Adaptive Linearly Constrained Inverse QRD-RLS Beamforming Algorithm for Moving Jammers Suppression

Shiunn-Jang Chern and Chung-Yao Chang

***Abstract*—In this paper, a general, linearly constrained (LC) recursive least squares (RLS) array-beamforming algorithm, based on an inverse QR decomposition, is developed for suppressing the moving jammers, efficiently. In fact, by using the inverse QR decomposition-recursive least squares (QRD-RLS) algorithm approach, the lease-squares (LS) weight vector can be computed without back substitution and is suitable to be implemented using the systolic array to achieve fast convergence and good numerical properties. The merits of this new constrained algorithm is verified by evaluating the performance, in terms of the learning curve, to investigate the convergence property and numerical efficiency, and the output signal to interference and noise ratio. We show that our proposed algorithm outperforms the conventional linearly constrained LMS (LCLMS) algorithm, and the one using the fast linear constrained RLS algorithm and its modified version.**

***Index Terms*—Adaptive beamformer, interference and noise ratio, inverse QRD-RLS algorithm, linearly constrained (LC), moving jammers, signal-to-interference-and-noise-ratio (SINR), systolic array.**

I. Introduction

MANY adaptive array beamforming algorithms, based on linearly constraints, have been proposed for suppressing undesired interference [1]–[4]. Moreover, these constrained approaches can be applied to wireless communication systems for multiuser detection [19]. An array beamformer is a processor used in conjunction with an array of sensors to provide a versatile form of spatial filtering. The sensor array collects spatial samples of propagation wave fields, which are processed by the beamformer. In case that the desired signal and interfering signal occupy the same temporal frequency band, the conventional temporal filtering approach cannot be used to separate signal from interference. In fact, the desired signal and jammers usually originate from different spatial locations. This spatial separation can be exploited to separate signals from interference using a spatial filtering at the receiver. Such that, the adaptive array system can be employed to automatically adjust its directional response to null the interferences or jammers and thus, enhances the reception of the desired signal. They are two types of adaptive array structures, *viz.*, broadband array structure and the narrowband array structure. In this paper, the narrowband array structure is considered for moving jammers suppression.

The linearly constrained minimum-variance (LCMV) beamformer is considered to be one of the most popular approaches for suppressing the undesired interference [5], [6]. The Frost's beamforming algorithm [1], can be viewed as an adaptive implementation of the LCMV beamformer. However, under certain circumstances, the conventional Frost's beamforming algorithm may have some problem associated with the performance degradation in multiple jammers environment. To circumvent this drawback, in [3] the adaptive transformed domain normalized LMS algorithm with discrete cosine transform (DCT) and discrete Harley transform (DHT) for broadband adaptive array structure were proposed to improve the nulling capability as well as convergence speed. Also, in [4] the linearly constrained robust fast least square (FLS) beamforming algorithm was suggested to achieve better performance for broadband beamformer.

It is known that, in general, the recursive least squares (RLS) algorithm offers better convergence rate, steady-state means-square error (MSE), and parameter tracking capability over the adaptive leaset mean square (LMS) based algorithm . But, the widespread acceptance of RLS filters has been impeded by a sometime unacceptable numerical performance in limited precision environments. This degradation of performance is especially noticeable for the family of techniques collectively known as "fast" RLS filters [7]–[9]. To overcome this problem, a well known numerical stable RLS algorithm, which is called the QR-decomposition RLS (QRD-RLS) algorithm was proposed in [5], [10], and [11]. Basically, it computes the QR decomposition of the input data matrix using *Givens rotation* and solving the LS weight vector by the back substitution. This, in turn, causes the numerical dynamic range of the transformed computational problem to be reduced. Also, it has the benefit of using the QR-based approaches, that is, the rotation-based computations are easily mapped onto systolic array structures for a parallel implementation with VLSI technology [5], [9]. However, in some practical applications if the least-squares (LS) weight vector is desired in each of iteration then back substitution steps must be performed accordingly . Due to the fact that back substitution is a costly operation to be performed in array structure, in such circumstance the so-called inverse QRD-RLS (IQRD-RLS) [12], [13] algorithm was proposed, where the LS weight vector can be computed without back substitution.

Manuscript received August 31, 2000; revised April 12, 2001. This work was supported by the National Science Council, Taiwan, under Contract NSC 89-2213-E-110-055.

The authors are with the Department of Electrical Engineering, National Sun Yat-Sen University, Kaohsiung 80424, Taiwan, R.O.C.

Publisher Item Identifier 10.1109/TAP.2002.801276.

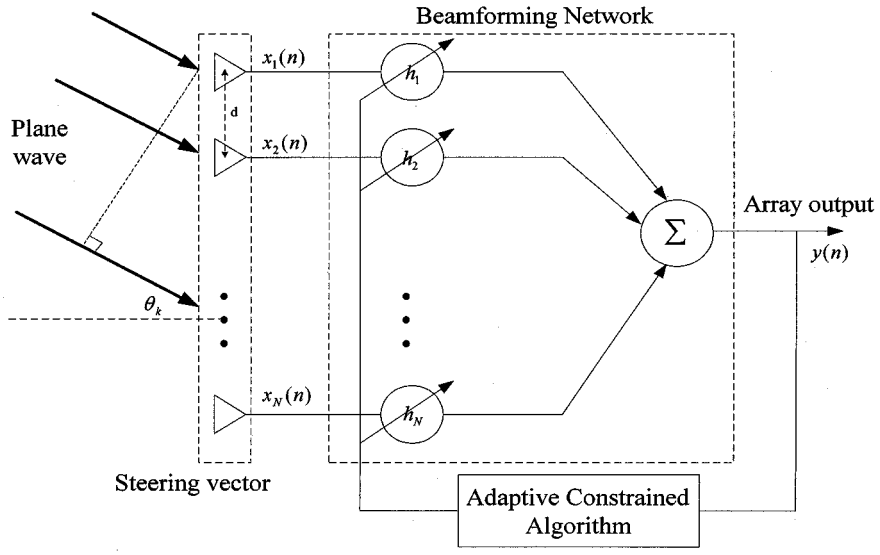

Fig. 1. Configuration of linearly constrained adaptive-array beamformer.

In this paper, a general linearly constrained adaptive beamformer based on the inverse IQRD-RLS algorithm is developed with an adaptive narrowband array structure. It is notice that the proposed IQRD-RLS beamforming algorithm can be viewed as a general formulation or the extension of the one discussed in [17], and is discussed in Appendix B as a reference. The difference of the IQRD-RLS algorithm from the FLS algorithm [4] is that the adaptation gain or Kalman gain is evaluated using *Givens rotation* (QR decomposition) [12], [13]. This results in having better numerical accuracy and stable steady-state MSE and better capability to null multiple undesired interferences than the RLS or FLS algorithm. In this paper, we will first derive the adaptive linearly constrained IQRD-RLS beamforming algorithm and discuss its rationale. After that, computer simulation is carried out to verify the robustness of the method presented with respect to the moving jammers suppression capability.

II. Linearly Constrained Inverse QRD-RLS Beamformer

In this section, the optimal LS solution of the linearly constrained beamformer, based on the IQRD-RLS algorithm [12], [13] is derived with narrowband array structure. Also, its adaptive implementation algorithm is developed and discussed in what follows. First, we will review the formulation of antenna array.

A. Formulation of Antenna Array

The basic operation of the adaptive antenna array is usually described in terms of a receiving system steering a null, that is, a reduction in sensitivity in a certain position, toward a source of interference. It consists of a number of antenna elements coupled together via some form of amplitude control and phase shifting network to form a single output. The amplitude and phase control can be regarded as a set of complex weights, as illustrated in Fig. 1.

To start our derivation, let us consider a uniformly linear array (ULA) and a wavefront, generated by a desired source of wavelength λ, propagating in an N element array of sensors from a direction θ_k off the array boresight. Now, taking the first element in the array as the phase reference and with equal array spacing d, the relative phase shift of the received signal at the nth element can be expressed as

$$\phi_{nk} = \frac{2\pi}{\lambda} d(n-1)\sin\theta_k. \tag{1}$$

Moreover, we assume that the spacing between array elements is set to be $\lambda/2$, the array response vector of this N-antenna ULA can be denoted by

$$\mathbf{a}(\theta_k) = [1, e^{-j\pi\sin(\theta_k)}, \ldots, e^{-j(N-1)\pi\sin(\theta_k)}]^H. \tag{2}$$

Thus, we choose θ_k toward the direction of arrival (DOA) of desired source signal and suitably adjust the weights of adaptive array; the array will pass the desired source signal from direction θ_0 and steer nulls toward interference sources located at θ_k, for $k \neq 0$. It can be shown that an N-element array has $N-1$ degrees of freedom giving up to $N-1$ independent pattern nulls. So it has better performance if the array has more antenna elements.

