IMPLEMENTATION OF ADAPTIVE GENERALIZED SIDELOBE CANCELLERS USING EFFICIENT COMPLEX VALUED ARITHMETIC

GEORGE-OTHON GLENTIS[*]

[*] TEI of Crete, Branch at Chania, Department of Electronics
3 Romanou Str, Chania 73133, Crete, Greece
email: gglentis@chania.teiher.gr

Low complexity realizations of Least Mean Squared (LMS) error, Generalized Sidelobe Cancellers (GSCs) applied to adaptive beamforming are considered. The GSC method provides a simple way for implementing adaptive Linear Constraint Minimum Variance (LCMV) beamformers. Low complexity realizations of adaptive GSCs are of great importance for the design of high sampling rate, and/or small size and low power adaptive beamforming systems. The LMS algorithm and its Transform Domain (TD-LMS) counterpart are considered for the adaptive processing task involved in the design of optimum GSC systems. Since all input signals are represented by complex variables, complex valued arithmetic is utilized for the realization of GSC algorithms, either on general purpose computers, or on dedicated VLSI ASICs. Using algorithmic strength reduction (SR) techniques, two novel algorithms are developed for efficient realizations of both LMS GSCs and TD-LMS GSC schemes. Both of the proposed algorithms are implemented using real valued arithmetic only, whilst reducing the number of multipliers by 25% and 20%, respectively. When VLSI implementation aspects are considered, both the proposed algorithms result in reduced power dissipation and silicon area realizations. The performance of the proposed realizations of the LMS based GSC methods is illustrated in the context of typical beamforming applications.

Keywords: adaptive beamforming, generalized sidelobe canceller, LMS algorithm, complex valued arithmetic

1. Introduction

Adaptive beamforming is a powerful technique of enhancing a signal of interest while suppressing the interference signal and noise at the output of an array of sensors. An array of sensors consists of a set of sensor elements that are spatially arranged at known locations (Compton, 1988; Drabowitch *et al.*, 1998; Haykin, 1996; Hudson, 1991; Johnson and Dudgeon, 1993; Monzingo and Miller, 1980; Mucci, 1984; Pillai, 1989; Widrow and Stearns, 1985). By tuning the amplitude and phase of the wavefronts at each sensor element, it is possible to electronically steer the beam to a desired direction and to place nulls in other directions (Krim and Viberg, 1996; van Veen and Buckley, 1988). In other words, an adaptive array continuously modifies its beampattern in a desired way by means of an adaptive optimization algorithm. The array beampattern is optimized so that maximum gains are offered in specific directions corresponding to the desired signal, while maximum attenuation is placed in specific directions that correspond to the undesired interference signal or jammer. Adaptive beamforming has been sucessfully utilized in a wide range of applications. Typical examples include antennas (Chryssomallis 2000; Godara, 1997; Kohno, 1998; Rappaport, 1998), radar (Nitzberg, 1999), radio astronomy (Li *et al.*, 2002), speech processing, (Li *et al.*, 2002), wireless communication (Litva and Lo, 1996; Winters, 1998), biomedical signal processing (Bailler *et al.*, 2001), nondestructive testing of materials (Ghorayeb *et al.*, 1994), etc.

A broadband adaptive beamformer consists of a multi-input single output linear combiner and an adaptive algorithm that adjusts the weights in some optimal way (Buckley 1986; 1987; Frost, 1972; van Veen and Buckley, 1988). Broadband beamforming is employed when the nature of the signals of interest is wideband. The Linearly Constrained Minimum Variance (LCMV) beamformer is designed by minimizing the array output power subject to a set of linear constraints. An efficient adaptive implementation of the LCMV method was proposed by Griffiths and Jim, and it is known as the Generalized Sidelobe Canceller (GSC), where the constrained optimization problem is transformed to an unconstrained one (Griffiths and Jim, 1982). The GSC method has been considered for the implementation of adaptive beamforming in various applications. Typical examples include: speech acquisition and enhancement in noisy and reverberating environments (Hoshuyama *et al.*, 1999; Gannot *et al.*, 2001), interference cancellation in radio astronomy, where array radio telescopes are used for deep space observations

(Le *et al.*, 2002; Leshem *et al.*, 2000), combating multiple access interference and multipath fading in CDMA telecommunications systems, multiuser detection, adaptive interference cancellation and channel equalization, in training or in blind mode (Honig and Tsatsanis, 2000; Lee and Tsai, 2001; Xu and Tsatsanis, 1999; Yu and Ueng, 2000), the design of adaptive antennas in mobile communications systems, for spatial filtering for interference reduction (Boukalov and Haggman, 2000; Godara, 1997; Paulraj and Papadias, 1997; Rapaport, 1998), adaptive operation of electronically phased radars (Farina, 1992; Nitzberg, 1999), sidelobe and/or hot clutter cancelling in radar systems (Hendon and Reed, 1990; Kogon *et al.*, 1996; Scott and Mulgrew, 1995).

