Chapter 12
On Linearly Constrained QRD-Based Algorithms

Shiunn-Jang Chern

Abstract The linearly constrained adaptive filtering (LCAF) technique has been extensively used in many engineering applications. In this chapter, we introduce the linearly constrained minimum variance (LCMV) filter, implemented using the linearly constrained recursive least squares (RLS) criterion, with the inverse QR decomposition (IQRD) approach. First, the direct form of recursively updating the constrained weight vector of LS solution based on the IQRD is developed, which is named as the LC-IQRD-RLS algorithm. With the IQRD approach, the parameters related to the *Kalman gain* are evaluated via *Givens* rotations and the LS weight vector can be computed without back-substitution. This algorithm is suitable to be implemented using systolic arrays with very large scale integration technology and DSP devices. For the sake of simplification, an alternative indirect approach, referred to as the generalized sidelobe canceler (GSC), is adopted for implementing the LCAF problem. The GSC structure essentially decomposes the adaptive weight vector into constrained and unconstrained components. The unconstrained component can then be freely adjusted to meet any criterion since the constrained component will always ensure that the constraint equations are satisfied. The indirect implementation could attain the same performance as that using the direct constrained approach and possesses better numerical properties. Via computer simulation, the merits of the LC-IQRD-RLS algorithms over the conventional LC-RLS algorithm and its modified version are verified.

12.1 Introduction

The linearly constrained (LC) adaptive filtering (LCAF) technique is known to have many applications in areas such as minimum variance spectral analysis, time delay estimation, and antenna array signal processing [1–5]. More recently, these

Shiunn-Jang Chern
National Sun Yat-Sen University, Kaohsiung City, Taiwan – R.O.C.
e-mail: chern@ee.nsysu.edu.tw

J.A. Apolinário Jr. (ed.), *QRD-RLS Adaptive Filtering*,
DOI 10.1007/978-0-387-09734-3_12, © Springer Science+Business Media, LLC 2009

constrained approaches have been applied to wireless communication systems for multiuser detection [6, 7]. Among the adaptive filtering algorithms, in most practical applications, the RLS algorithm has shown to offer better convergence rate and steady-state mean-squared error (MSE) over the least mean squares (LMS)-based algorithms. Unfortunately, the widespread acceptance of the RLS filter has been refrained due to numerical instability problems when it is implemented in limited-precision environments. Performance degradation is especially noticeable for the family of techniques collectively known as “fast” RLS filters [8–10]. To circumvent this problem, a well-known numerical stable RLS algorithm, which is called the QR-decomposition RLS (QRD-RLS) algorithm was proposed [8, 11–13]. It computes the QR decomposition (triangular orthogonalization) of the input data matrix using Givens rotation and then solves the LS weight vector by means of the back-substitution procedure.

The QRD-RLS algorithm can be implemented using the systolic array [14–17] with very large scale integration (VLSI) technology and DSP devices. However, in some practical applications, if the LS weight vector is desired in each iteration, back-substitution steps must be performed accordingly. Due to the fact that back-substitution is a costly operation to be performed in an array structure, the so-called inverse QRD-RLS (IQRD-RLS) algorithm proposed in [18] is preferred, for the LS weight vector can be computed without implementing back-substitution.

In this chapter, we first introduce the LC-RLS filtering algorithm based on an IQRD, where a direct form of recursively updating the constrained weight vector of LS solution is developed. The basic approach of the LC-IQRD-RLS algorithm is very similar to that discussed in [2]. That is, based on the Kalman gain of the conventional IQRD-RLS algorithm, the LC-IQRD-RLS algorithm can be developed where the unconstrained form of the weight vector and the *a priori* estimation error can be avoided. In the IQRD-RLS algorithm, the parameters related to the Kalman gain are evaluated using the *Givens rotations* (QR decomposition), which is quite different from the one discussed in [2] (using the fast RLS algorithm), yielding different development. Usually, the IQRD-RLS algorithm has better numerical accuracy than the “fast” RLS algorithm. Indeed, the numerical instability may cause the constraint drift problem [19] for the constrained approach based on the conventional fast RLS algorithm, named the constrained fast LS (CFLS) algorithm [2].

In this chapter, an alternative indirect approach, referred to as the generalized side-lobe canceler (GSC), is employed [6, 8, 20, 21] for various applications. The GSC structure essentially decomposes the adaptive weight vector into constrained and unconstrained components. The unconstrained component can then be freely adjusted to meet any criterion since the constrained component will always ensure that the constraint equations are satisfied. The GSC-based algorithm could attain the same performance as the direct constrained approach and possesses better numerical properties.

In [22], the authors have proved that the optimal linearly constrained solution derived with the direct and indirect (GSC structure) structures are equivalent for the conjugate gradient and LMS-based algorithms, if the blocking matrix satisfies the following condition, e.g., $\mathbf{B}^{\mathrm{H}}\mathbf{B} = \mathbf{I}$. Furthermore, in [21] a more general proof of the equivalency of direct and GSC structures for the LMS-based algorithms was given.

This chapter is organized as follows. First, in Section 12.2, we derive the optimal linearly constrained LS weight vector solution, based on the IQRD, and discuss the rationale behind it. After that, in Section 12.3, the LC-IQRD-RLS algorithm is developed and applied to the linearly constrained minimum variance (LCMV) filtering problems to achieve the desired performance. In Section 12.4, an alternative indirect approach using the GSC structure is developed. To document the merits of the proposed algorithm, in Section 12.5, two applications with computer simulations results are given to show the efficiency in terms of fast convergence and numerical stability of the LC-IQRD-RLS and the GSC-IQRD-RLS algorithms over the constrained fast RLS (CFLS) algorithm. Finally, we give a conclusion in Section 12.6 to summarize this chapter.

Notations and used symbols: Vectors and matrices are denoted by boldface lower and upper case letters. All vectors are column vectors. $(\cdot)^{-1}$, $(\cdot)^{*}$, $(\cdot)^{\mathrm{T}}$, and $(\cdot)^{\mathrm{H}}$ denote inverse, complex conjugate, transpose, and conjugate transpose, respectively. $\|\cdot\|$ denotes Frobenius norm and $\mathbf{I}_N$ is the $N \times N$ identity matrix. Null matrix or vector is denoted by $\mathbf{0}$ with corresponding dimension.

12.2 Optimal Linearly Constrained QRD-LS Filter

As an introduction, we consider the configuration of the LCAF as depicted in Figure 12.1.[1] Here, $\mathbf{x}(k) = [x(k), x(k-1), \ldots, x(k-N)]^{\mathrm{T}}$ denotes the vector of sampled input signals at the time index k and the weight vector is defined as $\mathbf{w}(k) = [w_0(k), w_1(k), \ldots, w_N(k)]^{\mathrm{T}}$. In this chapter, we focus on the LCMV filtering problem, which uses the blind optimization approach, with the exponentially weighted least-squares (LS) method, the cost function is defined as

$$J_{LS}(\mathbf{w}) = \sum_{i=0}^{k} \lambda^{k-i} |y(i)|^2 = \sum_{i=0}^{k} \lambda^{k-i} |\mathbf{w}^{\mathrm{H}}\mathbf{x}(i)|^2. \tag{12.1}$$

The LS solution of $\mathbf{w}(k)$ is obtained by minimizing (12.1) with respect to $\mathbf{w}$, subject to multiple constraints. In (12.1), the parameter λ $(0 \ll \lambda \leq 1)$ is the forgetting factor that controls the speed of convergence and tracking capability of the algorithm. For convenience, we rewrite (12.1) in a matrix form, i.e.,

$$J_{LS}(\mathbf{w}) = \|\boldsymbol{\Lambda}^{1/2}(k)\mathbf{y}(k)\|^2 = \|\boldsymbol{\Lambda}^{1/2}(k)\mathbf{X}(k)\mathbf{w}\|^2 \tag{12.2}$$

[1] Note that the same algorithms developed in this chapter can also be employed for the case of a constrained linear combiner, when the input signal vector does not correspond to a delay line.

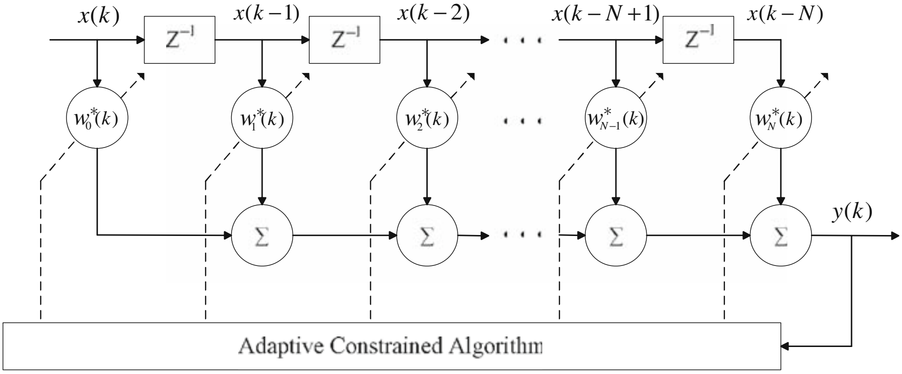

Fig. 12.1 The structure of a linearly constrained FIR filter.