B. The Optimal Solution of the LC-IQRD-RLS Beamformer

It is known that the principle of a LCMV beamformer [5] is to minimize the powers of background noise and the interference at the linear-array output, while maintaining a chosen frequency response in the look direction. The vector of sampled signals at the time index n is denoted by $\mathbf{x}(n) = [x_1(n), x_2(n), \ldots, x_N(n)]^H$ and the corresponding vector of the weights appearing at each tap is designated as $\mathbf{h}(n) = [h_1(n), h_2(n), \ldots, h_N(n)]^H$. Where the superscript H is denoted as the Hermitian operation and N is the number of array elements. The output signal is given by

$$y(n) = \mathbf{h}^H(n)\mathbf{x}(n). \tag{3}$$

In the method of exponentially weighted least square, we choose the weights at time n, so as to minimize the cost function that consists of the sum of weighted output power

$$\xi(n) = \sum_{i=1}^{n} w^{n-i}|y(i)|^2 = \sum_{i=1}^{n} w^{n-i}|\mathbf{h}^H(n)\mathbf{x}(i)|^2 \quad (4)$$

where parameter, w $(0 < w \leq 1)$, is an exponential weighting factor or forgetting factor that controls the speed of convergence and tracking capability of the algorithm. For convenience, let us denote a $n \times N$ data matrix, in terms of snapshots $\mathbf{x}(1), \mathbf{x}(2), \mathbf{x}(3), \ldots$, which is defined by $\mathbf{X}(n) = [\mathbf{x}(1), \mathbf{x}(2), \ldots, \mathbf{x}(n)]^T$, with $\mathbf{x}(i) = [x_1(i), x_2(i), \ldots, x_N(i)]^T$. In consequence, we may rewrite (4) in a matrix form

$$\xi(n) = \left\| \Lambda^{1/2}(n)\mathbf{X}(n)\mathbf{h}(n) \right\|^2. \quad (5)$$

Also, $\Lambda^{1/2}(n) = \text{diag}\left[\sqrt{w^{n-1}}, \sqrt{w^{n-2}}, \ldots, \sqrt{w}, 1\right]$ is a diagonal matrix. In the linearly constraint problem, the constraints of the weights are introduced by a linear system

$$\mathbf{C}^H\mathbf{h}(n) = \mathbf{f}. \quad (6)$$

In (6), $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K]$ is a $N \times K$ constraint matrix, constructed by array steering vectors, $\mathbf{c}_i = [c_{i1}, c_{i2}, \ldots, c_{iN}]^H$, and $\mathbf{f} = [f_1, f_2, \ldots, f_k]^H$ is the K-element response column vector. Therefore, the constrained optimization problem becomes to minimize the cost function defined in (5), subject to the constraints defined in (6).

It is known that in the conventional QRD-RLS algorithm, an orthogonal matrix $\mathbf{Q}(n)$ is employed to do the triangular factorization of the data matrix, $\Lambda^{1/2}(n)\mathbf{X}(n)$, by Givens rotation

$$\mathbf{Q}(n)\Lambda^{1/2}(n)\mathbf{X}(n) = \begin{bmatrix} \mathbf{R}(n) \\ \mathbf{O} \end{bmatrix} \quad (7)$$

where $\mathbf{R}(n)$ is an $N \times N$ upper triangular matrix, and $\mathbf{O}$ is the $(n-N) \times N$ null matrix. Since orthogonal matrix is length preserving, using the result of (7), the cost function can be rewritten as

$$\xi[\mathbf{h}(n)] = ||[\mathbf{R}(n)\mathbf{h}(n)]||^2 \quad (8)$$

where $||(\bullet)||$ denoted the Euclidean norm of $(\bullet)$. Now, the constrained optimization problem becomes to minimize (8), subject to the constraints defined in (6). Consequently, proceed in a similar way as in [14], we may derive the constrained optimal solution of the LS weight vector, via Lagrange multiplier method based on the *IQRD-RLS* notation, that is

$$\mathbf{h}(n) = \left[\mathbf{R}^H(n)\mathbf{R}(n)\right]^{-1}\mathbf{C}\left\{\mathbf{C}^H\left[\mathbf{R}^H(n)\mathbf{R}(n)\right]^{-1}\mathbf{C}\right\}^{-1}\mathbf{f}. \quad (9)$$

Based on (9), in Section II-C, the recursive implementation of the optimum linearly constrained LS solution, using the IQRD-RLS algorithms, can be developed.

C. Recursive Implementation of LC-IQRD-RLS Algorithm

To derive the recursive equation of (9), we define a new $N \times N$ matrix $\mathbf{S}(n)$

$$\mathbf{S}(n) = \mathbf{R}^{-1}(n)\mathbf{R}^{-H}(n) \quad (10)$$

and its inverse matrix to be

$$\mathbf{S}^{-1}(n) = \mathbf{R}^H(n)\mathbf{R}(n). \quad (11)$$

In fact, it can be easily shown that matrix $\mathbf{S}^{-1}(n)$ is equivalent to the following definition:

$$\mathbf{S}^{-1}(n) = \mathbf{X}^H(n)\Lambda(n)\mathbf{X}(n) = \sum_{i=1}^{n} w^{n-i}\mathbf{x}^*(i)\mathbf{x}^T(i). \quad (12)$$

Also, following the terminology in the Kalman filtering, matrix $\mathbf{S}^{-1}(n)$ is referred to as a "correlation matrix" of the exponential weighted sensor outputs averaged over n snapshots, while its inverse is denoted by $\mathbf{S}(n) = \mathbf{X}^{-1}(n)\Lambda^{-1}(n)\mathbf{X}^{-H}(n)$ whenever it exists. For convenience, we define $\Gamma(n) = \mathbf{S}(n)\mathbf{C}$ and $\Phi(n) = \mathbf{C}^H\Gamma(n) = \mathbf{C}^H\mathbf{S}(n)\mathbf{C}$, in consequence, (9) can be expressed as

$$\mathbf{h}(n) = \Gamma(n)\Phi^{-1}(n)\mathbf{f}. \quad (13)$$

Here (13) can be viewed as the LCMV beamformer based on the inverse QR decomposition.

In what follows, a recursive implementation of (13) will be derived based on the *inverse Cholesky factor* $\mathbf{R}^{-1}(n)$. In fact, in (13) both $\Gamma(n)$ and $\Phi^{-1}(n)$ are related to the *inverse Cholesky factor*, some of the useful parameters and alternative recursive forms of $\Gamma(n)$ and $\Phi^{-1}(n)$ are derived and discussed in Appendix A. First, we recall from (A-1), $\mathbf{T}(n)$ is an unitary matrix, which is used for updating the Cholesky factor $\mathbf{R}(n-1)$ to $\mathbf{R}(n)$, and can be shown to be equivalent to $\mathbf{P}(n)$, where $\mathbf{P}(n)$ is an orthogonal matrix used to update $\mathbf{R}^{-1}(n)$ from $\mathbf{R}^{-1}(n-1)$ [see (A9)]. The key point is to derive a recursive formulation for implementing (13), and the parameters involved in the recursive equation should be related to the scalar parameter, $t(n)$, and vector $\mathbf{g}(n)$ (defined in (A6) and (A7), respectively) while updating the *inverse Cholesky factor* $\mathbf{R}^{-1}(n)$ from $\mathbf{R}^{-1}(n-1)$. To simplify matters, we define the auxiliary matrix

$$\mathbf{A}(n) = \mathbf{R}^{-H}(n)\mathbf{C} \quad (14)$$

and the row vector $\alpha(n) = \mathbf{g}^H(n)\mathbf{C}$, such that $\Phi(n) = \mathbf{A}^H(n)\mathbf{A}(n)$ and $\Gamma(n) = \mathbf{R}^{-1}(n)\mathbf{A}(n)$, respected. From the definition of $\mathbf{A}(n)$, it follows that

$$\mathbf{A}^H(n-1)\mathbf{R}(n-1) = \mathbf{C}^H = \mathbf{A}^H(n)\mathbf{R}(n). \quad (15)$$

In consequence, we may have the following related equation:

$$\begin{bmatrix} \mathbf{A}(n) \\ \alpha(n) \end{bmatrix} = \mathbf{P}(n)\begin{bmatrix} \frac{1}{\sqrt{w}}\mathbf{A}(n-1) \\ 0 \end{bmatrix}. \quad (16)$$

By definition and using (A8), $\Gamma(n)$ and $\Phi(n)$ can be expressed in recursive forms

$$\begin{aligned}\Gamma(n) =&\mathbf{R}^{-1}(n)\mathbf{A}(n) = \mathbf{R}^{-1}(n)\mathbf{R}^{-H}(n)\mathbf{C}\\ =&\left[\frac{1}{w}\mathbf{R}^{-1}(n-1)\mathbf{R}^{-H}(n-1) - \mathbf{g}(n)\mathbf{g}^{H}(n)\right]\mathbf{C}\\ =&w^{-1}\Gamma(n-1) - \mathbf{g}(n)\alpha(n)\end{aligned} \tag{17}$$

and

$$\begin{aligned}\Phi(n) =&\mathbf{A}^{H}(n)\mathbf{A}(n) = \mathbf{C}^{H}\mathbf{R}^{-1}(n)\mathbf{R}^{-H}(n)\mathbf{C}\\ =&\mathbf{C}^{H}\left[\frac{1}{w}\mathbf{R}^{-1}(n-1)\mathbf{R}^{-H}(n-1) - \mathbf{g}(n)\mathbf{g}^{H}(n)\right]\mathbf{C}\\ =&w^{-1}\Phi(n-1) - \alpha^{H}(n)\alpha(n)\end{aligned} \tag{18}$$

or with $\gamma(n) = \sqrt{w}\alpha^{H}(n)$, we have

$$\Phi(n) = w^{-1}[\Phi(n-1) - \gamma(n)\gamma^{H}(n)]. \tag{19}$$

Applying the inversion matrix lemma to (19), we have the recursive equation of matrix $\Phi^{-1}(n)$

$$\Phi^{-1}(n) = w[\mathbf{I} + \sqrt{w}\mathbf{q}(n)\alpha(n)]\Phi^{-1}(n-1) \tag{20}$$

where $\mathbf{q}(n)$ is defined by

$$\mathbf{q}(n) = \frac{\sqrt{w}\Phi^{-1}(n-1)\alpha^{H}(n)}{1 - w\alpha(n)\Phi^{-1}(n-1)\alpha^{H}(n)}. \tag{21}$$

It should be noted that, by definition of (18), $\Phi(n)$, e.g., $\Phi(n) = \mathbf{A}^{H}(n)\mathbf{A}(n)$, and in consequence of $\Phi^{-1}(n)$ defined in (20), both are in a *square root* form, and is well known that it will be in the absence of process noise by using the *Cholesky factorizaton*. Also, based on (20) and (21) we can show

$$\mathbf{q}(n) = w^{-1/2}[\Phi^{-1}(n)\alpha^{H}(n)]. \tag{22}$$

Here, (22) is useful for deriving the recursive implementation of (13).

