is not correct. A modified QR-RLS MVDR method is proposed to solve the problem. Some numerical experiments are carried out in Section 4 to show the performance of the proposed method. Followed by a brief conclusion in Section 5.

2. QR-RLS AND MVDR BEAMFORMER

The RLS adaptive filter is formulated as [7]

$$\begin{aligned} \min_{\mathbf{w}(n)} \quad & \sum_{i=1}^{n} \lambda^{n-i}|e(i)|^2, \\ e(i) &= d(i) - \mathbf{w}^H(n)\mathbf{u}(i), \end{aligned} \tag{1}$$

where $0 < \lambda \leq 1$ is the exponential forgetting factor, $e(i)$ is the difference between the desired signal $d(i)$ and the output signal of adaptive filter $\mathbf{w}^T(n)\mathbf{u}(i)$. $\mathbf{w}(n)$ is tap-weight vector and $\mathbf{u}(i)$ is the tap-input vector,

$$\begin{aligned} \mathbf{w}(n) &= [w_0(n)\, w_1(n)\, \cdots\, w_{M-1}(n)]^T, \\ \mathbf{u}(i) &= [u(i)\, u(i-1)\, \cdots\, u(i-M+1)]^T. \end{aligned} \tag{2}$$

The optimal solution $\hat{\mathbf{w}}(n)$ of (1) is given as

$$\hat{\mathbf{w}}(n) = \Phi^{-1}(n)\mathbf{z}(n), \tag{3}$$

where

$$\begin{aligned} \Phi(n) &= \lambda\Phi(n-1) + \mathbf{u}(n)\mathbf{u}^H(n), \\ \mathbf{z}(n) &= \lambda\mathbf{z}(n-1) + \mathbf{u}(n)d^*(n). \end{aligned} \tag{4}$$

Using the matrix inversion Lemma [7], a recursive least square method is derived. The optimal solution of the least square problem can also be solved using QR based method due to its numerical stability, pipeline implementation and robustness to finite precision problem. As shown in [7, 8], the QR-RLS algorithm is given by

$$\begin{bmatrix} \lambda^{\frac{1}{2}}\Phi^{\frac{1}{2}}(n-1) & \mathbf{u}(n) \\ \lambda^{\frac{1}{2}}\mathbf{p}^H(n-1) & d(n) \\ \mathbf{0}^T & 1 \end{bmatrix} \Theta(n) = \begin{bmatrix} \Phi^{\frac{1}{2}}(n) & \mathbf{0} \\ \mathbf{p}^H(n) & \frac{\xi(n)}{\gamma^{\frac{1}{2}}(n)} \\ \mathbf{u}^H(n)\Phi^{\frac{-H}{2}}(n) & \gamma^{\frac{1}{2}}(n) \end{bmatrix}, \tag{5}$$

where $\Phi^{\frac{1}{2}}(n-1)$ is the square-root of $\Phi(n-1)$, i.e., $\Phi(n-1) = \Phi^{\frac{1}{2}}(n-1)\Phi^{\frac{H}{2}}(n-1)$. The matrix $\Theta(n)$ is any unitary rotation that operates on the elements of the input data vector $\mathbf{u}(n)$ in the prearray, annihilating then one by one so as to produce a block zero empty in the top block row of the postarray. The vector $\mathbf{p}(n)$ is defined as

$$\mathbf{p}(n) = \Phi^{-\frac{1}{2}}(n)\mathbf{z}(n).$$

To apply the well studied QR-RLS method in adaptive beamformeing, in this section, the least square MVDR beamformer is compared with the RLS method. The MVDR beamformer is formulated as

$$\begin{aligned} \min_{\mathbf{w}(n)} \quad & \sum_{i=1}^{n} \lambda^{n-i}|e(i)|^2, \\ e(i) &= \mathbf{w}^H(n)\mathbf{u}(i), \\ s.t.\ & \mathbf{w}^H(n)\mathbf{s}(\theta_0) = 1, \end{aligned} \tag{6}$$

where $\mathbf{s}(\theta_0)$ is the array steering vector corresponding to the direction of the target source, θ_0. The optimal solution $\hat{\mathbf{w}}(n)$ is given by

$$\begin{aligned} \hat{\mathbf{w}}(n) &= \frac{\Phi^{-1}(n)\mathbf{s}(\theta)}{\mathbf{s}^H(\theta)\Phi^{-1}(n)\mathbf{s}(\theta)} \\ &= \frac{\Phi^{-H/2}(n)\mathbf{a}(n)}{||\mathbf{a}(n)||^2}, \end{aligned} \tag{7}$$