where $\mathbf{y}(k) = [y(0), y(1), ...y(k)]^{\mathrm{T}}$ is denoted as the output vector. Also, $\boldsymbol{\Lambda}^{1/2}(k) = diag[\sqrt{\lambda^k}, \sqrt{\lambda^{k-1}}, \cdots, \sqrt{\lambda}, 1]$ is a diagonal matrix and $\mathbf{X}(k)$ is the $(k+1)\times(N+1)$ data matrix denoted by $\mathbf{X}(k) = [\mathbf{x}(0), \mathbf{x}(1), ..., \mathbf{x}(k)]^{\mathrm{H}}$. It can be noticed that the definition of the data matrix $\mathbf{X}(k)$ in this chapter is slightly different from the definition found in (2.14), Chapter 2; however, this difference in the way the input data matrix is defined results in equivalent algorithms. In the conventional QRD-RLS algorithm, an orthogonal matrix $\mathbf{Q}(k)$ of order $(k+1)\times(k+1)$, is used to perform the triangular orthogonalization of the data matrix $\boldsymbol{\Lambda}^{1/2}(k)\mathbf{X}(k)$ by means of *Givens rotations* [8, 12], that is,

$$\mathbf{Q}(k)\boldsymbol{\Lambda}^{1/2}(k)\mathbf{X}(k) = \begin{bmatrix} \mathbf{U}(k) \\ \mathbf{O} \end{bmatrix}, \tag{12.3}$$

where $\mathbf{U}(k)$ is an $(N+1)\times(N+1)$ upper triangular matrix, and $\mathbf{O}$ is a $(k-N)\times(N+1)$ null matrix.

We note that, in order to be consistent with the definition of the data matrix $\mathbf{X}(k)$, the definition of $\mathbf{U}(k)$ is also slightly different from that given in introductory chapters. Also, the *Givens rotation* is known to have the ability to be implemented in parallel and systolic structure. In consequence, (12.2) can be rewritten as

$$J_{LS}(\mathbf{w}) = \|\mathbf{U}(k)\mathbf{w}\|^2 . \tag{12.4}$$

For the linearly constrained optimization, (12.4) is minimized subjects to L linear constraints, e.g., $\mathbf{C}^{\mathrm{H}}\mathbf{w} = \mathbf{f}$, where the $(N+1)\times L$ constraint matrix is denoted as $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_L]$ and $\mathbf{f} = [f_1, f_2, ..., f_L]^{\mathrm{T}}$ is the L-element response column vector. Proceeding in a similar way as in [2], via Lagrange multipliers with the QR-decomposition approach, the linearly constrained LS solution is given by [1]

$$\mathbf{w}(k) = \left[\mathbf{U}^{\mathrm{H}}(k)\mathbf{U}(k)\right]^{-1}\mathbf{C}\left\{\mathbf{C}^{\mathrm{H}}\left[\mathbf{U}^{\mathrm{H}}(k)\mathbf{U}(k)\right]^{-1}\mathbf{C}\right\}^{-1}\mathbf{f}\,. \tag{12.5}$$

Based on the optimal constrained LS solution of (12.5), in next section we will develop the recursive form of (12.5), named the LC-IQRD-RLS algorithm.

12.3 The Adaptive LC-IQRD-RLS Filtering Algorithm

To derive the recursive form of (12.5) for using in a LCAF, as depicted in Figure 12.2, we define a new $(N+1)\times(N+1)$ matrix $\mathbf{S}(k)$, i.e.,

$$\mathbf{S}(k) = [\mathbf{U}^{\mathrm{H}}(k)\mathbf{U}(k)]^{-1}. \tag{12.6}$$

Its inverse can be easily shown to be equivalent to the following definition:

$$\begin{aligned}\mathbf{S}^{-1}(k) &= \mathbf{U}^{\mathrm{H}}(k)\mathbf{U}(k) \\ &= \mathbf{X}^{\mathrm{H}}(k)\boldsymbol{\Lambda}(k)\mathbf{X}(k) \\ &= \sum_{i=1}^{k}\lambda^{k-i}\mathbf{x}(i)\mathbf{x}^{\mathrm{H}}(i).\end{aligned} \tag{12.7}$$

For convenience, we define $\boldsymbol{\Gamma}(k) = \mathbf{S}(k)\mathbf{C}$ and $\boldsymbol{\Phi}(k) = \mathbf{C}^{\mathrm{H}}\boldsymbol{\Gamma}(k) = \mathbf{C}^{\mathrm{H}}\mathbf{S}(k)\mathbf{C}$; as a consequence, (12.5) can be expressed as

$$\mathbf{w}(k) = \boldsymbol{\Gamma}(k)\boldsymbol{\Phi}^{-1}(k)\mathbf{f}. \tag{12.8}$$

We may view (12.8) as the LCMV weight vector solution implemented by the LS approach with the IQRD. In what follows, based on the inverse *Cholesky factor* $\mathbf{U}^{-1}(k)$, the recursive form of (12.8) is developed. Also, some of the useful parameters and alternative recursive equations of $\boldsymbol{\Gamma}(k)$ and $\boldsymbol{\Phi}^{-1}(k)$, which are related to the inverse *Cholesky factor*, are derived. Recalling from [11], the upper triangular matrix of $\mathbf{U}(k)$ as defined in (12.3) can be written in a recursive form

$$\begin{bmatrix}\mathbf{U}(k)\\ \mathbf{0}^{\mathrm{T}}\end{bmatrix} = \mathbf{T}(k)\begin{bmatrix}\sqrt{\lambda}\mathbf{U}(k-1)\\ \mathbf{x}^{\mathrm{H}}(k)\end{bmatrix}, \tag{12.9}$$

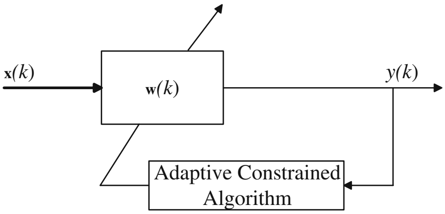

Fig. 12.2 The block diagram of the adaptive linearly constrained filter.

where $\mathbf{T}(k)$ is the $(N+2)\times(N+2)$ orthogonal matrix, which annihilates the Hermitian of the input vector, $\mathbf{x}^{\mathrm{H}}(k)$, by rotating it into $\sqrt{\lambda}\mathbf{U}(k-1)$. Thus, matrix $\mathbf{T}(k)$ can be formed as the product of $(N+1)$ *Givens rotations*. By multiplying both sides of (12.9) with their respective Hermitian on the left, it gives

$$\mathbf{U}^{\mathrm{H}}(k)\mathbf{U}(k) = \lambda\mathbf{U}^{\mathrm{H}}(k-1)\mathbf{U}(k-1) + \mathbf{x}(k)\mathbf{x}^{\mathrm{H}}(k). \tag{12.10}$$

By using the matrix inversion lemma in (12.10), and after some mathematical manipulation, we obtain

$$\mathbf{U}^{-1}(k)\mathbf{U}^{-\mathrm{H}}(k) = \frac{1}{\lambda}\mathbf{U}^{-1}(k-1)\mathbf{U}^{-\mathrm{H}}(k-1) - \mathbf{g}(k)\mathbf{g}^{\mathrm{H}}(k), \tag{12.11}$$

where the intermediate vector, $\mathbf{g}(k)$ is defined by

$$\mathbf{g}(k) = \frac{\mathbf{U}^{-1}(k-1)\mathbf{z}(k)}{\sqrt{\lambda}t(k)}. \tag{12.12}$$

The scalar variable $t(k)$ and intermediate vector $\mathbf{z}(k)$ of (12.12) are defined, respectively, by $t(k) = \sqrt{1+\mathbf{z}^{\mathrm{H}}(k)\mathbf{z}(k)}$ and

$$\mathbf{z}(k) = \frac{\mathbf{U}^{-\mathrm{H}}(k-1)\mathbf{x}(k)}{\sqrt{\lambda}}. \tag{12.13}$$

Equation (12.11) implies the existence of an $(N+2)\times(N+2)$ orthogonal matrix $\mathbf{P}(k)$, which annihilates vector $\mathbf{z}(k)$, starting from the top, by rotating them into the element at the bottom of the augment vector $\left[\mathbf{z}^{\mathrm{T}}(k)\ 1\right]^{\mathrm{T}}$ [1], which is given by

$$\mathbf{P}(k)\begin{bmatrix}\mathbf{z}(k)\\ 1\end{bmatrix} = \begin{bmatrix}\mathbf{0}\\ t(k)\end{bmatrix}. \tag{12.14}$$

While updating the lower triangular matrix $\mathbf{U}^{-\mathrm{H}}(k)$ from $\mathbf{U}^{-\mathrm{H}}(k-1)$, with the rotation matrix $\mathbf{P}(k)$, we obtain the intermediate vector $\mathbf{g}^{\mathrm{H}}(k)$, i.e.,

$$\mathbf{P}(k)\begin{bmatrix}\lambda^{-1/2}\mathbf{U}^{-\mathrm{H}}(k-1)\\ \mathbf{0}^{\mathrm{T}}\end{bmatrix} = \begin{bmatrix}\mathbf{U}^{-\mathrm{H}}(k)\\ \mathbf{g}^{\mathrm{H}}(k)\end{bmatrix}. \tag{12.15}$$