Finally, to apply the recursive equations defined in (17) and (20)–(13) and after some simplification, we have the recursive implementation of (13) or the adaptive linearly constrained inverse QRD-RLS (LC-IQRD-RLS) beamforming algorithm

$$\begin{aligned}\mathbf{h}(n) = \mathbf{h}(n-1) - w[\mathbf{g}(n) - \sqrt{w}\Gamma(n)\mathbf{q}(n)]\\ \times\alpha(n)\Phi^{-1}(n-1)\mathbf{f}.\end{aligned} \tag{23a}$$

Moreover, with the definition of (A5) and (A7), (23a) can be further simplified

$$\mathbf{h}(n) = \mathbf{h}(n-1) - \rho(n)\varepsilon(n, n-1) \tag{23b}$$

with

$$\rho(n) = \mathbf{k}(n) - \frac{\sqrt{w}}{t(n)}\Gamma(n)\mathbf{q}(n) \tag{24}$$

TABLE I
SUMMARY OF THE ADAPTIVE LC-IQRD-RLS BEAMFORMING ALGORITHM

- Initialization $\mathbf{R}^{-1}(0) = \delta^{-1}\mathbf{I}$, δ =small positive constant
  $\Gamma(0) = \mathbf{R}^{-1}(0)\mathbf{R}^{-H}(0)\mathbf{C}$
  $\mathbf{h}(0) = \Gamma(0)[\mathbf{C}^{H}\Gamma(0)]^{-1}\mathbf{f}$
- For n=1,2,…, do
  1. Compute the intermediate vector $\mathbf{z}(n)$
     $$\mathbf{z}(n) = \frac{\mathbf{R}^{-H}(n-1)\mathbf{x}(n)}{\sqrt{w}}$$
  2. Evaluate the rotations that define $\mathbf{P}(n)$ which annihilates vector $\mathbf{z}(n)$ and compute the scalar variable $t(n)$
     $$\mathbf{P}(n)\begin{bmatrix}\mathbf{z}(n)\\ 1\end{bmatrix} = \begin{bmatrix}\mathbf{o}\\ t(n)\end{bmatrix}$$
  3. Update the lower triangular matrix $\mathbf{R}^{-H}(n)$ and compute the vector $\mathbf{g}(n)$ and $\alpha(n) = \mathbf{g}^{H}(n)\mathbf{C}$
     $$\mathbf{P}(n)\begin{bmatrix}w^{-1/2}\mathbf{R}^{-H}(n-1)\\ \mathbf{o}^{T}\end{bmatrix} = \begin{bmatrix}\mathbf{R}^{-H}(n)\\ \mathbf{g}^{H}(n)\end{bmatrix}$$
- Update following equations and intermediate inverse matrix:
  $$\Gamma(n) = w^{-1}\Gamma(n-1) - \mathbf{g}(n)\alpha(n)$$
  $$\mathbf{q}(n) = \frac{\sqrt{w}\Phi^{-1}(n-1)\alpha^{H}(n)}{1 - w\alpha(n)\Phi^{-1}(n-1)\alpha^{H}(n)}$$
  $$\Phi^{-1}(n) = w[\mathbf{I} + \sqrt{w}\mathbf{q}(n)\alpha(n)]\Phi^{-1}(n-1)$$
- Updating the LS weight vector:
  $$\mathbf{h}(n) = \mathbf{h}(n-1) - \rho(n)\varepsilon(n, n-1)$$
  with
  $$\rho(n) = \mathbf{k}(n) - \frac{\sqrt{w}}{t(n)}\Gamma(n)\mathbf{q}(n)$$
  $$\varepsilon(n, n-1) = \mathbf{x}^{H}(n)\mathbf{h}(n-1)$$

and

$$\varepsilon(n, n-1) = \mathbf{x}^{H}(n)\mathbf{h}(n-1). \tag{25}$$

In (25), $\varepsilon(n, n-1)$ can be viewed as the *a priori* output of beamformer. This completes our derivation for the adaptive linearly constrained inverse QRD-RLS beamforming algorithm. For convenience, a complete procedure of implementing the linearly constrained inverse QRD-RLS (LC-IQRD-RLS) algorithm is summarized in Table I, as reference. It should be noted that in Table I, the adaptation gain $\mathbf{k}(n)$, and the auxiliary vector, $\alpha(n)$, are related to vector $\mathbf{g}(n)$ and scalar variable $t(n)$, which are obtained while we compute the *inverse Cholesky factor* $\mathbf{R}^{-1}(n)$ from $\mathbf{R}^{-1}(n-1)$ via *Givens rotation*.

Since it is known that the QR decomposition or the inverse QR decomposition via *Givens rotation* has more robust stability characteristics, therefore, we can expect that the proposed algorithm will have better numerical stability and accuracy as comparing with the conventional fast RLS family. Moreover, for implementation in Table I the initial value of $\mathbf{R}^{-1}(0)$ is proportional to δ^{-1}, and from [5] the value of δ should be chosen suitably small compared to $0.01\sigma_x^2$, where σ_x^2 is the average power of input data. In our case, in general, the input data will be higher correlated when the jammers are more closely located. In such cases the value of δ has to be chosen slightly larger than the one

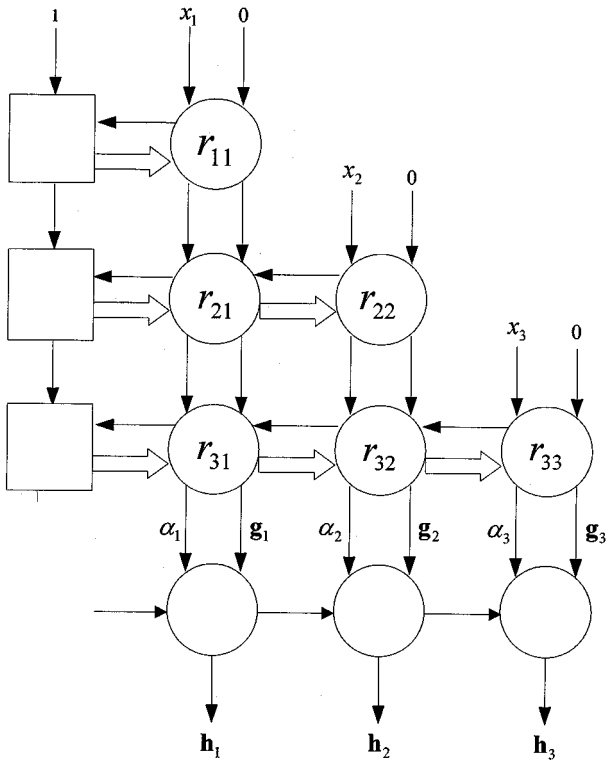

Fig. 2. Systolic array for inverse updating beamforming algorithm.

suggested in [5]. However, for sufficient large data lengths, the exact value of δ will be less significant.

III. The Systolic Array Implementation of LC-IQRD-RLS Algorithm

In the conventional RLS algorithm, calculation of Kalman gain required the inversion of the correlation matrix of input vector $\mathbf{x}(n)$. If the data matrix is in ill condition, or in the worst case has a rank less than the number of the weight vector elements, i.e., N, the conventional RLS algorithm will rapidly become numerically unstable as the inversion of correlation matrix becomes impossible. Moreover, it still has a shortcoming, namely, they generally do not lend themselves to efficient hardware implementation.

It is well known that the QR decomposition has the advantages of parallel execution and the computational power can be reached by connecting general purposed digital signal processing (DSP) device. As described earlier, using the QRD-RLS approach, the optimal weight vector is obtained via back substitution. Unfortunately, pipelining of the two steps (triangular update and back substitution) on a triangular array seems impossible because of its opposite executive direction. However, as depicted in Fig. 2, using the inverse QRD-RLS based method proposed in this paper, this disadvantage can be avoided. It can be mapped onto the systolic array structures for a parallel implementation with VLSI technology. As illustrated in Fig. 2, each processing cell, used for inverse updating the beamforming array, comprises three types of cells, whose functions are described in Fig. 3.