In the original GSC method proposed by Griffiths and Jim (1982), the optimum array parameters are adaptively estimated based on the available data set and using an LMS adaptive filter, resulting in an LMS GSC adaptive scheme. LMS like algorithms are popular due to low computational complexity and simplicity in the hardware realization of the underlying algorithmic structure (Haykin, 1996; Glentis *et al.*, 1999; Kalouptsidis and Theodoridis, 1993). However, the convergence rate of an LMS based GSC method heavily depends on the eigenvalue spread of the correlation matrix of the input data. In an attempt to improve the convergence rate of the original LMS GSC scheme, Discrete Unitary Transforms, such as the Discrete Fourier Transform, have been utilized in order to decorrelate the input data (An and Champagne, 1994; Chen and Fang, 1992; Chu and Fang, 1999; Herbordt and Kellermann, 2001; Goldstein *et al.*, 1994; Joho and Moschytz, 1997; Moon *et al.*, 2001; Yu and Leou, 2000). The Transformed Domain LMS GSC algorithms may have increased convergence rates for some classes of input signals, yet the computational complexity remains similar to that of the original LMS based scheme.

In this paper, two novel and efficient algorithms implementing the LMS GSC and the Transform Domain LMS GSC adaptive schemes are presented. Direct application of the *fast* complex valued multiplication method to the original LMS GSC and the TD-LMS GSC algorithms can reduce the number of real multiplications at the expense of an increased number of adders (Lamagna, 1982; Winograd, 1980). To overcome this difficulty, complex signals are treated as pairs of real signals, and operations are re-organized based on the real arithmetic only. The algorithmic strength reduction technique (Chandrakasan and Brodersen, 1995; Parhi, 1999; Shanbhag, 1998), is subsequently applied to the LMS GSC, as well as to the TD-LMS GSC adaptive algorithm, which results in significant computational savings. Both algorithms proposed in this paper utilize real arithmetic only for an efficient realization of both the transversal filtering and parameter updating parts. The computational complexity of the proposed schemes, measured by the number of real multipliers, is reduced by 25% and 20% for the LMS GSC and the TD-LMS GSC algorithms, with no or a marginal increase in the number of real adders, respectively. Computational savings, without sacrificing performance, are a task of primary interest in the design of high speed adaptive array systems, where the processing power of several GOPS (Giga operations per second) is needed (Ahlander *et al.*, 1996; Boukalov and Haggman, 2000; Martinez, 1999; Taveniku and Ahlander, 1997). On the other hand, in slow sampling rate adaptive beamforming systems, e.g., in speech processing and hearing aids applications, where special VLSI ASICs are required for small size and low power implementation, the reduction in the computational complexity is of paramount interest, since it is directly related to the size and power consumption of the final design (Parhi, 1999).

The performance of the proposed realizations of the LMS based GSC methods is illustrated in the context of beamforming applications.

2. Generalized Sidelobe Canceller

Let us consider a linear array of sensors, which consists of K equally spaced sensor elements. Let $P-1$ be the number of delay elements associated with each elementary array input. The input signal induced at each array element is denoted by $v_i(n),\ i=1,2,\dots,K$. The structure of the Generalized Sidelobe Canceller is depicted in Fig. 1. Here $\tilde{v}_i(n),\ i=1,2,\dots,K$, are the signals obtained by passing the array output through delay elements, needed to steer the array in the desired look direction, ϕ, i.e., $\tilde{v}_i(n)=v_i(n-\tau_i)$. These input signals are transformed by a vector $\mathbf{b}$ and a matrix $\mathbf{B}$ into a main channel signal, $x_0(n)$, and $K-1$ auxiliary channel signals, $x_i(n),\ i=1,\dots,K-1$, as

$$x_0(n)=\mathbf{b}^H\tilde{\mathbf{v}}(n), \tag{1}$$

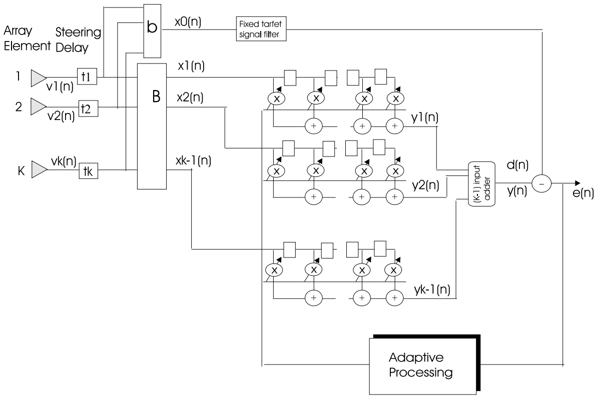


Fig. 1. The GSC adaptive beamformer. Small rectangular boxes represent unit delays.

$$[x_1(n)\ x_2(n)\ \dots\ x_{K-1}(n)]^T = \mathbf{B}\tilde{\mathbf{v}}(n), \quad (2)$$

where $\tilde{\mathbf{v}}(n) = [\tilde{v}_1(n) \dots \tilde{v}_K(n)]^T$. (As for the superscripts throughout this paper, '*' means the complex conjugate, 'T' denotes the vector transpose, and 'H' stands for the Hermitian operator (conjugate and transpose).) Constant $\mathbf{b}$ and $\mathbf{B}$ are designed such that the target signal is prevented from passing through to the auxiliary channels, while it is allowed to pass unimpeded to the main channel. A possible choice is to set all elements of $\mathbf{b}$ as equal to 1, while setting $\mathbf{B}$ to have elements from the Walsh-order Walsh function (Griffiths and Jim, 1982).