We note that both $\mathbf{g}(k)$ and $\mathbf{z}(k)$ just described are computed using the same set of *Givens rotations*, when $\mathbf{U}^{-\mathrm{H}}(k)$ is updated from $\mathbf{U}^{-\mathrm{H}}(k-1)$. It is of interest to point out that vector $\mathbf{g}(k)$ scaled by $t(k)$, i.e.,

$$\mathbf{k}(k) = \frac{\mathbf{g}(k)}{t(k)}, \tag{12.16}$$

can be viewed as the *adaptation* or *Kalman gain* of the IQRD-RLS algorithm. Substituting (12.6) into (12.11), with definition (12.12) and (12.13), we can easily show that

$$\mathbf{S}(k) = \lambda^{-1}[\mathbf{S}(k-1) - \mathbf{k}(k)\mathbf{x}^{\mathrm{H}}(k)\mathbf{S}(k-1)], \tag{12.17}$$

with

$$\mathbf{k}(k) = \frac{\lambda^{-1}\mathbf{S}(k-1)\mathbf{x}(k)}{1+\lambda^{-1}\mathbf{x}^{\mathrm{H}}(k)\mathbf{S}(k-1)\mathbf{x}(k)}. \tag{12.18}$$

With the results of (12.17) and (12.18), we can prove that $\mathbf{k}(k) = \mathbf{S}(k)\mathbf{x}(k)$. Now, with the recursive equation of (12.17), the $(N+1)\times K$ matrix $\boldsymbol{\Gamma}(k) = \mathbf{S}(k)\mathbf{C}$ can be rewritten in a recursive form by doing the right multiplication on both sides of (12.17) by $\mathbf{C}$, i.e.

$$\begin{aligned}\boldsymbol{\Gamma}(k) &= \lambda^{-1}[\boldsymbol{\Gamma}(k-1) - \mathbf{k}(k)\mathbf{x}^{\mathrm{H}}(k)\boldsymbol{\Gamma}(k-1)] \\ &= \lambda^{-1}\boldsymbol{\Gamma}(k-1) - \mathbf{g}(k)\boldsymbol{\alpha}(k),\end{aligned} \tag{12.19}$$

where $\boldsymbol{\alpha}(k) = \mathbf{g}^{\mathrm{H}}(k)\mathbf{C}$. The recursive equation of $\boldsymbol{\Phi}(k) = \mathbf{C}^{\mathrm{H}}\boldsymbol{\Gamma}(k)$, previously defined when introducing (12.8), can be expressed as

$$\boldsymbol{\Phi}(k) = \lambda^{-1}[\boldsymbol{\Phi}(k-1) - \mathbf{k}(k)\mathbf{x}^{\mathrm{H}}(k)\boldsymbol{\Phi}(k-1)]. \tag{12.20}$$

Applying the matrix inversion lemma to (12.20), we have

$$\boldsymbol{\Phi}^{-1}(k) = \lambda[\mathbf{I} + \sqrt{\lambda}\mathbf{q}(k)\boldsymbol{\alpha}(k)]\boldsymbol{\Phi}^{-1}(k-1), \tag{12.21}$$

where $\mathbf{q}(k)$ is defined as

$$\mathbf{q}(k) = \frac{\sqrt{\lambda}\boldsymbol{\Phi}^{-1}(k-1)\boldsymbol{\alpha}^{\mathrm{H}}(k)}{1-\lambda\boldsymbol{\alpha}(k)\boldsymbol{\Phi}^{-1}(k-1)\boldsymbol{\alpha}^{\mathrm{H}}(k)}. \tag{12.22}$$

Based on (12.20) and (12.21), we can show that

$$\mathbf{q}(k) = \lambda^{-1/2}[\boldsymbol{\Phi}^{-1}(k)\boldsymbol{\alpha}(k)]. \tag{12.23}$$

Indeed, (12.23) is very useful for deriving the recursive form of (12.5). Finally, by applying the recursive equations defined in (12.19) and (12.21) to (12.8) and after simplification, we have the recursive implementation of (12.5), named the LC-IQRD-RLS algorithm

$$\mathbf{w}(k) = \mathbf{w}(k-1) - \lambda[\mathbf{g}(k) - \sqrt{\lambda}\boldsymbol{\Gamma}(k)\mathbf{q}(k)]\boldsymbol{\alpha}(k)\boldsymbol{\Phi}^{-1}(k-1)\mathbf{f}. \tag{12.24}$$

With the definition of (12.12) and (12.13), (12.24) can be further simplified as

$$\mathbf{w}(k) = \mathbf{w}(k-1) - \boldsymbol{\rho}(k)e(k), \tag{12.25}$$

where

$$\boldsymbol{\rho}(k) = \mathbf{k}(k) - \frac{\sqrt{\lambda}}{t(k)}\boldsymbol{\Gamma}(k)\mathbf{q}(k), \text{ and} \tag{12.26}$$

$$e(k) = \mathbf{w}^{\mathrm{H}}(k-1)\mathbf{x}(k). \tag{12.27}$$

In (12.27), $e(k)$ can be viewed as the *a priori* output of the LCMV filter. This completes our derivation for the adaptive LC-IQRD-RLS algorithm, which is summarized in Table 12.1 for reference. For simplification, an alternative indirect approach, within the GSC framework, of the optimal linearly constrained LS solution, based on the IQRD-RLS algorithm, is developed in the next section.

Table 12.1 Summary of the adaptive LC-IQRD-RLS algorithm.

LC-IQRD-RLS

- Initialization (δ=small positive constant):

$$\begin{aligned}
\mathbf{U}^{-1}(0) &= \delta^{-1}\mathbf{I} \\
\boldsymbol{\Gamma}(0) &= \mathbf{U}^{-1}(0)\mathbf{U}^{-\mathrm{H}}(0)\mathbf{C} \\
\boldsymbol{\Phi}^{-1}(0) &= [\mathbf{C}^{\mathrm{H}}\boldsymbol{\Gamma}(0)]^{-1} \\
\mathbf{w}(0) &= \boldsymbol{\Gamma}(0)[\mathbf{C}^{\mathrm{H}}\boldsymbol{\Gamma}(0)]^{-1}\mathbf{f}
\end{aligned}$$

- For k=1,2,..., do

1. Compute the intermediate vector $\mathbf{z}(k)$:

$$\mathbf{z}(k) = \frac{\mathbf{U}^{-\mathrm{H}}(k-1)\mathbf{x}(k)}{\sqrt{\lambda}}$$

2. Evaluate the rotations that define $\mathbf{P}(k)$ which annihilates vector $\mathbf{z}(k)$ and compute the scalar variable $t(k)$:

$$\mathbf{P}(k)\begin{bmatrix}\mathbf{z}(k) \\ 1\end{bmatrix} = \begin{bmatrix}\mathbf{0} \\ t(k)\end{bmatrix}$$

3. Update the lower triangular matrix $\mathbf{U}^{-\mathrm{H}}(k)$ and compute vectors $\mathbf{g}(k)$ and $\boldsymbol{\alpha}(k)$:

$$\mathbf{P}(k)\begin{bmatrix}\lambda^{-1/2}\mathbf{U}^{-\mathrm{H}}(k-1) \\ \mathbf{0}^{\mathrm{T}}\end{bmatrix} = \begin{bmatrix}\mathbf{U}^{-\mathrm{H}}(k) \\ \mathbf{g}^{\mathrm{H}}(k)\end{bmatrix}$$

$$\boldsymbol{\alpha}(k) = \mathbf{g}^{\mathrm{H}}(k)\mathbf{C}$$

4. Update the following equations and intermediate inverse matrix:

$$\begin{aligned}
\boldsymbol{\Gamma}(k) &= \lambda^{-1}\boldsymbol{\Gamma}(k-1) - \mathbf{g}(k)\boldsymbol{\alpha}(k) \\
\mathbf{q}(k) &= \frac{\sqrt{\lambda}\boldsymbol{\Phi}^{-1}(k-1)\boldsymbol{\alpha}^{\mathrm{H}}(k)}{1-\lambda\boldsymbol{\alpha}(k)\boldsymbol{\Phi}^{-1}(k)\boldsymbol{\alpha}^{\mathrm{H}}(k)} \\
\boldsymbol{\Phi}^{-1}(k) &= \lambda\left[\mathbf{I}+\sqrt{\lambda}\mathbf{q}(k)\boldsymbol{\alpha}(k)\right]\boldsymbol{\Phi}^{-1}(k-1)
\end{aligned}$$

5. Update the LS weight vector:

$$\begin{aligned}
e(k) &= \mathbf{w}^{\mathrm{H}}(k-1)\mathbf{x}(k) \\
\boldsymbol{\rho}(k) &= \mathbf{k}(k) - \frac{\sqrt{\lambda}}{t(k)}\boldsymbol{\Gamma}(k)\mathbf{q}(k) \\
\mathbf{w}(k) &= \mathbf{w}(k-1) - \boldsymbol{\rho}(k)e^{*}(k)
\end{aligned}$$

12.4 The Adaptive GSC-IQRD-RLS Algorithm

The overall weight vector with the GSC [26] structure illustrated in Figure 12.3 is equivalent to that of Figure 12.2. With the GSC structure, an alternative indirect approach of the optimal constrained LS weight vector can be developed. First, the original weight vector of Figure 12.2 is decomposed into two parts, i.e.,

$$\mathbf{w}(k) = \mathbf{w}_c - \mathbf{B}\mathbf{w}_a(k) \; . \tag{12.28}$$

In (12.28), the weight vector of the upper path, $\mathbf{w}_c$, is referred to as *quiescent vector*, while the $(N+1)\times(N+1-L)$ matrix $\mathbf{B}$ of the lower path is the rank reduction or blocking matrix. Indeed, $\mathbf{B}$ could be any matrix, whose columns span the left null space of $\mathbf{C}$, e.g., $\mathbf{B}$ is full rank and satisfies

$$\mathbf{B}^{\mathrm{H}}\mathbf{C} = \mathbf{0} \quad \text{and} \quad \mathbf{C}^{\mathrm{H}}\mathbf{B} = \mathbf{0} \; . \tag{12.29}$$