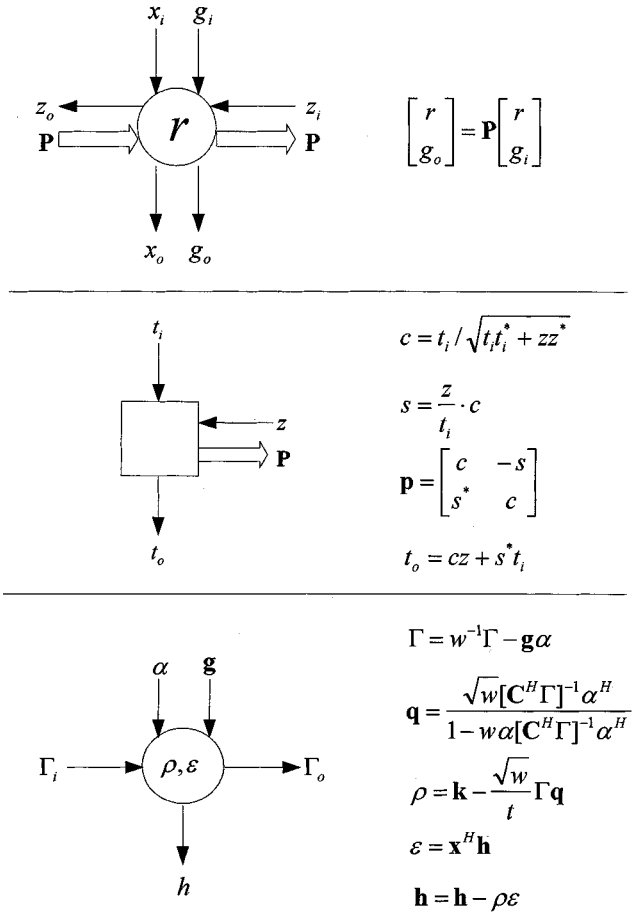

Fig. 3. Processing cells required for the inverse updating the beamforming array.

Basically, this array is made up of three sections: The triangular array in the right-upper part in Fig. 2 stores and updates the inverse Cholesky factor. Each element of the triangular part of the array has an associated internal state r_{ij}, which is equivalent to the corresponding element of the inverse Cholesky factor $\mathbf{R}(n)$. And the square cells in the left-upper part generate the rotation parameters. Finally, the row in the lower part updates the intermediate parameters and computes the weight vector. The systolic array has a latency of N, symbol periods arises from the requirement to clock the data through the processing elements, but provided the model order is not too high, the delay can be tolerated in this application. It is noted that division operator, in the pipelined implementing of the proposed algorithm developed in this paper, is avoided. All the intermediate parameters and inverse matrix have their corresponding recursive form.

An advantage of the systolic array over than other architectures is that the individual processing element computational complexity does not grow with increasing order N. The larger filter orders of implementation proposed in this paper, which is using systolic array, can be accommodated than with contemporary algorithms such as the standard RLS algorithm has a computational complexity of $O(N^2)$. Whenever multiple constraints with steering vectors are considered, it is sufficient to replicate the row in the lower part to update the weight vector. Conversely, only one triangular array is required irrespective of the number of look directions. Also, by using the systolic array

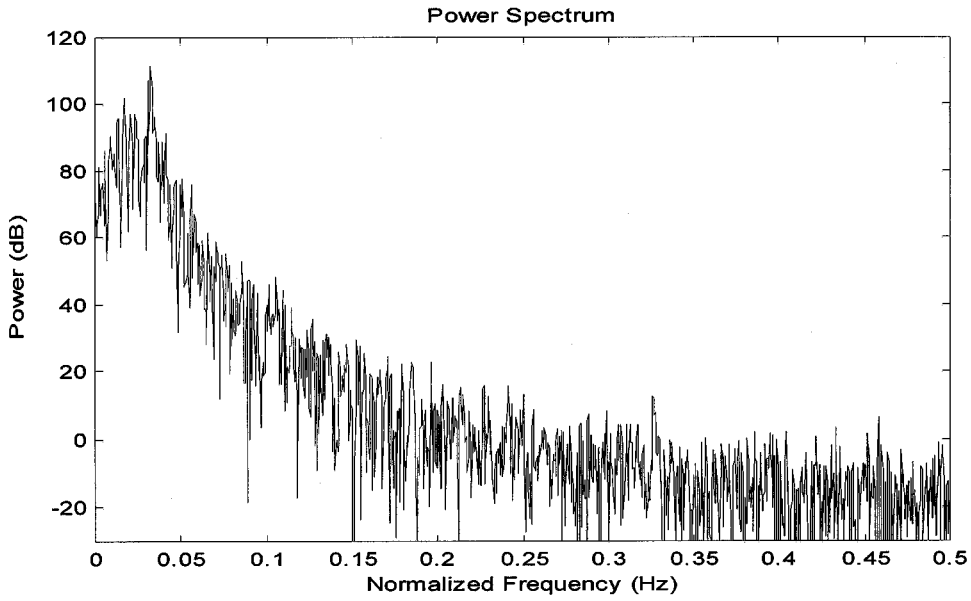

Fig. 4. Power spectrum of look direction desired signal.

structure described in Figs. 2 and 3, the adaptive LC-IQRD-RLS beamforming algorithm can be implemented in real-time application, efficiently.

IV. Computer Simulation Results

In this section, computer simulations are carried out to validate and investigate the performance of the presented method. We assume that the received signal in each sensor consists of a desired source signal buried in white Gaussian noise and three directional interferences (or jammers) incident at angles θ_1, θ_2, and θ_3, respectively. For convenience, the look direction of the desired source signal is chosen to be $\theta = 0°$. Also, the desired source signal is generated using the AR(2) model, that is

$$u(n) + a_1 u(n-1) + a_2 u(n-1) = v(n). \tag{26}$$

Here the coefficients of AR(2) model are set to be $a_1 = -1.9114$, $a_2 = 0.95$ and the variance of white process $v(n)$ is chosen to be $\sigma_v^2 = 0.0038$ to have unit variance of desired source signal $u(n)$, e.g., $\sigma_u^2 = 1$. The power spectrum of desired source signal, from look direction, is illustrated in Fig. 4.

Next, in the following simulations, we assume that the linear array has seven sensors $(N = 7)$ and the analog frequencies of corresponding jammers are chosen to be 600, 700, and 800 Hz, also, the sampling frequency is set to be 2000 Hz. The corresponding incident angles of these three jammers are $-25°$, $45°$, and $50°$. For further discussion, we define JNR_1, JNR_2, JNR_3 as the corresponding jammer-to-noise ratios and assume that $\mathrm{JNR}_1 \leq \mathrm{JNR}_2 = \mathrm{JNR}_3$. To see the effect due to eigenvalue spread two sets of jammer's power ratio (JPR), e.g., the ratio between JNR_1 and $\mathrm{JNR}_2 (= \mathrm{JNR}_3)$, that is, $\mathrm{JNR} = 1$ and 100, are considered.

Since in the adaptive LCLMS beamforming algorithm (or the Frost's algorithm) the upper bound of the step-size to assure the convergence is $2/3Tr[\mathbf{R}_{xx}]$. For convenience, the step size with one-half of the upper bound is selected, and two specific sets of values of the step size are listed in Table II for computer simulation.

TABLE II
The Values of Step Size for Varied JPR

	JPR	Upper bound of μ	μ is chosen
Case1	1	3.1726×10^{-5}	1.5863×10^{-5}
Case2	100	4.7591×10^{-6}	2.3795×10^{-6}

To investigate the performance of the presented method, first, we will examine the capability of interference rejection and the results are compared with the conventional Frost's linear constrained least mean square (LCLMS) beamforming algorithm and the one discussed in [4], using the linearly constrained fast LS (LCFLS) and robust FLS (LCRFLS) beamforming algorithms. In general, since the averaged power of jammer is much larger than the desired source signal, the SNR is set to 0 dB. In the first case, three jammers with equal power, e.g., $\mathrm{JNR}_1 = \mathrm{JNR}_2 = \mathrm{JNR}_3 = 30$ dB, and incident angels, $-25°$, $45°$, and $50°$, are considered. The results, in terms of nulling capability, are given in Fig. 5 for different methods with 200 iterations, which is the average of 500 independent runs. As observed from Fig. 5, the presented method performs superior to the LCFLS and the LCRFLS algorithms, and much better than the conventional Frost's algorithm. In the second case, for $\mathrm{JNR}_1 = 10$ dB, and $\mathrm{JNR}_2 = \mathrm{JNR}_3 = 40$ dB, similar results are observed in Fig. 6. Also, for comparison, the results of these two cases are listed in Table III.