The desired response signal, $z(n)$, is obtained by passing the primary signal $x_0(n)$ through a fixed target signal filter that is used to control the frequency response of the beamformer in the look direction. The auxiliary signals are fed to a set of tapped delay lines, each with $P-1$ unit delay elements. The output signal $y(n)$ is obtained by the linear regression

$$y(n) = \mathbf{W}^H \mathbf{X}(n), \quad (3)$$

where $\mathbf{W}$ is a vector that carries the coefficients of the multichannel linear combiner, and $\mathbf{X}(n)$ is a vector that carries the auxiliary input data. More specifically,

$$\mathbf{X}(n) = [\mathbf{x}_1^T(n)\,\mathbf{x}_2^T(n) \dots \mathbf{x}_{K-1}^T(n)]^T, \quad (4)$$

where $\mathbf{x}_i(n)$ is given by

$$\mathbf{x}_i(n) = \big[x_i(n)\, x_i(n-1) \dots x_i(n-P+1)\big]^T. \quad (5)$$

$\mathbf{W}$ is organized in a similar way. Vectors $\mathbf{X}(n)$ and $\mathbf{W}$ have dimensions $(K-1)P \times 1$.

The weight vector $\mathbf{W}$ is estimated on the basis of the incoming data statistics, minimizing the MSE of the error signal between the conventional beamformer output, $z(n)$, and the output of the sidelobe canceller, $y(n)$, i.e.,

$$\mathbf{W}: \quad \min_W \boldsymbol{\mathcal{E}}\left(|e(n)|^2\right), \quad e(n) = z(n) - y(n). \quad (6)$$

Thus, the optimum MSE solution is obtained by solving a system of linear equations of the form

$$\mathbf{RW} = \mathbf{r}. \quad (7)$$

Here $\mathbf{R} = \boldsymbol{\mathcal{E}}[\mathbf{X}(n)\mathbf{X}^H(n)]$ is the covariance matrix of the auxiliary input signals and $\mathbf{r} = \boldsymbol{\mathcal{E}}[\mathbf{X}(n)z^*(n)]$ is the cross correlation vector between the auxiliary input signals and the conventional beamformer output. The relationship between the LCMV and GSC solutions for the adaptive beamforming problem is well established, and it is discussed in detail in (Griffiths and Jim, 1982; Haykin, 1996; van Veen and Buckley, 1988).

In practice, the covariance matrix $\mathbf{R}$ and the cross correlation vector $\mathbf{r}$ are not known in advance, and have to be estimated for a given number of samples. The direct computation of the sample based estimates of $\mathbf{R}$ and $\mathbf{r}$, followed by a linear system solver, could be utilized for the estimation of the optimum parameters, resulting in a *batch* processing method. However, this is a major drawback in many real time applications since large amounts of data have to be collected and stored in advance. An alternative way to alleviate this difficulty is the use of recursive stochastic approximation schemes which update the estimator of the optimum parameters whenever new data are available. This approach leads to *adaptive* processing schemes for the estimation of the system parameters sought (Haykin, 1996; Glentis *et al.*, 1999; Kalouptsidis and Theodoridis, 1993).

2.1. Adaptive LMS GSC Algorithms

Two different approaches leading to two widely used algorithmic families have been adopted for the adaptive estimation of optimum MSE parameters on the basis of the available data set (Widrow and Stearns, 1985; Haykin, 1996). The first one is based on the stochastic approximation of the steepest descent method and is known as the Least Mean Squared (LMS) family. The latter is based on the stochastic approximation of the Gauss-Newton method and is known as the Recursive Least Squares (RLS) family. The simplest form of the LMS algorithm offers adaptive filtering with a cost of twice the number of the unknown system parameters. However, the convergence rate of the algorithm heavily depends on the eigenvalue spread of the correlation matrix of the input data. On the other hand, although the RLS algorithm does not suffer from such a drawback, it has a complexity that is proportional to the squared number of the unknown system parameters. Classic or fast, RLS and QR-RLS type adaptive algorithms were proposed for adaptive beamforming (Farina *et al.*, 1996; Farina and Timmoneri, 1999; Huard and Yen, 1994; Kim *et al.*, 1992; Lee *et al.*, 1987; Li and Gaillard, 1988; Timmoneri *et al.*, 1994; Yuen, 1991). However the computational complexity of these methods is either $O(K^2P^2)$ flops or $O(K^2P)$ flops per processed sample, depending on the method applied. Since adaptive beamforming has to be performed in real time, low cost algorithms should be employed. The LMS based schemes offer adaptive processing at a lower cost, and thus they are more attractive for real time implementation of the adaptive beamforming processing task.

The LMS algorithm applied to the GSC case is summarized as follows (Griffiths and Jim, 1982):

$$y(n) = \mathbf{W}^H(n-1)\mathbf{X}(n), \quad (8)$$

$$e(n) = z(n) - y(n), \quad (9)$$

$$\mathbf{W}(n) = \mathbf{W}(n-1) + \mu\mathbf{X}(n)e^*(n). \quad (10)$$

The parameter μ is a positive constant that controls the convergence speed of the algorithm. The LMS GSC adaptive algorithm, reorganized in a channel-wise format, is presented in Table 1.