Therefore, the columns of $\mathbf{B}$ form a basis for the null space of $\mathbf{C}^{\mathrm{H}}$ and $(N+1)\times(N+1-L)$ matrix $\mathbf{B}$ can be obtained from $\mathbf{C}$ by any orthogonalization procedures. On the other hand, the upper path vector $\mathbf{w}_c$ simply ensures that the constraint equations are satisfied. The overall system function with the GSC structure depicted in Figure 12.3, ideally, should be equivalent to the direct approach of Figure 12.2; therefore, substituting (12.28) into the definition of constraints, $\mathbf{C}^{\mathrm{H}}\mathbf{w}(k) = \mathbf{f}$, yields

$$\mathbf{C}^{\mathrm{H}}\mathbf{w}(k) = \mathbf{C}^{\mathrm{H}}\left[\mathbf{w}_c - \mathbf{B}\mathbf{w}_a(k)\right] = \mathbf{C}^{\mathrm{H}}\mathbf{w}_c - \mathbf{C}^{\mathrm{H}}\mathbf{B}\mathbf{w}_a(k) = \mathbf{f} \; . \tag{12.30}$$

From (12.30), it can be easily shown that

$$\mathbf{w}_c = \mathbf{C}(\mathbf{C}^{\mathrm{H}}\mathbf{C})^{-1}\mathbf{f} = \mathbf{F} \; . \tag{12.31}$$

Based on the above discussion, we learn that the general mathematical framework for the GSC structure relies on the unconstrained optimization. Clearly the $(N+1-L)\times 1$ weight vector, $\mathbf{w}_a(k)$, is unconstrained and can be freely adapted using any optimization criterion, while the overall weight vector will remain constrained.

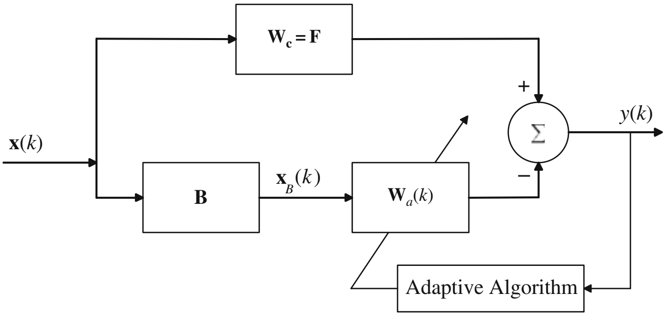

Fig. 12.3 The block diagram of the adaptive linearly constrained filter with the GSC structure.

With the GSC structure, the optimization problem becomes to choose the adaptive weight $\mathbf{w}_a(k)$ from any $\mathbf{w}_a$ in lower branch, to cancel jamming (interfering) signals in upper branch after computing $\mathbf{w}_c$ and $\mathbf{B}$. Under the condition described above, the cost function of (12.1) to be minimized, with the GSC structure, can be rewritten as

$$\begin{aligned} J_{LS}(\mathbf{w}_a) &= \sum_{i=0}^{k} \lambda^{k-i} |[\mathbf{w}_c - \mathbf{B}\mathbf{w}_a]^{\mathrm{H}} \mathbf{x}(i)|^2 \\ &= \sum_{i=0}^{k} \lambda^{k-i} |\mathbf{w}_c^{\mathrm{H}} \mathbf{x}(i) - [\mathbf{B}\mathbf{w}_a]^{\mathrm{H}} \mathbf{x}(i)|^2 \,. \end{aligned} \tag{12.32}$$

If we let $d(i)$ be the desired component of Figure 12.3, e.g., $d(i) = \mathbf{w}_c^{\mathrm{H}}\mathbf{x}(i)$, (12.32) can be represented as

$$\begin{aligned} J_{LS}(\mathbf{w}_a) &= \sum_{i=0}^{k} \lambda^{k-i} |[d(i) - \mathbf{w}_a^{\mathrm{H}} \mathbf{B}^{\mathrm{H}} \mathbf{x}(i)|^2 \\ &= \sum_{i=0}^{k} \lambda^{k-i} |e_B(i/k)|^2 \; = \|\boldsymbol{\Lambda}^{1/2}(k)\mathbf{e}_B(k/k)\|, \end{aligned} \tag{12.33}$$

where the error vector is designated by $\mathbf{e}_B(k/k) = [e_B(0/k), e_B(1/k), ..., e_B(k/k)]^{\mathrm{T}}$. Correspondingly, if we denote $\mathbf{d}(k) = [d(0), d(1), ..., d(k)]^{\mathrm{T}}$ as a desired signal vector, (12.33) becomes

$$J_{LS}(\mathbf{w}) = \|\boldsymbol{\Lambda}^{1/2}[\mathbf{d}^*(k) - \mathbf{X}_B(k)\mathbf{w}_a]\|^2. \tag{12.34}$$

Accordingly, the data matrix based on the structure of GSC is given by $\mathbf{X}_B(k) = [\mathbf{x}_B(0), \mathbf{x}_B(1), \cdots, \mathbf{x}_B(k)]^{\mathrm{H}}$, and $\mathbf{x}_B(i) = \mathbf{B}^{\mathrm{T}}\mathbf{x}(i)$. With the approach of the conventional QRD-RLS algorithm [8, 12, 18], an orthogonal matrix $\mathbf{Q}_B(k)$, is used to carry out the triangular orthogonalization of the data matrix, $\boldsymbol{\Lambda}^{1/2}(k)\mathbf{X}_B(k)$, via *Givens rotations*, that is,

$$\mathbf{Q}_B(k)\boldsymbol{\Lambda}^{1/2}(k)\mathbf{X}_B(k) = \begin{bmatrix} \mathbf{U}_B(k) \\ \mathbf{0} \end{bmatrix}, \tag{12.35}$$

where $\mathbf{U}_B(k)$ is an $(N+1-L) \times (N+1-L)$ upper triangular matrix, and $\mathbf{0}$ is a $(k-N+L) \times (N+1-L)$ null matrix. Similarly, to perform the orthogonal matrix, $\mathbf{Q}_B(k)$, on the weighted desired vector, $\boldsymbol{\Lambda}^{1/2}(k)\mathbf{d}(k)$, it gives

$$\mathbf{Q}_B(k)\boldsymbol{\Lambda}^{1/2}(k)\mathbf{d}^*(k) = \begin{bmatrix} \mathbf{z}_B(k) \\ \mathbf{v}_B(k) \end{bmatrix} \,, \tag{12.36}$$

where vectors $\mathbf{z}_B(k)$ and $\mathbf{v}_B(k)$ are with the dimensions of $(N+1-L) \times 1$ and $(k-N+L) \times 1$, respectively. Since orthogonal matrices are length preserving, using the results of (12.12) and (12.13), the cost function defined in (12.11) can be expressed as

$$J_{LS}(\mathbf{w}_a) = \left\| \begin{bmatrix} \mathbf{z}_B(k) - \mathbf{U}_B(k)\mathbf{w}_a \\ \mathbf{v}_B(k) \end{bmatrix} \right\|^2 \,. \tag{12.37}$$

It is straightforward to see that the norm in Equation (12.37) is minimized if the top partition of (12.37) is set to null. From (12.37), the optimum LS weight vector $\mathbf{w}_a(k)$ based on the QR decomposition can be obtained

$$\mathbf{U}_B(k)\mathbf{w}_a(k) = \mathbf{z}_B(k). \tag{12.38}$$

Similarly, the LS weight vector $\mathbf{w}_a(k)$ based on the IQRD can be represented by

$$\mathbf{w}_a(k) = \mathbf{U}_B^{-1}(k)\mathbf{z}_B(k). \tag{12.39}$$

As a consequence, the IQRD-RLS algorithm for updating the LS weight vector, $\mathbf{w}_a(k)$, is given by [18]

$$\mathbf{w}_a(k) = \mathbf{w}_a(k-1) + \frac{\mathbf{g}_B(k)}{t_B(k)} e_B^*(k), \tag{12.40}$$

with the *a priori* estimation error, $e_B(k)$ being defined by

$$e_B(k) = d(k) - \mathbf{w}_a^{\mathrm{H}}(k-1)\mathbf{x}_B(k). \tag{12.41}$$

The scalar variable $t_B(k)$ and the $(N+1)\times 1$ intermediate vector $\mathbf{g}_B(k)$ are defined as

$$t_B(k) = \sqrt{1+\mathbf{z}_B^{\mathrm{H}}(k)\mathbf{z}_B(k)}, \tag{12.42}$$

and

$$\mathbf{g}_B(k) = \frac{\mathbf{U}_B^{-1}(k-1)\mathbf{z}_B(k)}{\sqrt{\lambda}t_B(k)}, \tag{12.43}$$

respectively, and the intermediate vector $\mathbf{z}_B(k)$, which provides the key to the parallelization of the IQRD-RLS approach, is designated by

$$\mathbf{z}_B(k) = \frac{\mathbf{U}_B^{-\mathrm{H}}(k-1)\mathbf{x}_B(k)}{\sqrt{\lambda}}. \tag{12.44}$$