1-4244-0469-X/06/$20.00 ©2006 IEEE

QR-RLS	MCDR	Description
$e(n)$	$-e'(n)$	Estimation error
$d(n)$	0	Desired signal
$\mathbf{p}(n)$	$\mathbf{a}(n)$	Auxiliary vector
$\mathbf{u}(n)$	$\mathbf{u}(n)$	Snapshot

Table 1: Correspondence between the QR-RLS adaptive filtering and MVDR beamforming variables

where

$$\mathbf{a}(n) = \Phi^{-\frac{1}{2}}(n)\mathbf{s}(\theta). \tag{8}$$

Therefore, the output error signal $e(n)$ is

$$e(n) = \hat{\mathbf{w}}^H(n)\mathbf{u}(n) = \frac{\mathbf{a}^H(n)\Phi^{-\frac{1}{2}}(n)\mathbf{u}(n)}{||\mathbf{a}(n)||^2} \triangleq \frac{e'(n)}{||\mathbf{a}(n)||^2} \tag{9}$$

If we reexpress $e(n)$ as

$$e(n) = d(n) - \hat{\mathbf{w}}^H(n)\mathbf{u}(n), \tag{10}$$

compare it with (1), the QR-RLS based MVDR can be deduced by replacing the variables in QR-RLS in (5) by the corresponding variable of MVDR as shown in Table 1. Detail discussion of this mapping can be found in [7].

3. PROPOSED METHOD

Unfortunately, the QR-RLS MVDR in [7] which is derived using the idea presented in Section 2 is not correct. The algorithm can only work for $\lambda = 1$. If $\lambda < 1$, the output error $e'(n)$ as well as $||\mathbf{a}(n)||^2$ approach to zeros after sufficient number of iterations.

The reason why QR-RLS MVDR in [7] cannot work can be explained as following. Compare the output errors in (10), (6) and (1), it seems that the MVDR can be implemented as RLS method by forcing $d(n)$ in (10) to zero. However, since $d(n)$ also affect the calculation of the optimal weight $\hat{\mathbf{w}}(n)$, forcing $d(n)$ to zero will result in zero tap-weight vector. Refer to (4), we find that if $d(n) = 0$, the vector $\mathbf{z}(n)$ can be expressed as

$$\mathbf{z}(n) = \lambda\mathbf{z}(n-1) = \lambda^n\mathbf{z}(0)$$

where $\mathbf{z}(0)$ is the initial vector. If $\lambda < 1$, we have

$$\lim_{n\to\infty}\mathbf{z}(n) = \mathbf{0}$$

Therefore, the optimal solution $\hat{\mathbf{w}}(n)$ approaches to zero. Similar conclusion can also be derived that $||\mathbf{a}(n)||^2$ approaches zero.

This is a basic concept in adaptive filtering. It is known that the adaptive filter only works when the desired signal is correlated with the input signal, otherwise, zero tap-weight vector produces. If the desired signal $d(n)$ is forced to zero, it must be uncorrelated with any other signals. Consequently, it produces zero tap-weight. Fortunately, for the MVDR beamformer, the $\mathbf{z}(n)$ can be considered as the array steering vector $\mathbf{s}(\theta_0)$, which is given and fixed. The recursive update of $\mathbf{s}(\theta_0)$ is not required. Therefore, we can modify the QR-RLS MVDR as the form shown in Theorem 1. It produces correct output signal and demonstrates superior numerical properties.