Table 1. Computation flow chart of the LMS GSC algorithm.

LMS GSC algorithm			
Eqn.	Function	CMUL	CADD
1	FOR $i = 1$ TO $K-1$ $\mathbf{x}_i(n) = [x_i(n) \;\; x_i(n-1)$ $\dots \;\; x_i(n-P)]^T$ $y_i(n) = \mathbf{w}_i^H(n-1)\mathbf{x}_i(n)$ END i	P	$P-1$
2	$y(n) = \sum_{i=1}^{K-1} y_i(n)$	0	$K-2$
3	$e(n) = d(n) - y(n)$	0	1
4	FOR $i = 1$ TO $K-1$ $\mathbf{w}_i(n) = \mathbf{w}_i(n-1)$ $+\mu\mathbf{x}_i(n)e^*(n)$ END i	P	P
	TOTAL COST	$2(K-1)P$	$2(K-1)P$

In an attempt to improve the convergence properties of the adaptive LMS GSC, efficient schemes have been developed that utilize appropriate preconditioning to accelerate the convergence speed of the original LMS scheme (An and Champagne, 1994; Chen and Fang, 1992; Chu *et al.*, 1999; Herbordt and Kellermann, 2001; Goldstein *et al.*, 1994; Joho and Moschytz, 1997; Moon *et al.*, 2001; Yu and Leou, 2000). Fast transforms, such as DFT or DCT, have been employed in order to decorrelate the input data, and, as a result, to obtain low cost performance indices better that those of the conventional LMS. Although preconditioning using a fixed transformation may not always result in dramatic improvements, the low computational complexity often makes this choice a possible alternative.

The Transform Domain GSC-LMS algorithm is described by the following set of equations (Chen and Fang, 1992):

$$\mathbf{f}(n) = \boldsymbol{\mathcal{S}}\mathbf{X}(n), \tag{11}$$

$$e(n) = z(n) - \boldsymbol{\mathcal{W}}^H(n-1)\mathbf{f}(n), \tag{12}$$

$$\boldsymbol{\mathcal{W}}(n) = \boldsymbol{\mathcal{W}}(n-1) + \mu\boldsymbol{\mathcal{P}}^{-1}\mathbf{f}(n)e^*(n). \tag{13}$$

The transformation matrix $\boldsymbol{\mathcal{S}}$ has the form

$$\boldsymbol{\mathcal{S}} = \begin{bmatrix} \mathbf{S} & 0 & \dots & 0 \\ 0 & \mathbf{S} & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{S} \end{bmatrix},$$

where $\mathbf{S}$ is a unitary matrix of dimensions $P \times P$. Thus the transformed domain data vector $\mathbf{f}(n)$ is obtained by a channel-wise transformation of the original data vector $\mathbf{X}(n)$. Obviously,

$$\mathbf{f}(n) = [\mathbf{f}_1^T(n) \dots \mathbf{f}_{K-1}^T(n)]^T, \tag{14}$$

where

$$\mathbf{f}_i(n) = \mathbf{S}\mathbf{x}_i(n), \quad i = 1, \dots, K-1. \tag{15}$$

$\boldsymbol{\mathcal{W}}(n)$ is the vector which contains the coefficients of the transformed array and is organized as follows:

$$\boldsymbol{\mathcal{W}}(n) = [\mathbf{W}_1^T(n) \;\; \mathbf{W}_2^T(n) \;\; \dots \;\; \mathbf{W}_{K-1}^T(n)]^T. \tag{16}$$

The estimated transformed filter coefficients are related to the original ones, as $\boldsymbol{\mathcal{W}} = \boldsymbol{\mathcal{S}}^H\mathbf{W}$. The parameter μ is a positive constant that controls the convergence speed of the algorithm.

$\boldsymbol{\mathcal{P}}$ is the covariance matrix of the transformed data, defined as $\boldsymbol{\mathcal{P}} = \boldsymbol{\mathcal{E}}[\boldsymbol{\mathcal{X}}(n)\boldsymbol{\mathcal{X}}^H(n)]$. The role of $\boldsymbol{\mathcal{P}}^{-1}$ in the recursive equation (13) is to reduce the eigenvalue spread of the corresponding system matrix. The inverse $\boldsymbol{\mathcal{P}}^{-1}$ is approximated by a diagonal matrix of the form

$$\boldsymbol{\mathcal{P}}^{-1} \approx \begin{bmatrix} \mathbf{P}_1^{-1} & 0 & \dots & 0 \\ 0 & \mathbf{P}_2^{-1} & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{P}_{K-1}^{-1} \end{bmatrix}. \tag{17}$$