It should be noted that both $\mathbf{g}_B(k)$ and $t_B(k)$ are evaluated entirely by rotation-based method, using *Givens rotations*, when $\mathbf{U}_B^{-1}(k)$ is updated from $\mathbf{U}_B^{-1}(k-1)$. From [18], it is known that there exists a rotation matrix $\mathbf{P}'(k)$ such that

$$\mathbf{P}'(k)\begin{bmatrix}\mathbf{z}_B(k)\\ 1\end{bmatrix} = \begin{bmatrix}\mathbf{o}\\ t_B(k)\end{bmatrix}, \tag{12.45}$$

where $\mathbf{P}'(k)$ successively annihilates the elements of the vector $\mathbf{z}_B(k)$, starting from the top, by rotating them into the element at the bottom of the augmented vector $[\mathbf{z}_B^{\mathrm{T}}(k), 1]^{\mathrm{T}}$. In consequence, we can evaluate $\mathbf{g}_B(k)$ as follows:

$$\mathbf{P}'(k)\begin{bmatrix}\lambda^{-1/2}\mathbf{U}_B^{-\mathrm{H}}(k-1)\\ \mathbf{0}^{\mathrm{T}}\end{bmatrix} = \begin{bmatrix}\mathbf{U}_B^{-\mathrm{H}}(k)\\ \mathbf{g}_B^{\mathrm{H}}(k)\end{bmatrix}. \tag{12.46}$$

It is of interest to point out that the scalar variable $t_B(k)$ and the intermediate vector $\mathbf{g}_B(k)$ are evaluated based on (12.45) and (12.46). Also, $\mathbf{g}_B(k)$ scaled by $t_B(k)$ can be viewed as the adaptation gain of the IQRD-RLS algorithm, which is defined as

$$\mathbf{k}_B(k) = \frac{\mathbf{g}_B(k)}{t_B(k)}. \tag{12.47}$$

Equation (12.40) tells us that the LS weight vector is updated by incrementing its old value by an amount equal to the *a priori* estimation error, $e_B(k/k-1)$ times the time-varying gain vector, $\mathbf{k}_B(k)$. Moreover, it can be shown that $\mathbf{k}_B(k) = \mathbf{S}_B(k)\mathbf{x}_B(k)$ with $\mathbf{S}_B(k) = \mathbf{U}_B^{-1}(k)\mathbf{U}_B^{-\mathrm{H}}(k)$. This completes the derivation of the GSC-IQRD-RLS algorithm which is summarized in Table 12.2. We note that, for this case where the

Table 12.2 Summary of the adaptive GSC-based IQRD-RLS algorithm.

GSC-IQRD-RLS

- Initialization δ=small positive constant:

$$\begin{aligned} \mathbf{U}_B^{-1}(0) &= \delta^{-1}\mathbf{I} \\ \mathbf{w}_c &= [\mathbf{C}(\mathbf{C}^{\mathrm{H}}\mathbf{C})^{-1}]\mathbf{f} \\ \mathbf{w}_a(0) &= \mathbf{0} \end{aligned}$$

- For k = 1, 2, ... , do

1. Compute the intermediate desired signal $d(k)$ and input vector $\mathbf{x}_B(k)$:

$$\begin{aligned} d(k) &= \mathbf{w}_c^{\mathrm{H}}\mathbf{x}(k) \\ \mathbf{x}_B(k) &= \mathbf{B}^{\mathrm{H}}\mathbf{x}(k) \end{aligned}$$

2. Compute the intermediate vector $\mathbf{z}_B(k)$:

$$\mathbf{z}_B(k) = \frac{\mathbf{U}_B^{-\mathrm{H}}(k-1)\mathbf{x}_B(k)}{\sqrt{\lambda}}$$

3. Evaluate the rotations that define $\mathbf{P}'(k)$ which annihilates vector $\mathbf{z}_B(k)$ and compute the scalar variable $t_B(k)$:

$$\mathbf{P}'(k)\begin{bmatrix} \mathbf{z}_B(k) \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ t_B(k) \end{bmatrix}$$

4. Update the lower triangular matrix $\mathbf{U}_B^{-\mathrm{H}}(k)$ and compute vector $\mathbf{g}_B(k)$:

$$\mathbf{P}'(k)\begin{bmatrix} \lambda^{-1/2}\mathbf{U}_B^{-\mathrm{H}}(k-1) \\ \mathbf{0}^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} \mathbf{U}_B^{-\mathrm{H}}(k) \\ \mathbf{g}_B^{\mathrm{H}}(k) \end{bmatrix}$$

5. Updating the LS weight vector:

$$\begin{aligned} e_B(k/k-1) &= d(k) - \mathbf{w}_a^{\mathrm{H}}(k-1)\mathbf{x}_B(k) \\ \mathbf{w}_a(k) &= \mathbf{w}_a(k-1) + \frac{\mathbf{g}_B(k)}{t_B(k)}e_B^*(k/k-1) \end{aligned}$$

adaptive part within the GSC structure is unconstrained, it is immediate to find a direct correspondence between the variables used here and those of the IQRD-RLS algorithm presented in Chapter 3.

12.5 Applications

Before proceeding further to conclude the advantages of the adaptive LC-IQRD-RLS and GSC-IQRD-RLS algorithms, it is instructive to develop an appreciation for the versatility of these important algorithms by applying them to LCMV filtering problems.

12.5.1 Application 1: Adaptive LCMV filtering for spectrum estimation

In the first application, we consider the spectral analysis which is very significant in many signal processing applications, such as speech signal processing and interference suppression. In this application, we would like to investigate the nulling capability of the LC-IQRD-RLS algorithm and compare it to the general constrained fast LS algorithm. As described in [2], with the general fast LS algorithm (see [2], Table 1), desired performance may not be satisfied due to round-off error during the adaptation processes. Common to adaptive filter parameter updating algorithm, a correction term, proportional to the quantity $\mathbf{f} - Cb\ f^{\mathrm{H}}\mathbf{w}(k)$, is an intuitively reasonable form to obtain a more robust modified version, referred to as the robust constrained FLS (RCFLS) algorithm. To do so, the frequency response of the LCMV filtering, for eliminating the undesired frequencies, is examined. We assume that input signal consists of three sinusoids buried in additive white noise, i.e.,

$$\mathbf{x}(k) = 10\sin(0.3k\pi) + 100\sin(0.66k\pi) + \sin(0.7k\pi) + b(k). \tag{12.48}$$

The corresponding normalized frequencies and amplitudes are set to be 0.15, 0.33, and 0.35, and 10, 100, and 1, respectively, and $b(k)$ denotes the additive white noise with zero-mean and variance, σ_b^2 such that the signal-to-noise ratio (SNR) is 40 dB.

Moreover, the filter is constrained to have unit response at two frequencies, viz., 0.1 and 0.25 (normalized digital frequencies). Each of the two unit response frequencies generates two-point constraints. In such case, $L = 4$ and $(N+1) = 11$ (weight coefficients), the constraint parameters are

$$\mathbf{C}^{\mathrm{T}} = \begin{bmatrix} 1 & \cos(0.2\pi) & \cdots & \cos((N)0.2\pi) \\ 1 & \cos(0.5\pi) & \cdots & \cos((N)0.5\pi) \\ 0 & \sin(0.2\pi) & \cdots & \sin((N)0.2\pi) \\ 0 & \sin(0.5\pi) & \cdots & \sin((N)0.5\pi) \end{bmatrix}_{4\times 11} \quad \text{and } \mathbf{f}^{\mathrm{T}} = \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}. \tag{12.49}$$

We know that, usually, it is more difficult to separate two signals closely in adjacent frequency band, especially when one has relatively larger power than the other; the signal with smaller power may be ignored and yields the wrong result.

From Figure 12.4, we observe that the learning curve of the RCFLS algorithm might be disturbed by the inversion of the correlation matrix (while computing the adaptation gain) which is inherently numerically unstable. The nulling capability for the undesired signal, the one with normalized frequency 0.35 and less power, is affected by the adjacent signal with frequency 0.33 and having relatively larger power. However, the LC-IQRD-RLS algorithm has faster convergence rate and better numerical stability than the RCFLS algorithm. For comparison, the nulling gains, in dB, for different frequency components using the LC-IQRD-RLS algorithm and the RCFLS algorithm, are listed in Table 12.3.