Theorem 1. *The QR-RLS based MVDR beamformer can be formulated as following,*

$$\begin{bmatrix} \lambda^{\frac{1}{2}}\Phi^{\frac{1}{2}}(n-1) & \mathbf{u}(n) \\ \lambda^{\frac{-1}{2}}\mathbf{a}^H(n-1) & 0 \\ \mathbf{0}^T & 1 \end{bmatrix} \Theta(n) = \begin{bmatrix} \Phi^{\frac{1}{2}}(n) & \mathbf{0} \\ \mathbf{a}^H(n) & -\frac{e'(n)}{\gamma^{\frac{1}{2}}(n)} \\ \mathbf{u}^H(n)\Phi^{-\frac{H}{2}}(n) & \gamma^{\frac{1}{2}}(n) \end{bmatrix}$$

where

$$\mathbf{a}(n) = \Phi^{-\frac{1}{2}}(n)\mathbf{s}(\theta_0).$$

The matrix $\Theta(n)$ is any unitary rotation that operates on the elements of the input data vector $\mathbf{u}(n)$ in the prearray, annihilating then one by one so as to produce a block zero empty in the top block row of the postarray. The alternative output signal $e'(n)$ is obtained as

$$e'(n) = -\left[-\frac{e'(n)}{\gamma^{\frac{1}{2}}(n)}\right]\left[\gamma^{\frac{1}{2}}(n)\right],$$

and the output residual signal $e(n)$ is

$$e(n) = \frac{e'(n)}{||\mathbf{a}(n)||^2}.$$

Proof. Refer to Appendix A. □

Remark 1. *The quantities for calculating $e'(n)$ can be obtained in the transformed matrix. The residual signal $e(n)$ can be calculated by normalized $e'(n)$ by the norm of $\mathbf{a}(n)$ as shown in (9). As shown in the Appendix A, the proposed method does not change $\mathbf{s}(\theta_0)$ during updating for any value of λ. The problem of the conventional QR-RLS MVDR is solved.*

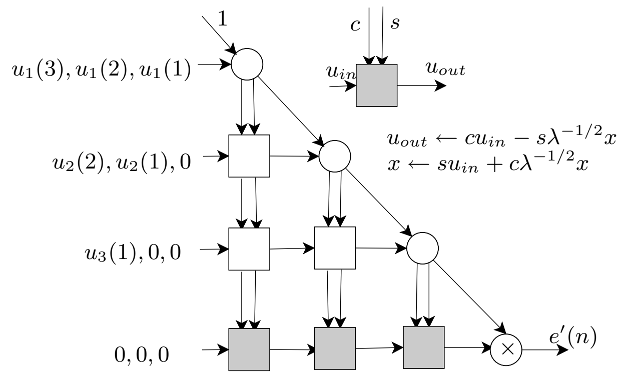

Fig. 1: Systolic array implementation of the proposed beamformer.

The systolic implementation of the proposed method is shown in Fig. 1. For simplicity, we assume that there are three sensors. Compared with the implementation in [7], the difference is that the last processing element row (grayed ones) is changed. The other processing elements can be found in [7].

4. NUMERICAL STUDY

In this section, some numerical experiments were carried out to illustrate the performance of the proposed method. A uniform linear array (ULA) with eight sensors and half-wavelenght inter-elements space is used in simulation. The target signal is simulated as random signal with Gaussian distribution. The direction of the target signal is assumed from the broadside of the ULA. The performance of QR-RLS [7] and the proposed QR-RLS is studied. In the simulations, the alternative output signal $e'(n)$ and the norm corrected output signal $e(n)$ are both illustrated. Moreover, we also compare the differences of $e'(n)$ or $e(n)$ between the ideal signals.

In the first experiment, the performance of each algorithm versus the forgetting factor is studied. In the simulation, we use 100 snapshots. The results show in Fig. 2 was obtained when $\lambda = 0.999999$. It seems that both algorithms work. It should be noted that the error

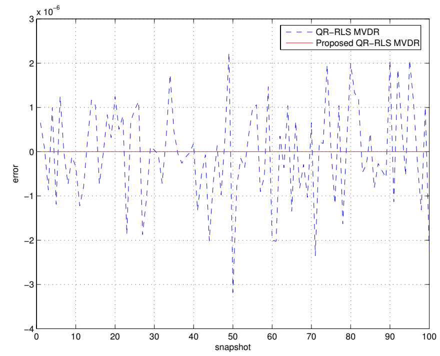

Fig. 2: Comparison of the error between $e'(n)$ and the ideal one when $\lambda = 0.999999$