Each submatrix $\mathbf{P}_i$ has a diagonal structure with entries being the powers of the individual input signals associated with each frequency bin. It is defined as

$$\mathbf{P}_i = \mathrm{diag}\left[p_{i,1}, \; p_{i,2}, \; \dots p_{i,P}\right], \tag{18}$$

where $p_{i,k}$ is the power of the i-th input signal at the k-th frequency bin, i.e.,

$$p_{i,k} = \boldsymbol{\mathcal{E}}\left[|f_{i,k}(n)|^2\right]. \tag{19}$$

In practice, $\mathbf{P}_i$ is a time varying matrix whose elements are calculated in terms of available data, e.g., using an exponentially weighted power estimator implemented by the difference equation

$$\begin{gathered} p_{i,k}(n) = \lambda p_{i,k}(n-1) + (1-\lambda)|f_{i,k}(n)|^2, \\ \lambda \in (0,1). \end{gathered} \tag{20}$$

When the Discrete Fourier Transform (DFT) is used as the unitary transform, the input data are continuously transformed into the frequency domain. The Sliding Window Discrete Fourier Transform (SW-DFT) can be utilized to perform this task (Narayan *et al.*, 1983; Shynk, 1992).

Table 2. Computation flow chart of the Transform Domain LMS GSC algorithm.

Transform Domain LMS GSC algorithm					
Eqn.	Function	CMUL	CADD	RMUL	RADD
	FOR $i=1$ TO $K-1$				
1	$u_i(n) = x_i(n) - \rho^P x_i(n-P)$	0	1	0	0
	FOR $m=0$ TO $P-1$				
2	$f_{i,m+1}(n) = \rho e^{-\jmath\frac{2\pi m}{P}} f_{i,m+1}(n-1) + u_i(n)$	1	1	0	0
3	$p_{i,m+1}(n) = \lambda p_{i,m+1}(n-1) + (1-\lambda)\lvert f_{i,m+1}(n)\rvert^2$	0	0	2	2
4	$F_{i,m+1}(n) = \mu\frac{f_{i,m+1}(n)}{p_{i,m+1}(n)}$	0	0	2	0
	END m				
	$\mathbf{f}_i(n) = [f_1(n)\dots f_P(n)]^T$				
	$\mathbf{F}_i(n) = [F_1(n)\dots F_P(n)]^T$				
5	$y_i(n) = \mathbf{W}_i^H(n-1)\mathbf{f}_i(n)$	P	$P-1$	0	0
	END i				
6	$y(n) = \sum_{i=1}^{K-1} y_i(n)$	0	$K-2$	0	0
7	$e(n) = d(n) - y(n)$	0	1	0	0
	FOR $i=1$ TO $K-1$				
8	$\mathbf{W}_i(n) = \mathbf{W}_i(n-1) + \mathbf{F}_i(n)e^*(n)$	P	P	0	0
	END i	0	0	0	0
	TOTAL COST	$3(K-1)P$	$(K-1)(3P+1)$	$4(K-1)P$	$2(K-1)P$

The SW-DFT estimates the DFT transform of a rectangular window of the signal, which is continuously updated with new samples as the oldest ones are discarded. In this case, $\mathbf{f}_i(n)$ is the SW DFT of the input data $\mathbf{x}_i(n)$, and each element of $\mathbf{W}_i(n)$ is associated with a specific frequency band. The Sampling Frequency (FS) structure, a method for implementing the SDFT based on a set of filter banks, is very popular in adaptive filtering (Shynk, 1992) due to the low computational complexity, the regularity and modularity, i.e., the facts that are of great importance when a high speed implementation on VLSI array processors is considered. The SW-DFT is implemented using the system of first order recursive equations of the form

$$\begin{aligned} f_{i,m+1}(n) &= \rho e^{-\jmath\frac{2\pi m}{P}} f_{i,m+1}(n-1) + x_i(n) \\ &\quad - \rho^P x_i(n-P), \quad m=0,\dots,P-1. \end{aligned} \tag{21}$$

Here $\rho \in (0,1)$ is a stabilization factor that is used to compensate for the marginal stability of the original realization.

The TD-LMS GSC adaptive algorithm is tabulated in a channel-wise format, cf. Table 2.

3. Algorithmic Strength Reduction

Complex valued arithmetic is required for the implementation of both the LMS GSC and TD-LMS GSC algorithms, since in most applications the input signals of both algorithms are represented by complex variables. Complex valued addition is realized by a set of two real valued additions, i.e.,

$$(a+\jmath b) + (c+\jmath d) = (a+c) + \jmath(b+d). \tag{22}$$

Complex valued multiplication can be realized by the *classical* methods that require four real valued multiplications and two real valued additions, i.e.,

$$(a+\jmath b)(c+\jmath d) = (ac-bd) + \jmath(ad+bc). \tag{23}$$

Alternatively, the *fast* complex valued multiplication method can be applied, where the inherent dependencies of the partial products and sums, are utilized for the reduction of the number of real valued multipliers, at the expense of some extra real valued adders (Parhi, 1999; Shanbhag, 1998; Winograd, 1980). In this case, three real valued multiplications and five real valued additions are required. One possible implementation of a *fast* complex valued multiplication is described by the following equation (Shanbhag, 1998):

$$\begin{aligned} ac - bd &= (a-b)d + a(c-d), \\ ad + bc &= (a-b)d + b(c+d). \end{aligned} \tag{24}$$

The advantages of the *fast* complex valued multiplication approach are the following: (A) Although Eqn. (24) requires two extra operations, it does however utilize fewer multiplications. Since multiplication is executed more slowly than addition on most digital computers, it is expected that the *fast* complex valued multiplication method is faster than the classical one. (B) As far as the design of dedicated VLSI ASIC processors is considered, the reduction in the number of the required multiplications can (a) reduce the total silicon area, since multipliers occupy much more space compared with adders, and (b) reduce the power consumption of the ASIC, since the power consumption of CMOS circuits depends on the effective switched capacitance of the underlying architecture, which is mainly dependent on the number of logic transitions per unit time.