Next, let us consider the problem of constrained drift which may be defined as the squared norm of $\mathbf{C}^T\mathbf{w}(k) - \mathbf{f}$. Also, using the notation of functions found in Matlab®, the tolerance (Tol) of the numerical accuracy may be expressed as

$$Tol = max(size(A)) \times norm(A) \times eps,$$

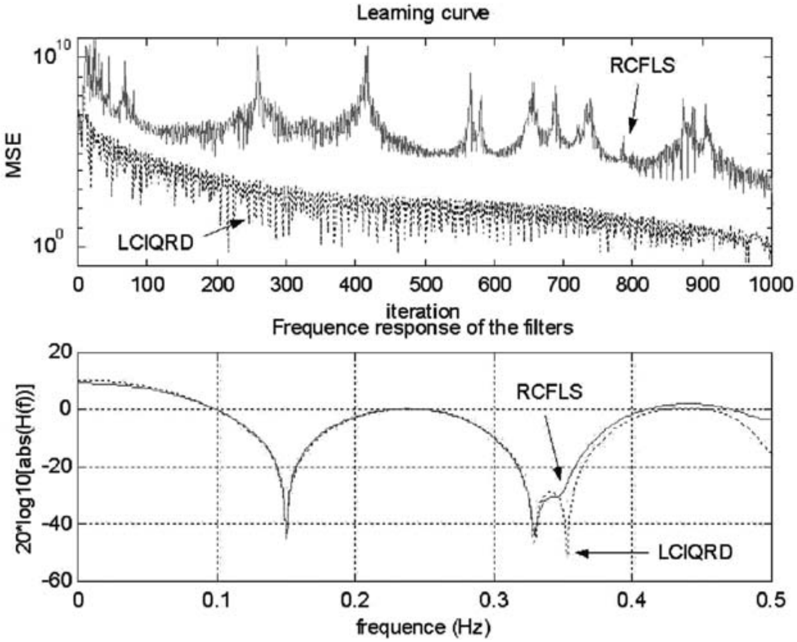

Fig. 12.4 Learning curve and frequency response of two algorithms after 1000 iterations with 100 independent runs.

Table 12.3 Comparison of nulling capability of RCFLS and LC-IQRD algorithms.

Iteration	Algorithms	Normalized frequency of the signals		
		0.15	0.33	0.35
1000	RCFLS	−44.05(dB)	−43.50(dB)	−28.35(dB)
1000	LCIQRD	−44.12(dB)	−44.04(dB)	−49.55(dB)
10000	RCFLS	−44.10(dB)	−45.63(dB)	−47.32(dB)
10000	LCIQRD	−44.99(dB)	−46.07(dB)	−54.46(dB)

where *eps* is the floating point relative accuracy and A is denoted as the correlation matrix. The smaller value of *eps* implies that a larger word-length is required to achieve a specific Tol; for instance, if we set $eps = 2^{-10}$, approximately, the word-length will be 10 bits. From the implementation point of view, *eps* can also be treated as the number of multiplication operator in the specific DSP device. Alternatively, Tol may be used to set a limit or the precision for our simulation environment. That is, for any singular value less than Tol will be treated as null or deriving a rounding error, during the computation procedure. In our simulation, the value of Tol is chosen to be 0.9453 and the corresponding value of *eps* to achieve the numerical accuracy of Tol is $eps = 2^{-7}$. In Figure 12.5, the results of the RCFLS algorithm, in terms of MSE, output power and the constrained drift, with the parameters described above are given. The jitters phenomenon of the RCFLS algorithm are found in

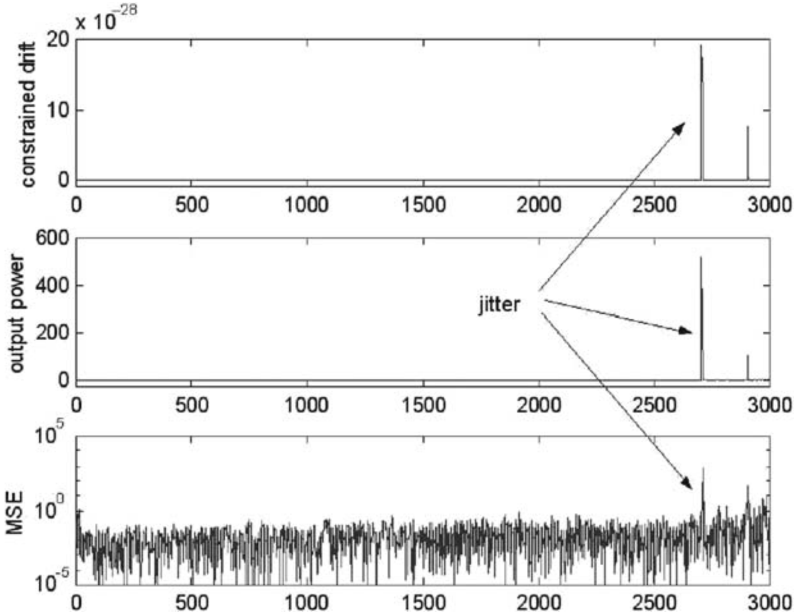

Fig. 12.5 Numerical properties of the RCFLS algorithm with $eps = 2^{-7}$.

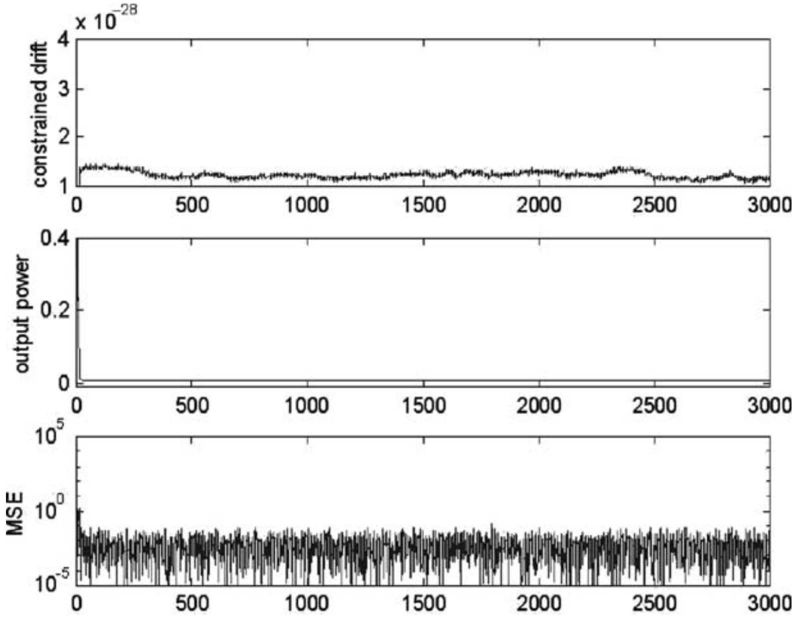

Fig. 12.6 Numerical properties of the LC-IQRD (LC-IQRD-RLS) algorithm with $eps = 2^{-7}$.

the learning curve (MSE), output power and the constrained drift, as depicted in Figure 12.5. However, as shown in Figure 12.6 with the same parameter as in Figure 12.5, the results of the LC-IQRD-RLS algorithm, in terms of MSE, output power and constrained drift, are much better than those of the RCFLS algorithm.

12.5.2 Application 2: Adaptive LCMV antenna array beamformer

There are two types of antenna array beamformers, viz., broadband array structure and narrowband array structure. In this application, the narrowband array beamformer structure is considered for interferences (or undesired signals) suppression. Basically, an array beamformer is a processor used in conjunction with an array of sensors to provide a versatile form of spatial filtering. Since the desired signal and the interference (or jammer) usually originate from different spatial locations, we are able to remove the jammer from the received signal by exploiting the spatial diversity at the receiver. The LCMV beamformer is known to be one of the most popular approaches for suppressing the undesired interference [16, 20, 23]. However, by using the adaptive array beamforming approach, the array system can automatically adjust its directional response to null the jammers, and thus enhances the reception of the desired signal.

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

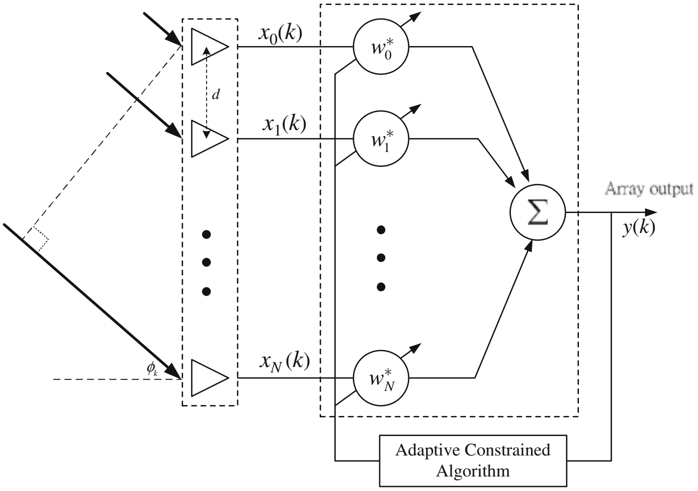

Fig. 12.7 Configuration of linearly constrained uniform spaced narrowband array beamformer.

position, towards 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 Figure 12.7. To start our derivation, let us consider a uniform linear array (ULA) and a wavefront, generated by a desired source of wavelength λ, propagating in an $N+1$ element array of sensors from a direction ϕ_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 $\phi_{nk} = \frac{2\pi}{\lambda}d(k-1)\sin\phi_k$. Moreover, assuming that the spacing between the array elements is set to $\lambda/2$, the array response vector of this $(N+1)$-antenna ULA can be denoted by

$$\mathbf{a}(\phi_k) = \left[1,\, e^{-j\pi\sin(\phi_k)},\, \cdots,\, e^{-j(N)\pi\sin(\phi_k)}\right]^{\mathrm{T}}. \tag{12.50}$$