As observed from Table III, the performance using the adaptive LCLMS beamforming algorithm is affected by the eigenvalue spread as in case 2, and the performance becomes worse, while the other algorithms perform quite well. As indicated in [4], in this case, the adaptive LCRFLS beamforming algorithm has 2–8 dB improvement over the adaptive LCFLS beamforming algorithm, due to the effect of introducing a correction term. Although, both adaptive LCFLS and LCRFLS

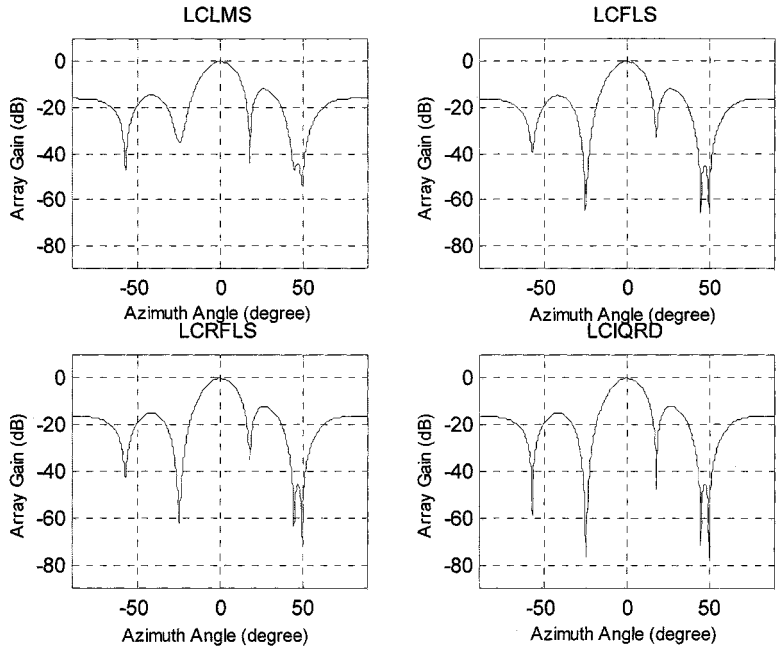

Fig. 5. Beam patterns of case 1 with different algorithms after 200 iterations (500 runs).

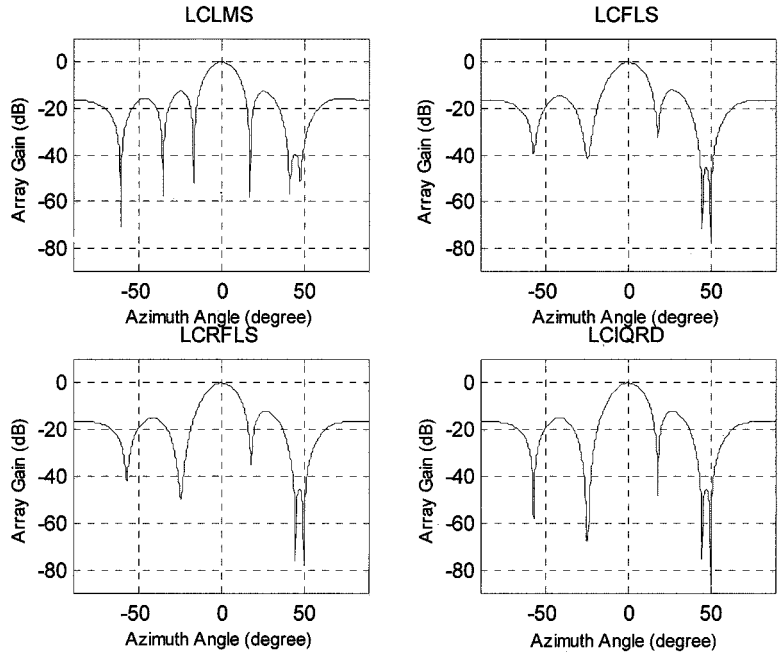

Fig. 6. Beam patterns of case 2 with different algorithms after 200 iterations (500 runs).

beamforming algorithms introduced reasonable deep nulls in the incident angels of 45° and 50°. For smaller power jammer incident from -25°, 20 dB and 13 dB decays are observed, compared with case 1, for the adaptive LCFLS and LCRFLS algorithms, respectively. However, in these two cases, the presented method performs quite well and quite close to the results of MVDR. Therefore, we may conclude that the adaptive LC-IQRD algorithm, proposed in this paper, has the best nulling capability for rejecting jammers and separating adjacent jammers compared to the others. It is of interested to point out that the results described above are obtained using the Pentium III 550 MHz (with 256-MB RAM). However, if PC of Pentium I 166 MMX (with 32 MB) is employed, the adaptive LCFLS algorithm will diverge as indicated in [14].

Next, we would like to examine the convergence property and the numerical stability of the presented method. To see the convergence property, in terms of learning curves, case 2 is considered, and the results are given in Fig. 7, where the minimal MSE is given by $J_{\min} = 0.956 \cong 1$ (0 dB). From Fig. 7, we learn that the adaptive LCLMS algorithm converges slowly approaching the minimal MSE after 3000 iterations. Although both the adaptive LCFLS and LCRFLS algorithms could converge faster, they will diverge after 1500 and 2000 iterations, respectively, due to the instability of updating the Kalman gain during adaptation

TABLE III
COMPARISON OF NULLING CAPABILITY FOR DIFFERENT METHODS

	Jammer's angles	Jammer's power	LCLMS	LCFLS	LCRFLS	LCIQRD	MVDR
Case1	-25^0	30 dB	-35.39 dB	-61.89 dB	-62.44 dB	-77.03 dB	-82.02 dB
	45^0	30 dB	-47.16 dB	-63.42 dB	-63.81 dB	-71.74 dB	-81.80 dB
	50^0	30 dB	-53.55 dB	-65.01 dB	-69.45 dB	-76.92 dB	-83.24 dB
Case2	-25^0	10 dB	-12.59 dB	-41.97 dB	-49.44 dB	-67.86 dB	-72.91 dB
	45^0	40 dB	-39.81 dB	-71.76 dB	-75.43 dB	-76.18 dB	-84.26 dB
	50^0	40 dB	-39.07 dB	-75.55 dB	-76.27 dB	-86.19 dB	-88.57 dB

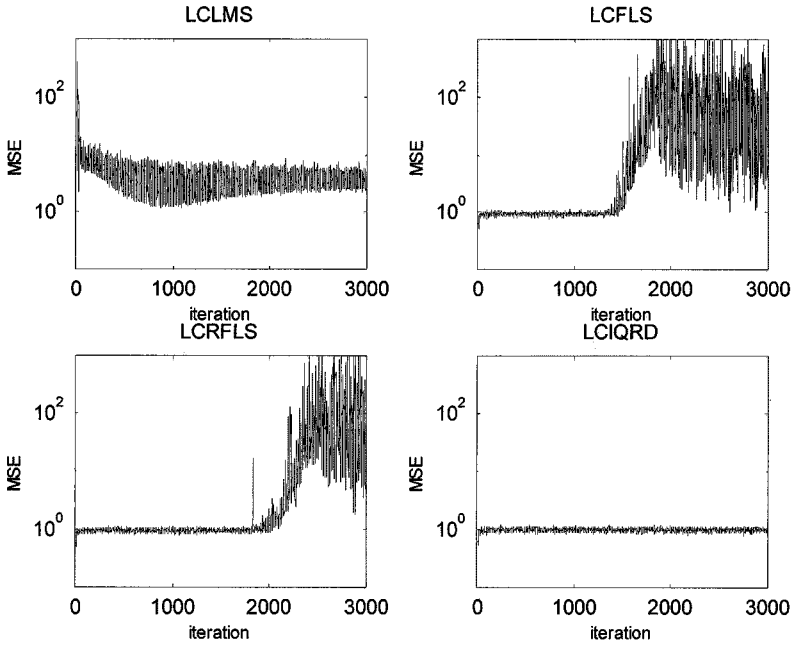

Fig. 7. Learning curves for case 2 after 3000 iterations (using 500 runs).

processes. However, by using the adaptive LC-IQRD algorithm, proposed in this paper, fast convergence and better numerical stability can be achieved.

Finally, it is of interest to investigate the tracking capability due to moving jammers. To do so, we consider the case with three moving jammers, which may occur in wireless communication environment, with their azimuthal trajectories being denoted by

$$\theta_1(n) = -25^\circ - 0.1^\circ n, \quad \theta_2(n) = 50^\circ - 0.05^\circ n$$
$$\theta_3(n) = 45^\circ - 0.01^\circ n \tag{27}$$

where n is the snapshot time index. To be more specific, the value of output signal to interference and noise ratio (SINR) can be evaluated based on the following definition: [5]

$$\text{SINR}_{\text{out}}(n) = \frac{P_s|\mathbf{h}^H(n)\mathbf{s}|^2}{\mathbf{h}^H(n)\mathbf{R}_{\text{in}}(n)\mathbf{h}(n)} \tag{28}$$

where P_s is the averaged power of target signal, $\mathbf{s}$ is the steering vector in the constrained matrix $\mathbf{C}$, and $\mathbf{R}_{\text{in}}(n)$ is the covariance matrix of interferences (jammers) and noise. Also, the optimal SINR is defined by [5]

$$\text{SINR}_{\text{out\_optimal}}(n) = p_s\mathbf{s}^H\mathbf{R}_{\text{in}}^{-1}(n)\mathbf{s}. \tag{29}$$

First, let us examine the results shown in Fig. 8, with the parameters as in case 1 except that SNR $= 20$ dB, when the jammers are still. From Fig. 8, we learn that the proposed adaptive LC-IQRD algorithm outperforms the adaptive LCLMS algorithm and the one using the LC-FLS algorithm and its modified version, e.g., the LC-RFLS algorithm. As expected, both adaptive LC-RFLS and LC-IQRD algorithms could converge to the optimal output SINR, e.g., 19.93 dB, faster. However, it diverges after long-term adaptation due to the instability as indicated in Fig. 7. While the LC-LMS algorithm may not be able to approach the desired value, after 200 snapshots, it could only achieve its best SINR value at about 16.5 dB.