The critical path of the *fast* complex valued multiplication method is slightly larger than that of the *classical* method. Indeed, the iteration period of Eqn. (23) is $T_{\text{classical}} = t_{RMUL} + t_{RADD}$, while $T_{\text{fast}} = t_{RMUL} + 2t_{RADD}$. ($t_{RMUL}$ and t_{RADD} denote the times required for the execution of real valued multiplication and addition, respectively.) However, this is not a major problem, and it can be resolved by proper pipelining of the task, using look ahead transformations (Shanbhag, 1998).

Efficient signal processing algorithms for digital and adaptive filtering have been developed by taking into account the reduced complexity *fast* complex valued multiplication method, (Baghaie, 1999; Baghaie and Laakso, 1998; Perry *et al.*, 1999; Shanbhag, 1998). In the context of VLSI signal processing, transformations are modifications in the computational structure of a given algorithm such that the input-output behavior is preserved. The application of the *fast* complex valued multiplication method belongs to a general class of algorithmic transformations, and is known as the *algorithmic strength reduction transform*. It has been successfully applied to the design of low power, high speed adaptive filters and equalizers (Chandrakasan end Brodersen, 1995; Shanbhag, 1998).

The direct implementation of the *fast* complex valued multiplication method in LMS GSC and TD-LMS GSC schemes can reduce the number of real multiplications. However, although the number of real valued multipliers is reduced, there is a significant increase in the number of real valued adders and in the total number of the overall real valued operations. A further improvement in the number of the required real valued adders can be achieved by the re-organization of both algorithms, using a set of auxiliary signals and filter parameters that are propagated through the algorithm and are updated by the pertinent recursive equations in accordance with the strength reduction real valued arithmetic imposed by Eqn. (24).

3.1. SR LMS GSC Algorithm

The application of the algorithmic strength reduction transform described by Eqn. (24) to the LMS GSC algorithm results in an equivalent algorithmic description, called the SR LMS GSC algorithm, where real valued arithmetic is only required. The main feature of the SR LMS GSC algorithm is the introduction of a set of auxiliary input signals that are utilized as inputs to the transversal filters, and the use of a set of transformed filter coefficients that are updated instead of the original ones.

Let us consider the channel-wise formulation of the LMS GSC adaptive algorithm outlined in Table 1. The input signal $x_i(n)$ is expressed in terms of real and imaginary parts as

$$x_i(n) = x_{\Re,i}(n) + \jmath x_{\Im,i}(n).$$

(Subscripts $\Re$ and $\Im$ denote the real and imaginary parts of a complex variable, respectively.) Consequently, the corresponding complex regressor vector $\mathbf{x}_i(n)$ is written as

$$\mathbf{x}_i(n) = \mathbf{x}_{\Re,i}(n) + \jmath \mathbf{x}_{\Im,i}(n), \tag{25}$$

where

$$\mathbf{x}_{\Re,i}(n) = \big[x_{\Re,i}(n)\ x_{\Re,i}(n-1)\ \ldots\ x_{\Re,i}(n-P+1)\big]^T, \tag{26}$$

$$\mathbf{x}_{\Im,i}(n) = \big[x_{\Im,i}(n)\ x_{\Im,i}(n-1)\ \ldots\ x_{\Im,i}(n-P+1)\big]^T. \tag{27}$$

In a similar way, the vector of complex filter coefficients $\mathbf{w}_i(n)$ is expressed as

$$\mathbf{w}_i(n) = \mathbf{w}_{\Re,i}(n) + \jmath \mathbf{w}_{\Im,i}(n). \tag{28}$$

Consider Eqn. (1) of Table 1. It can be equivalently written down as

$$y_i(n) = \mathbf{w}_i^H(n-1)\mathbf{x}_i(n) = \mathbf{x}_i^T(n)\mathbf{w}_i^*(n-1).$$

The application of the algorithmic strength reduction transformation, cf. Eqn. (24), to the above equation results in

$$y_i(n) = y_{\Re,i}(n) + \jmath y_{\Im,i}(n), \tag{29}$$

where

$$\begin{aligned} y_{\Re,i}(n) &= y_{i,1}(n) + 0.5 y_{i,3}(n), \\ y_{\Im,i}(n) &= y_{i,2}(n) + 0.5 y_{i,3}(n). \end{aligned} \tag{30}$$