Thus, we choose ϕ_k toward the direction of arrival (DOA) of the desired source signal and suitably adjust the weights of adaptive array; the array will pass the desired source signal from direction ϕ_0 and steer nulls toward interference sources located at ϕ_k for $k \neq 0$. It can be shown that an $(N+1)$ element array has $(N+1)\times 1$ degrees of freedom giving up to $(N+1)\times 1$ independent pattern nulls. So it has better performance if the array has more antenna elements. 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 ϕ_1, ϕ_2, and ϕ_3, respectively. For convenience, the look direction of the desired source signal is chosen to be $\phi_1 = 0°$. In the constrained approach of beamforming algorithm, the use of adaptive array for suppressing interference, with a look-direction constraint, is highly dependent on the accuracy of the *a priori* information (DOA) to achieve the maximum signal-to-interference-plus-noise ratio (SINR). However, an error in the steering angle, termed *pointing error*, would cause the adaptive array to tend to null out the desired signal as if it were a jammer. To verify the observation described earlier in this application, the problem of adaptive beamformer with main-beam constrained, associated with the pointing error, is considered. The deviation angle $\Delta\phi$ is defined as the discrepancy that the constraints look-direction, ϕ_c, deviates from the true arrival direction of the desired signal ϕ_1, i.e., $\Delta\phi = \phi_1 - \phi_c$. We note that, if pointing error exists (the look direction and the main-beam constraint is deviated due to estimation error of arrival angle), one of the conventional approaches would be the derivatives constraint approach [24, 25]. In such a case we may adapt the derivative constraints (DC) into the beamformer where constraint matrix **C**, with L linear constraints, is given by

$$\mathbf{C} = \left[\, \mathbf{a}(\phi_c)\ \mathbf{a}^{(1)}(\phi_c)\ \cdots\ \mathbf{a}^{L-1}(\phi_c) \,\right], \tag{12.51}$$

where $\mathbf{a}^i(\phi_c) = \frac{\partial a(\phi)}{\partial \phi^i}|_{\phi=\phi_c}$ is defined as the ith derivative of the steering vector with respect to ϕ.

Moreover, there are two design approaches to obtain the response vector **f**. The first approach is to use the conventional beamformer response, i.e.,

$$\mathbf{f} = \frac{1}{M}[\mathbf{a}^{\mathrm{H}}(\phi_c)\mathbf{a}(\phi_c), \mathbf{a}^{\mathrm{H}}(\phi_c)\mathbf{a}^{(1)}(\phi_c), \cdots, \mathbf{a}^{\mathrm{H}}(\phi_c)\mathbf{a}^{(L-1)}(\phi_c)]^{\mathrm{T}} = \frac{1}{M}\left[\mathbf{a}^{\mathrm{H}}(\phi_c)\mathbf{C}\right]^{\mathrm{T}}. \tag{12.52}$$

This will make the beamformer force the lobe shape of the main beam. In the second scheme, we set the other derivatives to a zero response, i.e., $\mathbf{f} = [1, 0, \cdots, 0]^{\mathrm{T}}$, and this could make the beamformer achieve the main beam with a flat top shape. Since we can expect that the jammer power is, in general, much larger than the desired signal source, the SNR is set to 0 dB. In our simulations, we have used three jammers with different jammer power ratios (JPR), e.g., $JNR_1 = 10$ dB, $JNR_2 = 20$ dB, and $JNR_3 = 40$ dB, corresponding to incident angles $-30°$, $35°$, and $40°$. First, we consider the case of main-beam constraint (single constraint) only, and assumed that there is no pointing error, e.g., $\Delta\phi = \phi_1 - \phi_c = 0°$. The results, in terms of nulling capability, are given in Figure 12.8 for both LC-IQRD-RLS and GSC-IQRD-RLS algorithms, with the forgetting factor $\lambda = 0.98$. These results were evaluated after 200 snapshots and correspond to an average of 500 independent runs. From Figure 12.8, we observe that both algorithms have identical beampatterns. We have also compared the results obtained with constrained LS algorithms with those of the linearly constrained LMS (Frost's algorithm) [5] and its GSC counterpart [26]. As can be seen in Figure 12.8, the LC-IQRD-RLS algorithm outperforms, in terms of nulling capability shown in the beampatterns, the LC-LMS and the GSC-LMS algorithms. Moreover, as described in the literature for the case where $\mathbf{B}^{\mathrm{H}}\mathbf{B} = \mathbf{I}$, the LC-LMS algorithm is identical to the GSC-LMS algorithm [21]. It is worth noting that, for the LC-IQRD-RLS algorithm, the condition $\mathbf{B}^{\mathrm{H}}\mathbf{B} = \mathbf{I}$ is not required for achieving the equivalency with the GSC-IQRD-RLS algorithm.

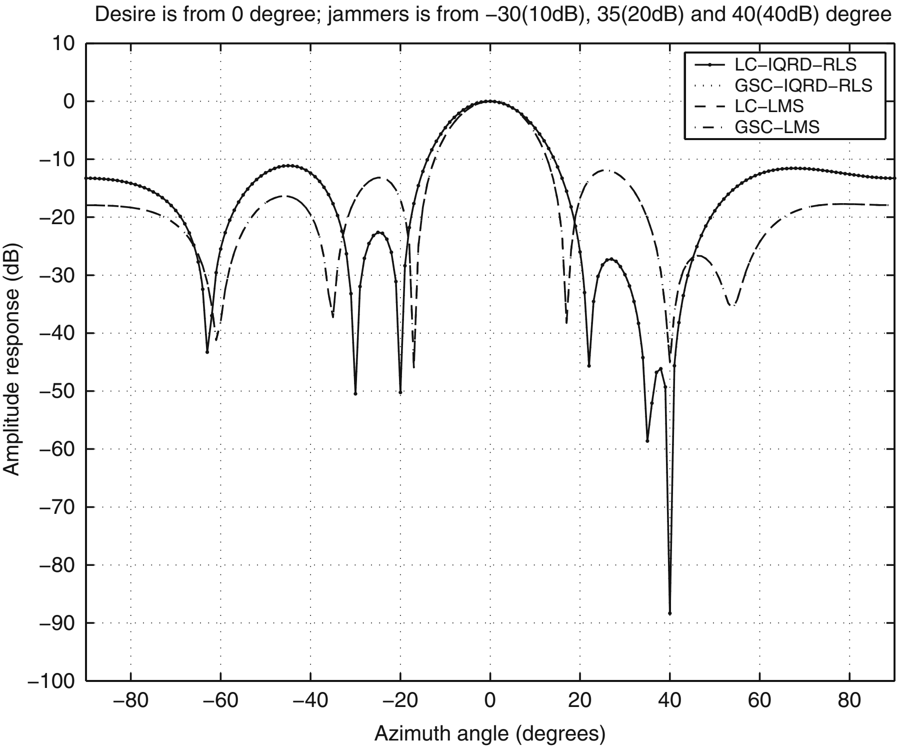

Fig. 12.8 Comparison of the GSC-IQRD-RLS algorithm with other algorithms, $\lambda = 0.98$ (without pointing error).

Next, we would like to see the effect due to pointing error, and to verify that the equivalency is also true for multiple constraint case. To do so, we consider the case that a pointing error, $\Delta\phi = \phi_1 - \phi_c = 3^\circ$, occurs (also known as DOA mismatch). Under this circumstance, with and without using the derivative constraint for the main-beam, the performance of the same algorithms are investigated. We let the other parameters be the same as in the case without having pointing error. From Figure 12.9, we learn that, with the single constraint (without using the derivative constraint), the gain of main beam is attenuated due to mismatch of the true look direction and the main-beam constraint for the LC-IQRD-RLS and the GSC-IQRD-RLS algorithms. Although the use of the LC-LMS algorithm has less effect due to pointing error, nulling capability is still worse than the one with the LC-IQRD-RLS algorithm. But, with the use of derivative constraints for the main-beam constraint (multiple constraints, e.g., the main beam and its first order constraints), the effect due to the pointing error has been alleviated while keeping better nulling capability than the LC-LMS algorithm. In this case, the performance of the LC-IQRD-RLS algorithm is again identical to the GSC-IQRD-RLS algorithm. Usually, we could use higher order derivative constraint to achieve better result of combating DOA

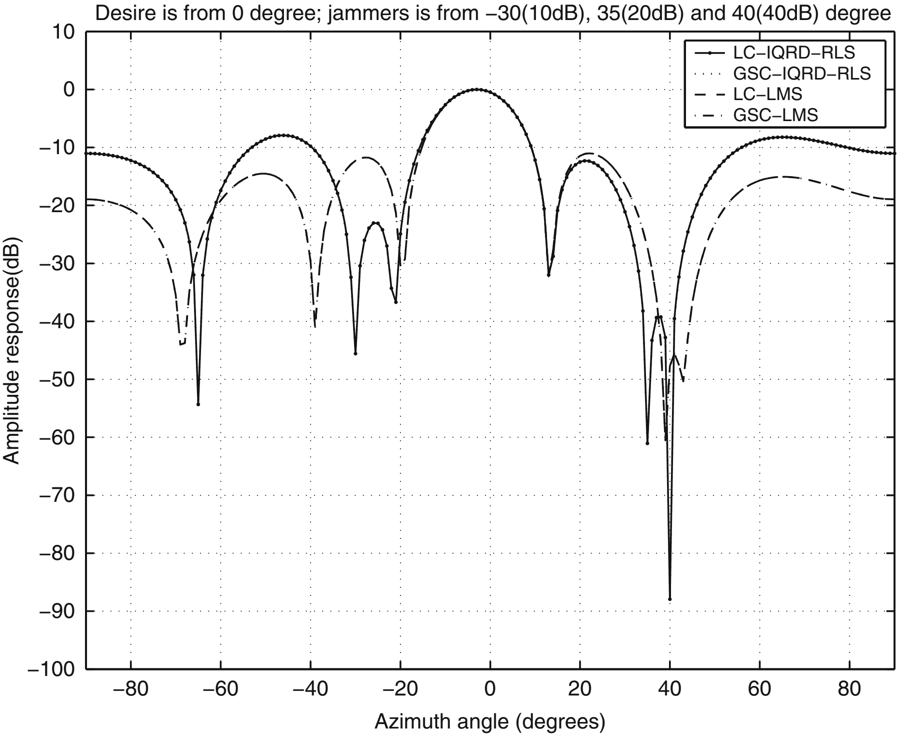

Fig. 12.9 Comparison of the GSC-IQRD-RLS algorithm with other algorithms, $\lambda = 0.98$ (pointing error with derivative constraint).

mismatch. For comparison, the results of nulling capability for Figures 12.8 and 12.9 are listed in Tables 12.4 and 12.5.