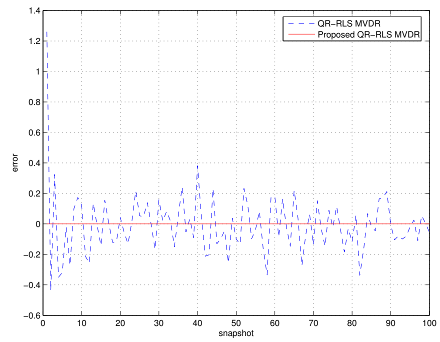

Fig. 3: Comparison of the error between $e'(n)$ and the ideal one when $\lambda = 0.8$

of the proposed method is almost zero. This is similar in other experiments. However, the QR-RLS MVDR method produces output with larger error than that of the proposed method. We change $\lambda = 0.8$ and show the results in Fig. 3 and 4. It is clear that QR-RLS MVDR method has larger error, especially for the output signal $e(n)$. The reason is that with small λ, $||\mathbf{a}_n||^2$ decreases quickly. Since the output signal $e(n)$ is the the divided value of $e'(n)$ to $||\mathbf{a}(n)||^2$, small value of $||\mathbf{a}(n)||^2$ causes serious numerical problem.

In Fig. 5, we show the performance of each beamformer versus different value of λ. It is clear that error of QR-RLS MVDR beamformer increases with the decreasing of λ. Strictly speaking, it only works when $\lambda = 1$. This conclusion also be supported by the simulation results shown in Fig. 2 - 4.

In Fig. 6 and 7, the simulation was carried out with $\lambda = 0.999$. In the beginning of the filtering, it seems QR-RLS MVDR produces correct output. However, with the increasing iteration number, the alternative output of QR-RLS MVDR approaches zero and its output diverges. This diverges is due to numerical problem. For the proposed method, it works well. This experiment shows that the conventional QR-RLS MVDR fails to work even with $\lambda \approx 1$ after sufficient number of iterations.

5. CONCLUSION

Direct mapping QR-RLS to QR-RLS MVDR beamformer was studied in some literatures. In this paper, we proved that the direct mapping by forcing desired signal of QR-RLS to be zero is not correct. The conventional QR-RLS MVDR method can only works when $\lambda = 1$. If $\lambda < 1$, the output signal of QR-RLS MVDR has serious numerical problem, especially when the system works for long time.

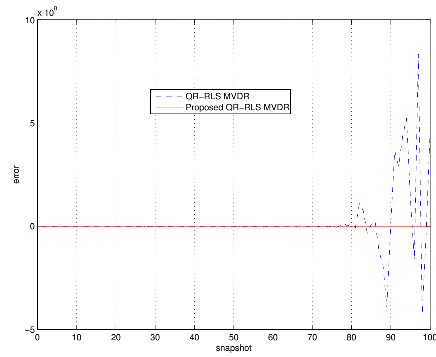

Fig. 4: Comparison of the error between $e(n)$ and the ideal one when $\lambda = 0.8$

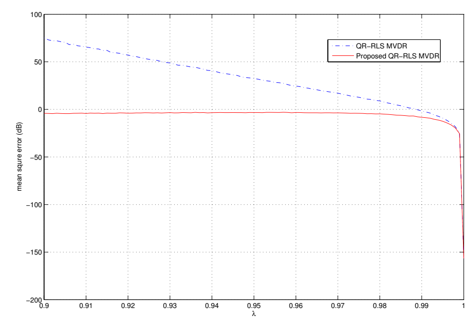

Fig. 5: Average error power versus different value of λ (1000 iterations)