The intermediate variables $y_{i,1}(n)$, $y_{i,2}(n)$ and $y_{i,3}(n)$ are estimated as follows:

$$\begin{aligned} y_{i,1}(n) &= \mathbf{c}_i^T(n-1)\mathbf{x}_{\Re,i}(n), \\ y_{i,2}(n) &= \mathbf{d}_i^T(n-1)\mathbf{x}_{\Im,i}(n), \qquad (31) \\ y_{i,3}(n) &= -2\mathbf{w}_{\Im,i}^T(n-1)\widehat{\mathbf{x}}_i(n) \\ &= -\left(\mathbf{c}(n-1) - \mathbf{d}(n-1)\right)^T \widehat{\mathbf{x}}_i(n). \qquad (32) \end{aligned}$$

Vectors $\mathbf{c}_i(n)$ and $\mathbf{d}_i(n)$ are transformed versions of the original filter coefficient vector $\mathbf{w}_i(n)$. They are defined as

$$\begin{aligned} \mathbf{c}_i(n) &= \mathbf{w}_{\Re,i}(n) + \mathbf{w}_{\Im,i}(n), \\ \mathbf{d}_i(n) &= \mathbf{w}_{\Re,i}(n) - \mathbf{w}_{\Im,i}(n). \end{aligned} \tag{33}$$

Here $\widehat{x}_i(n)$ is an auxiliary regressor defined as

$$\widehat{\mathbf{x}}_i(n) = \mathbf{x}_{\Re,i}(n) - \mathbf{x}_{\Im,i}(n). \tag{34}$$

Notice that the original filter $\mathbf{w}_i(n)$ can be recovered from the transformed parameters as

$$\begin{aligned} \mathbf{w}_{\Re,i}(n) &= 0.5\left(\mathbf{c}_i(n) + \mathbf{d}_i(n)\right), \\ \mathbf{w}_{\Re,i}(n) &= 0.5\left(\mathbf{c}_i(n) - \mathbf{d}_i(n)\right). \end{aligned} \tag{35}$$

Finally, the estimation error $e(n)$ takes the form

$$e(n) = e_{\Re}(n) + \jmath e_{\Im}(n), \tag{36}$$

where

$$e_{\Re}(n) = z_{\Re}(n) - y_{\Re}(n), \quad e_{\Im}(n) = z_{\Im}(n) - y_{\Im}(n). \tag{37}$$

Thus we may adapt the transformed filters $\mathbf{c}_i(n)$ and $\mathbf{d}_i(n)$ instead of $\mathbf{w}_i(n)$. Using Eqn. (4) of Table 1, we get the recursive equations of the form

$$\mathbf{c}_i(n) = \mathbf{c}_i(n-1) + \mu\left(e_{\Re}(n)\tilde{\mathbf{x}}_i(n) - e_{\Im}(n)\widehat{\mathbf{x}}_i(n)\right), \tag{38}$$

$$\mathbf{d}_i(n) = \mathbf{d}_i(n-1) + \mu\left(e_{\Re}(n)\widehat{\mathbf{x}}_i(n) + e_{\Im}(n)\tilde{\mathbf{x}}_i(n)\right), \tag{39}$$

where $\tilde{\mathbf{x}}_i(n) = \mathbf{x}_{\Re,i}(n) + \mathbf{x}_{\Im,i}(n)$. Then the complex vector

$$\mathbf{v}_i(n) = \mathbf{c}_i(n) + \jmath \mathbf{d}_i(n) \tag{40}$$

is updated as

$$\mathbf{v}_i(n) = \mathbf{v}_i(n-1) + \mu e(n)\left(\tilde{\mathbf{x}}_i(n) + \jmath \widehat{\mathbf{x}}_i(n)\right). \tag{41}$$

Finally, the application of the algorithmic strength reduction transform, Eqn. (24), to the above equation results in

$$\mathbf{c}_i(n) = \mathbf{c}_i(n-1) + \mu\left(\mathbf{g}_{i,1}(n) + \mathbf{g}_{i,3}(n)\right), \tag{42}$$

$$\mathbf{d}_i(n) = \mathbf{d}_i(n-1) + \mu\left(\mathbf{g}_{i,2}(n) + \mathbf{g}_{i,3}(n)\right), \tag{43}$$

where

$$\begin{aligned} \mathbf{g}_{i,1}(n) &= 2e_{\Re}(n)\mathbf{x}_{\Re,i}(n), \\ \mathbf{g}_{i,2}(n) &= 2e_{\Im}(n)\mathbf{x}_{\Re,i}(n), \\ \mathbf{g}_{i,3}(n) &= \hat{e}(n)\widehat{\mathbf{x}}_i(n), \end{aligned} \tag{44}$$

$$\hat{e}(n) = e_{\Re}(n) - e_{\Im}(n). \tag{45}$$

The SR LMS GSC adaptive algorithm is outlined in Table 3.