From Tables 12.4 and 12.5, we observe that the use of the first order main beam derivative constraint with the LC-LMS algorithm did not gain any benefit. Conversely, a significant gain improvement has been verified for the LC-IQRD-RLS algorithm.

From the results of the experiment carried out with this beamformer, we may conclude that, for the multiple constraint case, both the LC-IQRD-RLS and the GSC-IQRD-RLS algorithms did have the same performance. Although having a similar performance, the use of the GSC-IQRD-RLS algorithm has the advantage of requiring less computational complexity than the direct LC-IQRD-RLS algorithm.

Table 12.4 Comparison of nulling capability for various linearly constrained beamforming algorithms with single constraint (without pointing error).

	Desired signal	jammer 1	jammer 2	jammer 3
SNR	0 (dB)	10 (dB)	20 (dB)	40 (dB)
Azimuth algorithm	0^o	-30^o	35^o	40^o
LC-LMS algorithm	-0.445(dB)	-13.38(dB)	-27.44(dB)	-49.33(dB)
GSC-LMS algorithm	-0.445(dB)	-13.38(dB)	-27.44(dB)	-49.33(dB)
LC-IQRD-RLS algorithm	-4.169(dB)	-49.92(dB)	-58.08(dB)	-87.72(dB)
GSC-IQRD-RLS algorithm	-4.169(dB)	-49.92(dB)	-58.08(dB)	-87.72(dB)

Table 12.5 Comparison of nulling capability for various linearly constrained beamforming algorithms with multiples constraint (pointing error with derivative constraint).

	Desired signal	jammer 1	jammer 2	jammer 3
SNR	0 (dB)	10 (dB)	20 (dB)	40 (dB)
Azimuth algorithm	0^o	-30^o	35^o	40^o
LC-LMS algorithm	-0.4745(dB)	-12.3(dB)	-27.26(dB)	-47.7(dB)
GSC-LMS algorithm	-0.4745(dB)	-12.3(dB)	-27.26(dB)	-47.7(dB)
LC-IQRD-RLS algorithm	-0.4773(dB)	-45.58(dB)	-61(dB)	-87.95(dB)
GSC-IQRD-RLS algorithm	-0.4773(dB)	-45.58(dB)	-61(dB)	-87.95(dB)

12.6 Conclusion

In this chapter, we developed the direct and indirect IQRD-RLS-based constrained adaptive filtering algorithms, named the LC-IQRD-RLS and the GSC-IQRD-RLS algorithms, respectively, to implement the LCMV filter. The IQRD approach was chosen such that the constrained LS weight vector solution could be updated without using back-substitution, which is suitable to be implemented using a typical VLSI technology structure termed systolic array. To verify the merits of the LC-IQRD-RLS and the GSC-IQRD-RLS algorithms, we applied them to spectral analysis and smart antenna array beamforming problems. We have shown that, due to the numerical stability of evaluating the adaptation (or Kalman) gain via *Givens rotations*, the proposed LC-IQRD-RLS algorithm had less effect of constraint drift compared with the CFLS algorithm and its robust version [2]. Thus, we concluded that the LC-IQRD-RLS algorithm proposed in this chapter did perform better than the CFLS and RCFLS algorithms developed in [2], in terms of capability to null the undesired signal components output power as well as numerical efficiency in practical implementation.

References

1. S. J. Chern and C. Y. Chang, Adaptive linearly constrained inverse QRD-RLS beamformer for moving jammers suppression. IEEE Transactions on Antennas and Propagation, vol. 50, no. 8, pp. 1138–1150 (August 2002)

2. L. S. Resende, J. T. Romano, and M. G. Bellanger, A fast least-squares algorithm for linearly constrained adaptive filtering. IEEE Transactions on Signal Processing, vol. 44, no. 5, pp. 1168–1174 (May 1996)
3. D. H. Johnson and Dan E. Dudgeon, Array Signal Processing Concepts and Techniques. Prentice-Hall, Englewood Cliffs, NJ, USA (1993)
4. S. N. Lin and S. J. Chern, A new adaptive constrained LMS time delay estimation algorithm. Signal Processing (Elsevier), vol. 71, pp. 29–44 (November 1998)
5. O. L. Frost III, An algorithm for linearly constraint adaptive array processing. Proceedings of IEEE, vol. 60, no. 8, pp. 926–935 (August 1972)
6. S. J. Chern and C. Y. Chang, Adaptive MC-CDMA receiver with constrained constant modulus IQRD-RLS algorithm for MAI suppression. Signal Processing (Elsevier), vol. 83, no. 10, pp. 2209–2226 (October 2003)
7. J. B. Schodorf and D. W. Williams, Array processing techniques for multiuser detection. IEEE Transactions on Communications, vol. 45, no. 11, pp. 1375–1378 (November 1997)
8. S. Haykin, Adaptive Filter Theory. 3rd edition Prentice-Hall, Inc. Englewood Cliffs, NJ, USA (1996)
9. J. M. Cioffi and T. Kailath, Fast RLS transversal filters for adaptive filtering. IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 32, pp. 304–337 (June 1984)
10. J. M. Cioffi, Limited precision effects for adaptive filtering. IEEE Transactions on Circuits and Systems, vol. 34, pp. 821–833 (July 1987)
11. P. A. Regalia and M. G. Bellanger, On the duality between fast QR methods and lattice methods in least squares adaptive filtering. IEEE Transactions on Signal Processing, vol. 39, pp. 879–891 (April 1991)
12. G. H. Golub and C. F. Van Load, Matrix Computation. 3rd edition John Hopkins University Press, Baltimore, MD, USA (1996)
13. Z. S. Liu, QR Method of O(N) complexity in adaptive parameter estimation. IEEE Transactions on Signal Processing, vol. 43, no. 3, pp. 720–729 (March 1995)
14. J. M. Cioffi, High speed systolic implementation of fast QR adaptive filters. IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'88, New York, pp. 1584–1587 (April 1988)
15. H. Leung and S. Haykin, Stability of recursive QRD-RLS algorithm using finite precision systolic array implementation. IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 37, pp. 760–763 (May 1989)
16. M. Moonen, Systolic MVDR Beamforming with inverse updating. Proceedings IEE Pt F, vol. 140, no. 3, pp. 175–178 (March 1993)
17. C.-F. T. Tang, *Adaptive Array Systems Using QR-Based RLS and CRLS Techniques with Systolic Array Architectures*. Ph.D. thesis - Department of Electrical Engineering, University of Maryland, College Park, MD, USA (1991)
18. S. T. Alexander and A. L. Ghirnikar, A method for recursive least squares filtering based upon an inverse QR decomposition. IEEE Transactions on Signal Processing, vol. 41, no. 1, pp. 20–30 (January 1993)
19. D. T. M. Slock and T. Kailath, Numerical stable fast transversal filters for recursive least squares adaptive filtering. IEEE Transactions on Signal Processing, vol. 39, no. 1, pp. 92–114 (January 1991)
20. M. Moonen and I. K. Proudler, MVDR beamforming and generalized sidelobe cancellation based on inverse updating with residual extraction. IEEE Transactions on Circuit and System II, vol. 47, no. 4, pp. 352–358 (April 2000)
21. B. R. Breed and J. Strauss, A short proof of the equivalence of LCMV and GSC beamforming. IEEE Signal Processing Letters, vol. 9, no. 6, pp. 168–169 (June 2002)
22. J. A. Apolinário Jr. and M. L. R. de Campos, The constrained conjugate gradient algorithm. IEEE Signal Processing Letters, vol. 7, no. 12, pp. 351–354 (December 2000)
23. R. A. Games, W. L. Estman, and M. J. Sousa, Fast algorithm and architecture for constrained adaptive side-lobe cancellation. IEEE Transactions on Antennas and Propagation, vol. 41, no. 5, pp. 683–686 (May 1993)

24. C. Y. Chang and S. J. Chern, Derivative constraint narrowband array beamformer with new IQML algorithm for wideband and coherent jammers suppression. IEICE Transactions on Communications, vol. E86-B, no. 2, pp. 829–837 (February 2003)
25. W. G. Najm, Constrained least squares in adaptive, imperfect arrays. IEEE Transactions on Antennas and Propagation, vol. 38, no. 11, pp. 1874–1878 (November 1990)
26. L. J. Griffiths and C. W. Jim, An alternative approach to linearly constrained adaptive beamforming. IEEE Transactions on Antennas and Propagation, vol. AP-30, no. 1, pp. 27–34 (January 1982)