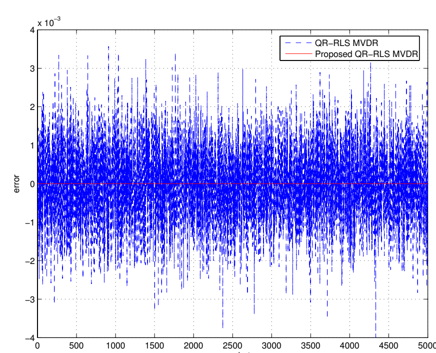

Fig. 6: Comparison of the error between $e'(n)$ and the ideal one when $\lambda = 0.999$

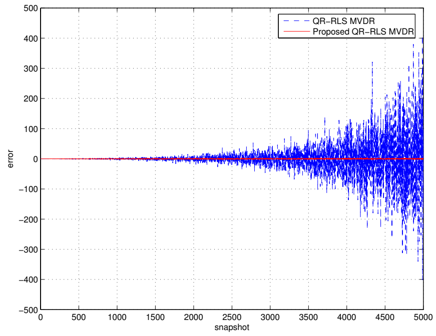

Fig. 7: Comparison of the error between $e(n)$ and the ideal one when $\lambda = 0.999$

The proposed method solves this numerical problem efficiently. It is obtained by transforming QR-RLS to MVDR problem with some additional modifications. The theoretical and numerical studies indicate that the proposed method has high performance and stable numerical properties.

A. THE PROOF OF THEOREM 1

Proof.

$$\mathbf{C}_1 = \begin{bmatrix} \lambda^{\frac{1}{2}}\Phi^{\frac{1}{2}}(n-1) & \mathbf{u}(n) \\ \lambda^{\frac{-1}{2}}\mathbf{a}^H(n-1) & 0 \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \lambda^{\frac{1}{2}}\Phi^{\frac{1}{2}}(n-1) & \mathbf{u}(n) \\ \lambda^{\frac{-1}{2}}\mathbf{a}^H(n-1) & 0 \\ \mathbf{0}^T & 1 \end{bmatrix}^H$$
$$= \begin{bmatrix} \lambda\Phi(n-1) + \mathbf{u}(n)\mathbf{u}^H(n) & \Phi^{\frac{1}{2}}(n-1)\mathbf{a}(n-1) & \mathbf{u}(n) \\ \mathbf{a}^H(n-1)\Phi^{\frac{1}{2}}(n-1) & \lambda^{-1}||\mathbf{a}(n-1)||^2 & 0 \\ \mathbf{u}^H(n) & 0 & 1 \end{bmatrix},$$

$$\mathbf{C}_2 = \begin{bmatrix} \Phi^{\frac{1}{2}}(n) & \mathbf{0} \\ \mathbf{a}^H(n) & \frac{-e'(n)}{r^{\frac{1}{2}}(n)} \\ \mathbf{u}^H(n)\Phi^{\frac{-1}{2}}(n) & r^{\frac{1}{2}}(n) \end{bmatrix} \begin{bmatrix} \Phi^{\frac{1}{2}}(n) & \mathbf{0} \\ \mathbf{a}^H(n) & \frac{-e'(n)}{r^{\frac{1}{2}}(n)} \\ \mathbf{u}^H(n)\Phi^{\frac{-1}{2}}(n) & r^{\frac{1}{2}}(n) \end{bmatrix}^H$$
$$= \begin{bmatrix} \Phi(n) & \Phi^{\frac{1}{2}}(n)\mathbf{a}(n) & \mathbf{u}(n) \\ \mathbf{a}^H(n)\Phi^{\frac{H}{2}}(n) & c_{22} & c_{23} \\ \mathbf{u}^H(n) & c_{23}^* & c_{33} \end{bmatrix},$$

where

$$c_{22} = ||\mathbf{a}(n)||^2 + |e'(n)|^2 r^{-1}(n),$$
$$c_{23} = \mathbf{a}^H(n)\Phi^{\frac{-1}{2}}(n)\mathbf{u}(n) - e'(n),$$
$$c_{33} = \mathbf{u}^H(n)\Phi^{-1}(n)\mathbf{u}(n) + r(n).$$

According to Lemma in [8], we need to prove that $\mathbf{C}_1 = \mathbf{C}_2$. For the $(1,1)$th element,

$$\Phi(n) = \lambda\Phi(n-1) + \mathbf{u}(n)\mathbf{u}^H(n).$$

For the $(1,2)$th element,

$$\mathbf{s}(\theta_0) = \Phi^{\frac{1}{2}}(n)\mathbf{a}(n) = \Phi^{\frac{1}{2}}(n-1)\mathbf{a}(n-1),$$

therefore, in the recursive update, no effect of forgetting factor on the array steering vector $\mathbf{s}(\theta_0)$. This guarantees that the output signal does not approach zero in recursive updating.