Next, for the case of moving jammers, we set the parameters to be the same as in case 1, and in this case, we have the desired output $\text{SINR}_{\text{out}}(n)$ to be -5.57 dB. As shown in Fig. 9, the conventional adaptive LCLMS algorithm (or Frost's algorithm) could not track the moving jammers, properly. In consequence, the performance of LCLMS algorithm will degrade significantly because of its slow tracking capability. Therefore, we may conclude that the presented method, with the LCIQRD algorithm, is more robust than the existing methods, particularly, when real time implementation is required at high data bandwidths.

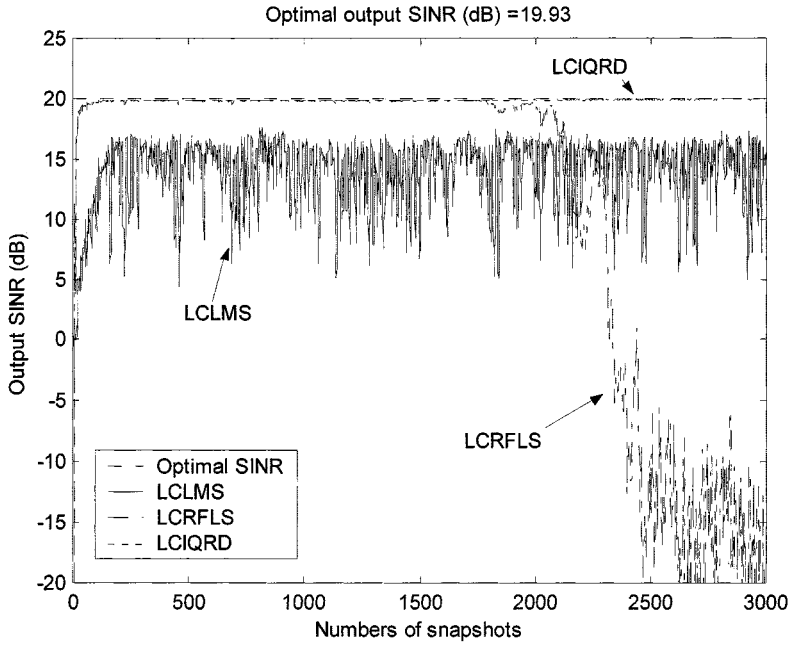

Fig. 8. The output SINR for case 1 with stationary jammers.

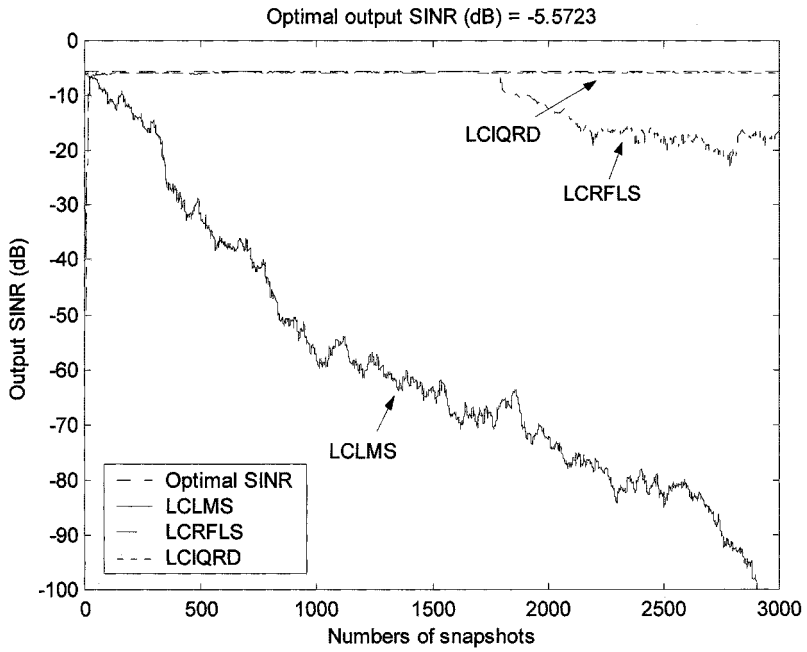

Fig. 9. The output SINR with case 1 for moving jammers.

For convenience, in Table IV the comparison of the proposed algorithm with the conventional Frost's algorithm and the FLS and robust FLS algorithms [4], in terms of numerical stability and computation efficiency, are given. It is known that the family of the "fast " least squares has serious numerical instability in limited precision environment, and has time shifting relationship between the tapped-delay line below each sensor of beamformer. But this is not the case when narrowband beamformer structure is employed. The sensor output is simply weighted and summed to compute the array output. Consequently, it will reduce the computational complexity, significantly. In this case, the Kalman gain of FLS can be evaluated using a conventional RLS procedure with a complexity proportional to N^2. Since in our proposed method, the QR decomposition is employed and is very efficient for evaluating the Kalman gain and the related parameters. Thus, with similar complexity as the LCFLS algorithm, the proposed method has better numerical stability.

V. Conclusion

In this paper, a generalized adaptive linearly constrained beamformer based on the inverse QRD-RLS algorithm has been derived. In fact, it can be viewed as an extention of the

TABLE IV
COMPARISON OF FOUR CONSTRAINED ALGORITHMS IN NARROWBAND BEAMFORMER

Algorithms	LCLMS	LCFLS	LCRFLS	LCIQRD
Converge rate	Slow (data correlation)	Fast	Fast	Fast
Steady state (MSE)	Large (data correlation)	Middle (round off error)	Small (correcting term)	Small
Numerical stability	Stable	Unstable (RLS algorithm)	Unstable (RLS algorithm)	Stable (orthogonal transformation)

TABLE V
COMPLEXITIES COMPARISON OF FOUR ALGORITHMS IN DIFFERENT UPDATE PROCEDURES

Algorithm		Kalman gain	Intermediate constrained matrix updating procedure	Update LS weights	Total complexity
LCLMS	MUL				N^2+2N+1
	ADD				N^2+2N-1
LCFLS	MUL	$6N$	$(2N+2)K^2+(6N+1)K$	$NK+(3N+1)$	$(2N+2)K^2+(7N+1)K+(9N+1)$
	ADD		$2NK^2+(5N-5)K$	$NK+(2N-1)$	$2NK^2+(6N-5)K+(2N-1)$
LCRFLS	MUL	$6N$	$(5N+1)K^2+(3N+1)K$	$2NK+2N$	$(5N+1)K^2+(5N+1)K+8N$
	ADD		$5NK^2+(N-1)K$	$2NK+(2N-1)$	$5NK^2+(3N-1)K+(2N-1)$
LCIQRD	MUL	$6N$	$(2N+2)K^2+(2N+2)K+1$	$NK+(2N+1)$	$(2N+2)K^2+(3N+2)K+(8N+2)$
	ADD		$2NK^2-K+1$	$NK+(2N-1)$	$2NK^2+(N-1)K+2N$

one developed in [17] (see Appendix B). To document the advantage of our proposed method the performance comparison of the learning curves and nulling capability for different methods are evaluated. The computer results have verified the merit of the proposed method. As observed from the results shown in Section IV, the proposed method could be used to achieve the desired value of SINR and having better tracking capability as compared to the conventional LMS and fast RLS algorithms. Also, the performance, in terms of numerical stability and computation efficiency, has been shown to be superior to other conventional algorithms. Moreover, by using the systolic array structures described in Figs. 2 and 3, the adaptive LC-IQRD-RLS beamforming algorithm can be implemented in real time application, efficiently. Thus, we concluded that the overall performance, in terms of computation efficiency, convergence property, and nulling capability of the presented method, did perform over the linearly constrained FLS and its modified version.

Moreover, as discussed in [16], the direct QR algorithm may suffer the accumulative round off error to give rise to numerical problems and the simple back substitution needs $O(N^2)$ operations. On the other hand, it has been shown that in the inverse QRD-RLS algorithm, the recursive updating of the triangular matrix requires only $O(N)$ operations. In [18], a detail comparison of implementing the associated intermediate cells of inverse QRD-RLS algorithm and the fast RLS algorithm is given. Consequently, we can conclude that the inverse QRD algorithm has the advantages of numerical stability and complexities than the QRD algorithm.

TABLE VI
COMPUTATIONAL EXPENSE OF THE EXISTING ALGORITHMS COMPARED WITH THE LC-IQRD-RLS ALGORITHM (100%)

Algorithm	N=7, K=1 (MUL)	Compare with LCIQRD
LCLMS	64	65.98 %
LCFLS	130	134.02 %
LCRFLS	128	131.96 %
LCIQRD	97	100.00 %

To see the merits of the LC-IQRD-RLS algorithm, the computational expense is calculated. The complexities in terms of multiplication and addition operators in each stage of the relative four algorithms are compared. Complexities of three stages are concerned, which included the Kalman gain stage, the intermediate constrained matrix updating stage, and the updating LS weights vector stage. For convenience, the number of multiplication and addition operators in each stage and overall complexities of four algorithms are listed in Table V. Also, to be more specific, the computation expense with $N = 7$ and $K = 1$ is given in Table VI. From Table VI, we learn that the proposed algorithm is more computational efficient than other existing algorithms, except the one using the LCLMS algorithm proposed by Frost [1].

APPENDIX A

In this appendix, some of the equations during the access of updating the inverse Cholesky factor in the inverse QRD-RLS

algorithm that are very useful for deriving the linearly constrained inverse QRD-RLS beamforming algorithm, are discussed. From [12], we know that the upper triangular matrix $\mathbf{R}(n)$ can be updated in a recursive form

$$\begin{bmatrix} \mathbf{R}(n) \\ \mathbf{o}^T \end{bmatrix} = \mathbf{T}(n) \begin{bmatrix} \sqrt{w}\mathbf{R}(n-1) \\ \mathbf{x}^H(n) \end{bmatrix} \tag{A1}$$

where $\mathbf{T}(n)$ is the $(N+1)\times(N+1)$ orthogonal matrix, which annihilates the transpose of the input vector, $\mathbf{x}^H(n)$, by rotating it into $\sqrt{w}\mathbf{R}(n-1)$. Thus, the matrix $\mathbf{T}(n)$ can be formed as the product of N *Givens rotation*. To begin the derivation, first we do the premultiplication on both sides of (A1) with their respective Hermitian, yields

$$[\mathbf{R}^H(n) \quad \mathbf{o}] \begin{bmatrix} \mathbf{R}(n) \\ \mathbf{o}^T \end{bmatrix} = [\sqrt{w}\mathbf{R}^H(n-1) \quad \mathbf{x}(n)]\,\mathbf{T}^H(n) \times \mathbf{T}(n) \begin{bmatrix} \sqrt{w}\mathbf{R}(n-1) \\ \mathbf{x}^H(n) \end{bmatrix}. \tag{A2}$$

Since $\mathbf{T}(n)$ is an orthogonal matrix, (A2) may be expanded to produce

$$\mathbf{R}^H(n)\mathbf{R}(n) = w\mathbf{R}^H(n-1)\mathbf{R}(n-1) + \mathbf{x}(n)\mathbf{x}^H(n). \tag{A3}$$

By using the inversion matrix lemma, we get

$$\mathbf{R}^{-1}(n)\mathbf{R}^{-H}(n) = \frac{1}{w}\mathbf{R}^{-1}(n-1)\mathbf{R}^{-H}(n-1) - \frac{\mathbf{R}^{-1}(n-1)\mathbf{z}(n)\mathbf{z}^H(n)\mathbf{R}^{-H}(n-1)}{wt^2(n)}. \tag{A4}$$

In (A4), the intermediate vector $\mathbf{z}(n)$ is designated as

$$\mathbf{z}(n) = \frac{\mathbf{R}^{-H}(n-1)\mathbf{x}(n)}{\sqrt{w}} \tag{A5}$$

which is the key to parallel implementation of the inverse QRD-RLS approach and the scalar variable $t(n)$ is defined by

$$t(n) = \sqrt{1 + \mathbf{z}^H(n)\mathbf{z}(n)}. \tag{A6}$$

Moreover, we may define a new $N \times 1$ vector $\mathbf{g}(n)$

$$\mathbf{g}(n) = \frac{\mathbf{R}^{-1}(n-1)\mathbf{z}(n)}{\sqrt{w}t(n)}. \tag{A7}$$

Using the definition of (A7), (A4) can be expressed as

$$\mathbf{R}^{-1}(n)\mathbf{R}^{-H}(n) = \frac{1}{w}\mathbf{R}^{-1}(n-1)\mathbf{R}^{-H}(n-1) - \mathbf{g}(n)\mathbf{g}^H(n). \tag{A8}$$

Equation (A8) implies the existence of an $(N+1)\times(N+1)$ orthogonal matrix $\mathbf{P}(n)$ such that

$$\mathbf{P}(n) \begin{bmatrix} w^{-1/2}\mathbf{R}^{-H}(n-1) \\ \mathbf{o}^T \end{bmatrix} = \begin{bmatrix} \mathbf{R}^{-H}(n) \\ \mathbf{g}^H(n) \end{bmatrix}. \tag{A9}$$

It can also be shown that $\mathbf{P}(n)$ is a rotation matrix, which successively annihilates the element of the vector $\mathbf{z}(n)$, starting from the top, by rotating them into the element at the bottom of the augment vector [12]

$$\mathbf{P}(n) \begin{bmatrix} \mathbf{z}(n) \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{o} \\ t(n) \end{bmatrix}. \tag{A10}$$

It should be noted that both $\mathbf{g}(n)$ and $t(n)$ are computed, using the *Givens rotation*, when $\mathbf{R}^{-1}(n)$ is updated from $\mathbf{R}^{-1}(n-1)$. Moreover, it is of interest to point out that the vector $\mathbf{g}(n)$ scaled by $t(n)$ can be viewed as the adaptation or Kalman gain of the inverse QRD-RLS algorithm. To see that we let $\mathbf{k}(n)$ be

$$\mathbf{k}(n) = \frac{\mathbf{g}(n)}{t(n)}. \tag{A11}$$

Substituting $\mathbf{S}(n)$, defined in (10), into (A4) and using the matrix inversion lemma, we can easily show that

$$\mathbf{S}(n) = \frac{1}{w}\mathbf{S}(n-1) - \frac{1}{w}\mathbf{k}(n)\mathbf{x}^H(n)\mathbf{S}(n-1) \tag{A12}$$

with

$$\mathbf{k}(n) = \frac{\frac{1}{w}\mathbf{S}(n-1)\mathbf{x}(n)}{1 + \frac{1}{w}\mathbf{x}^H(n)\mathbf{S}(n-1)\mathbf{x}(n)}. \tag{A13}$$

It is noticed, that matrix $\mathbf{S}(n)$ is in the form of *square root* form, e.g., $\mathbf{S}(n) = \mathbf{R}^{-1}(n)\mathbf{R}^{-H}(n)$, thus will be in absence of process noise by using the *Cholesky factorization*. Consequently, from (A12) and (A13), we can prove

$$\mathbf{k}(n) = \mathbf{S}(n)\mathbf{x}(n). \tag{A14}$$

In fact, (A12) and (A13) have a very similar form as that in the conventional RLS algorithm. Finally, by defining a $N \times K$ matrix

$$\Gamma(n) = \mathbf{S}(n)\mathbf{C} \tag{A15}$$

and after doing the right multiplication on both sides of (A12) by $\mathbf{C}$, we have

$$\Gamma(n) = \frac{1}{w}\Gamma(n-1) - \frac{1}{w}\mathbf{k}(n)\mathbf{x}^H(n)\Gamma(n-1). \tag{A16}$$

By premultiplying both sides of (A15) by $\mathbf{C}^H$ and let $\Phi(n) = \mathbf{C}^H\Gamma(n)$, we have

$$\Phi(n) = \frac{1}{w}\Phi(n-1) - \frac{1}{w}\mathbf{k}(n)\mathbf{x}^H(n)\Phi(n-1). \tag{A17}$$

The Kalman gain, $\mathbf{k}(n)$, $\Gamma(n)$, and $\Phi(n)$ derived in this appendix will be very useful for the development of the constrained linearly inverse QRD-RLS algorithm.

APPENDIX B

In this appendix, to exploit the relationship of the results presented in this paper with the one discussed in [17], we extend the result, based on the formulation addressed in [17], to derive the LS weight vector. It is noted, that in [17] only the residual output signal is required in the minimum variance distortionless response (MVDR) beamformer. It suggested that in the inverse QR update procedure the pretransformation matrix $\mathbf{T}$ for the input data vector is required and it is invertible and well conditioned. Where matrix $\mathbf{T}$ composed by the original constrained matrix $\mathbf{C}$ and the dummy constraints $\mathbf{C}_a$. The best choice for the dummy constraints $\mathbf{C}_a$ is that it should be an orthogonal basis for the null space $\mathbf{C}$, e.g., $\mathbf{C}^H\mathbf{C}_a = \mathbf{0}$. It is noted that

such a "blocking matrix" also appeared in the generalized sidelobe canceller. In [17], the $N \times N$ pretransformation matrix is defined as

$$\mathbf{T} = \begin{bmatrix} \mathbf{C} & \vdots & \mathbf{C}_a \end{bmatrix} = \begin{bmatrix} \mathbf{C} & \mathbf{0} \\ & \mathbf{I} \end{bmatrix} \tag{B1}$$

where $\mathbf{0}$ is a $K \times (N-K)$ zero matrix and $\mathbf{I}$ is a $(N-K) \times (N-K)$ identity matrix. Furthermore, the transformed triangular matrix and input data vector are denoted as

$$\tilde{\mathbf{R}}^{-H}(n) = \mathbf{R}^{-H}(n) \cdot \mathbf{I} \tag{B2}$$

and

$$\tilde{\mathbf{x}}(n) = \mathbf{T}^{-1} \cdot \mathbf{x}(n) \tag{B3}$$

respectively. By using the definition of (B2) and (B3), it can be easily verified that the intermediate vector $\mathbf{z}(n)$ defined in (A5) is consistent with that derived in [17] as follows:

$$\begin{aligned} \mathbf{z}(n) &= \frac{\tilde{\mathbf{R}}^{-H}(n-1)\tilde{\mathbf{x}}(n)}{\sqrt{w}} \\ &= \frac{(\mathbf{R}^{-H}(n-1)\mathbf{T}) \cdot (\mathbf{T}^{-1}\mathbf{x}(n))}{\sqrt{w}} \\ &= \frac{\mathbf{R}^{-H}(n-1)\mathbf{x}(n)}{\sqrt{w}}. \end{aligned} \tag{B4}$$