For the $(2,2)$th element,

$$||\mathbf{a}(n)||^2 + |e'(n)|^2 r^{-1}(n)$$
$$= \mathbf{a}^H(n)\mathbf{a}(n) + \frac{\mathbf{a}^H(n)\Phi^{\frac{-1}{2}}(n)\mathbf{u}(n)\mathbf{u}^H(n)\Phi^{\frac{-1}{2}}(n)\mathbf{a}(n)}{1 - \mathbf{u}^H(n)\Phi^{-1}(n)\mathbf{u}(n)}$$
$$= \mathbf{a}^H(n)\left(\mathbf{I} + \frac{\Phi^{\frac{-1}{2}}(n)\mathbf{u}(n)\mathbf{u}^H(n)\Phi^{\frac{-1}{2}}(n)}{1 - \mathbf{u}^H(n)\Phi^{-1}(n)\mathbf{u}(n)}\right)\mathbf{a}(n)$$
$$= \mathbf{a}^H(n)\left(\mathbf{I} - \Phi^{\frac{-1}{2}}(n)\mathbf{u}(n)\mathbf{u}^H(n)\Phi^{\frac{-1}{2}}(n)\right)^{-1}\mathbf{a}(n)$$
$$= \mathbf{a}^H(n)\Phi^{\frac{1}{2}}(n)\left(\Phi(n) - \mathbf{u}(n)\mathbf{u}^H(n)\right)^{-1}\Phi^{\frac{1}{2}}(n)\mathbf{a}(n)$$
$$= \mathbf{a}^H(n)\Phi^{\frac{1}{2}}(n)\lambda^{-1}\Phi^{-1}(n-1)\Phi^{\frac{1}{2}}(n)\mathbf{a}(n)$$
$$= \lambda^{-1}\mathbf{a}^H(n-1)\mathbf{a}(n-1)$$
$$= \lambda^{-1}||\mathbf{a}(n-1)||^2$$

For the $(2,3)$th element,

$$e'(n) = \mathbf{a}^H(n)\Phi^{\frac{-1}{2}}(n)\mathbf{u}(n)$$

which is defined in (9).

Since $r(n)$ is the conversion factor [7], the proof of the $(3,3)$th element is straightforward.

The matrices $\mathbf{C}_1$ and $\mathbf{C}_2$ are both Hermitian matrices. The other elements can be easily verified. □

B. REFERENCES

[1] J. E. Hudson, *Adaptive Array Principles*. Peter Peregrinus Ltd., 1981.

[2] R. Schreiber, “Implementation of adaptive array algorithm,” *IEEE Trans. Acoust. Speech, SIgnal Processing*, vol. 34, no. 5, pp. 1038–1045, Oct. 1986.

[3] A. W. Bojanczyk and F. T. Luk, “A novel MVDR beamforming algorithm,” *Proc. SPIE, advanced algorithms and architectures for signal processing II*, pp. 12–16, 1987.

[4] J. G. McWhirter and T. J. Shepherd, “Systolic array processor for MVDR beamforming,” *IEE Proceedings*, vol. 136, no. 2, pp. 75–80, April 1989.

[5] B. Yang and J. F. Bome, “Systolic implementation of a general adaptive array processing algorithm,” in *Proc. ICASSP*, April 1988, pp. 2785–2788.

[6] S. Y. Kung, *VLSI array processors*. Prentice Hall, 1998.

[7] S. Haykin, *Adaptive Filter Theory*, 3rd ed. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1996.

[8] A. H. Sayed and T. Kailath, “A state-space approach to adaptive RLS filtering,” *IEEE Signal Processing Magazine*, pp. 18–60, July 1994.