That is the inverse Cholesky factor updating procedure is the same as that proposed in this paper as listed in Table I. Next, to extend the result of [17] and obtain the constrained LS weight vector, the auxiliary matrix defined in (14) is rewritten as $\mathbf{A}(n) = [\mathbf{a}_1(n), \mathbf{a}_2(n), \ldots, \mathbf{a}_K(n)]$ with $\mathbf{a}_i(n) = [a_{i1}(n), a_{i2}(n), \ldots, a_{iN}(n)^T$. In consequence, we may rewrite (9) as follows to obtain the LS weights vector in the MVDR beamformer

$$\mathbf{h}(n) = \frac{\mathbf{R}^{-1}(n)\mathbf{R}^{-H}(n)\mathbf{C}}{\mathbf{C}^H\mathbf{R}^{-1}(n)\mathbf{R}^{-H}(n)\mathbf{C}}\mathbf{f} = \frac{\mathbf{R}^{-1}(n)\mathbf{A}(n)}{\mathbf{A}^H(n)\mathbf{A}(n)}\mathbf{f}. \tag{B5}$$

If we consider K independent constraints and MVDR response, i.e., $f_i = 1$, we have

$$\mathbf{h}_i(n) = \frac{\mathbf{R}^{-1}(n)\mathbf{a}_i(n)}{\|\mathbf{a}_i(n)\|^2} f_i = \frac{\mathbf{R}^{-1}(n)\mathbf{a}_i(n)}{\|\mathbf{a}_i(n)\|^2}, \quad \text{for } i = 1, 2, \ldots, K. \tag{B6}$$

The complete MVDR beamformer algorithm similar to Moonen's expression is given in Table VII.

By comparing the results addressed above with our proposed method, we could say that the one derived in this paper is a more general LCMV formulation, based on the inverse QR updating procedure, which can be applied to the LCMV spatial beamformer but is not limited to the case of scalar MVDR response ($f_i = 1$). That is, it can be widely applied to more general LCMV problems (in time-domain or space-domain), while the MVDR beamformer algorithm, derived in [6] and [17], is very suitable for those that require only the *posteriori* residual signal, i.e., $\varepsilon(n) = \mathbf{x}^H(n)\mathbf{h}(n)$. To be more specific, the smart antenna and space-time signal processing in the code-division multiple-access (CDMA) communication systems are the most significant applications. For instance, we need to estimate the direction-of-arrival (DOA) and adjust the weights of the adaptive beamformer simultaneously, in smart antenna application. Also, in the space–time signal processing, the combined structure composed by beamformer and multiuser detector with interference cancellation have to be considered at the same time. In such applications, the explicit LS weights vector is required and becomes more significant. However, in the MVDR beamformer problem addressed in [17], only the residual output signal is required, the LC-IQRD-RLS algorithm developed in this paper will be reduced to the result of [17].

TABLE VII
MVDR BEAMFORMER ALGORITHM BASED ON INVERSE QR UPDATING [17]

- Initialization $\mathbf{R}^{-1}(0) = \delta^{-1}\mathbf{I}$, δ =small positive constant

$$\tilde{\mathbf{R}}^{-H}(0) = \mathbf{R}^{-H}(0) \cdot \mathbf{T}$$

- For n=1,2,…, do

1. Pre-transformation for input data vector
$$\tilde{\mathbf{x}}(n) = \mathbf{T}^{-1} \cdot \mathbf{x}(n)$$
2. Compute the intermediate vector $\mathbf{z}(n)$
$$\mathbf{z}(n) = \frac{\tilde{\mathbf{R}}^{-H}(n-1)\tilde{\mathbf{x}}(n)}{\sqrt{w}} = \frac{\mathbf{R}^{-H}(n-1)\mathbf{x}(n)}{\sqrt{w}}$$
3. Evaluate the rotations that define $\mathbf{P}(n)$ which annihilates vector $\mathbf{z}(n)$ and compute the scalar variable $t(n)$
$$\mathbf{P}(n)\begin{bmatrix} \mathbf{z}(n) \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{o} \\ t(n) \end{bmatrix}$$
4. Update the lower triangular matrix $\mathbf{R}^{-H}(n)$ and compute the auxiliary matrix $\mathbf{A}(n) = \mathbf{R}^{-H}(n)\mathbf{C}$
$$\mathbf{P}(n)\begin{bmatrix} w^{-1/2}\mathbf{R}^{-H}(n-1) \\ \mathbf{o}^T \end{bmatrix} = \begin{bmatrix} \mathbf{R}^{-H}(n) \\ \mathbf{g}^H(n) \end{bmatrix}$$

- Compute LS weights vector, for $i = 1, \ldots, K$

$$\mathbf{h}_i(n) = \frac{\mathbf{R}^{-1}(n)\mathbf{a}_i(n)}{\|\mathbf{a}_i(n)\|^2}$$

REFERENCES

[1] O. L. Frost III, "An algorithm for linearly constraint adaptive array processing," *Proc. IEEE*, vol. 60, pp. 926–935, Aug. 1972.

[2] L. J. Griffths and J. W. Jim, "An alternative approach to linearly constraint adaptive beamforming," *IEEE Trans. Antennas Propagat.*, vol. AP-30, pp. 27–34, Jan. 1982.

[3] S. J. Chern and C. Y. Sung, "The performance of hybrid adaptive beamforming algorithm for jammers suppression," *IEEE Trans. Antennas Propagat.*, vol. 42, pp. 1223–1232, Sept. 1994.

[4] L. S. Resende, J. M. T. Romano, and M. G. Bellanger, "A robust FLS algorithm for LCMV adaptive broad band beamformer," *Proc. IEEE Int. Conf. Acoustics, Speed, and Signal Processing*, vol. 3, pp. 1826–1829, 1996.

[5] S. Haykin, *Adaptive Filter Theory*, 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[6] J. G. McWhirter and T. J. Shepherd, "Systolic array processor for MVDR beamforming," *Proc. Inst. Elect. Eng.* , pt. F, vol. 136, pp. 75–80, Apr. 1989.

[7] J. M. Cioffi and T. Kailath, "Fast RLS transversal filters for adaptive filtering," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 304–337, June 1984.

[8] J. M. Cioffi, "Limited precision effects for adaptive filtering," *IEEE Trans. Circuits Syst.*, vol. CAS-34, pp. 821–833, July 1987.

[9] P. Fabre and C. Gueguen, "Improvement of fast recursive least squares algorithms via normalization: A comparative study," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 296–308, Apr. 1986.
[10] H. Leung and S. Haykin, "Stability of recursive QRD-RLS algorithm using finite precision systolic array implementation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 760–763, May 1989.
[11] P. A. Regalia and M. G. Bellanger, "On the duality between fast QR methods and lattice methods in least squares adaptive filtering," *IEEE Trans. Signal Processing*, vol. 39, pp. 879–891, Apr. 1991.
[12] S. T. Alexander and A. L. Ghirnikar, "A method for recursive least squares filtering based upon an inverse QR decomposition," *IEEE Trans. Signal Processing*, vol. 41, pp. 20–30, Jan. 1993.
[13] C. T. Pan and R. J. Plemmons, "Least squares modifications with factorization: Parallel implications," *J. Comput. Applied Math.*, vol. 27, no. 1-2, pp. 109–127, Arp. 1989.
[14] L. S. Resende, J. T. Romano, and M. G. Bellanger, "A fast least-squares algorithm for linearly constrained adaptive filtering," *IEEE Trans. Signal Processing*, vol. 44, pp. 1168–1174, May 1996.
[15] B. D. Van and K. M. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE Acoust., Speech, Signal Processing Mag.*, vol. 5, pp. 4–24, Apr. 1988.
[16] M. Moonen, "Systolic MVDR beamforming with inverse updating," *Proc. Inst. Elect. Eng.* , pt. Pt. F, vol. 140, pp. 175–178, 1993.
[17] M. Moonen and I. K. Proudler, "MVDR beamforming and generalized sidelobe cancellation based on inverse updating with residual extraction," *IEEE Trans. Circuit Syst. II*, vol. 47, pp. 352–358, Apr. 2000.
[18] J. M. Cioffi, "High speed systolic implementation of fast QR adaptive filters," in *Proc. IEEE Int. Conf. Acoutics, Speed, and Signal Processing*, New York, Apr. 1988, pp. 1584–1587.
[19] J. B. Schodorf and D. W. Williams, "Array processing techniques for multiuser detection," *IEEE Trans. Commun.*, vol. 45, pp. 1375–1378, Nov. 1997.

Shiunn-Jang Chern received the B.S. degree from Tamkang University, Taipei, Taiwan, R.O.C., the M.S. degree from Southeastern Massachusetts University, North Dartmouth, MA, and the Ph.D. degree from McMaster University, Hamilton, ON, Canada, in 1977, 1982, and 1986, respectively.

He is currently a Professor in the Electrical Engineering Department, National Sun Yet-Sen University, Kaohsiung, Taiwan, R.O.C. His research interests include statistical analysis of an adaptive filtering algorithm, time-delay estimation, array signal processing, and wireless communication systems.

Chung-Yao Chang was born in Taiwan, R.O.C., on July 22, 1977. He received the B.S. degree in electrical engineering from the National Sun Yat-Sen University, Kaohsiung, Taiwan, R.O.C., in 1999. He is currently working toward the Ph.D. degree at the same university .

His research interests include adaptive filtering algorithm, array signal processing, and CDMA communication systems.