3.2. SR TD-LMS GSC Algorithm

Consider the application of the algorithmic strength reduction transform to the TD-LMS GSC adaptive algorithm. First, the computations involved in the Sliding Window DFT part should be accordingly re-organized. To this end, $f_{i,m+1}(n)$ is expressed in terms of real and imaginary parts as

$$f_{i,m+1}(n) = f^{\Re}_{i,m+1}(n) + \jmath f^{\Im}_{i,m+1}(n). \tag{46}$$

Similarly, the twiddle factor $e^{-\jmath\frac{2\pi m}{P}}$ is written as

$$e^{-\jmath\frac{2\pi m}{P}} = \cos\left(\frac{2\pi m}{P}\right) - \jmath\sin\left(\frac{2\pi m}{P}\right). \tag{47}$$

The application of the strength reduction transform implied by Eqn. (24) could reduce the number of multiplications required by the recursive estimation of $f_{i,m+1}(n)$. Since the twiddle factor is constant, we may compute the difference and the sum of its real and imaginary parts in advance. Thus, we are looking for a strength reduction transform where advantage of this fact is taken. Although the strength reduction transform defined by Eqn. (24) could by applied, there exists an alternative formulation that requires a lower number of additions. Actually, Eqn. (24) can be implemented in 16 different ways that require 3 multiplications and 5 additions (Perry *et al.*, 1999; Wenzler and Luder, 1995). The most suitable form in our case is the following:

$$\begin{aligned} ac - bd &= (a-b)d + (c-d)a, \\ ad + bc &= (a+b)c - (c-d)a. \end{aligned} \tag{48}$$

The application of the strength reduction transform, Eqn. (48), to Eqn. (2) of Table 2 results in the following component-wise set of recursions for $f_{i,m+1}(n)$:

$$\begin{aligned} f^{\Re}_{i,m+1}(n) = \rho&\left(\cos\left(\frac{2\pi m}{P}\right) + \sin\left(\frac{2\pi m}{P}\right)\right) f^{\Im}_{i,m+1}(n-1) \\ &+ \cos\left(\frac{2\pi m}{P}\right)\left(f^{\Re}_{i,m+1}(n-1)\right. \\ &\left. - f^{\Im}_{i,m+1}(n-1)\right) + u_{\Re,i}(n), \end{aligned} \tag{49}$$

$$\begin{aligned} f^{\Im}_{i,m+1}(n) = \rho&\left(\cos\left(\frac{2\pi m}{P}\right) - \sin\left(\frac{2\pi m}{P}\right)\right) f^{\Im}_{i,m+1}(n-1) \\ &- \cos\left(\frac{2\pi m}{P}\right)\left(f^{\Re}_{i,m+1}(n-1)\right. \\ &\left. - f^{\Im}_{i,m+1}(n-1)\right) + u_{\Im,i}(n). \end{aligned} \tag{50}$$

Since the first factor is common in both recursive equations described above, $f_{i,m+1}(n)$ can be efficiently estimated at a lower cost when compared with the original

Table 3. Computation flow chart of the proposed SR LMS GSC algorithm.

The SR LMS GSC algorithm			
Eqn.	Function	RMUL	RADD
	FOR $i=1$ TO $K-1$		
1	$\hat{x}_i(n)=x_{\Re,i}(n)-x_{\Im,i}(n)$	0	1
	$\mathbf{x}_{\Re,i}(n)=[x_{\Re,i}(n)\; x_{\Re,i}(n-1)\dots x_{\Re,i}(n-P+1)]^T$		
	$\mathbf{x}_{\Im,i}(n)=[x_{\Im,i}(n)\; x_{\Im,i}(n-1)\dots x_{\Im,i}(n-P+1)]^T$		
	$\widehat{\mathbf{x}}_i(n)=[\hat{x}_i(n)\; \hat{x}_i(n-1)\dots \hat{x}_i(n-P+1)]^T$		
2	$y_{i,1}(n)=\mathbf{c}_i^T(n-1)\mathbf{x}_{\Re,i}(n)$	P	$P-1$
3	$y_{i,2}(n)=\mathbf{d}_i^T(n-1)\mathbf{x}_{\Im,i}(n)$	P	$P-1$
4	$y_{i,3}(n)=-\left(\mathbf{c}_i(n-1)+\mathbf{d}_i(n-1)\right)^T\widehat{\mathbf{x}}_i(n)$	P	$2P-1$
5	$y_{\Re,i}(n)=y_{i,1}(n)+0.5y_{i,3}(n), \quad y_{\Im,i}(n)=y_{i,2}(n)+y_{i,3}(n)$	2	2
	END i		
7	$y_\Re(n)=\sum\limits_{i=1}^{K-1} y_{\Re,i}(n), \quad y_\Im(n)=\sum_{i=1}^{K-1} y_{\Im,i}(n)$	0	$2(K-2)$
8	$e_\Re(n)=d_\Re(n)-y_\Re(n), \quad e_\Im(n)=d_\Im(n)-y_\Im(n)$	0	2
9	$\hat{e}(n)=e_\Re(n)-e_\Im(n)$	0	1
	FOR $i=1$ TO $K-1$		
10	$\mathbf{g}_{i,1}(n)=2\mu e_\Re(n)\mathbf{x}_{\Re,i}(n)$	P	0
11	$\mathbf{g}_{i,2}(n)=2\mu e_\Im(n)\mathbf{x}_{\Re,i}(n)$	P	0
12	$\mathbf{g}_{i,3}(n)=\mu\hat{e}(n)\widehat{\mathbf{x}}_i(n)$	P	0
13	$\mathbf{c}_i(n)=\mathbf{c}_i(n-1)+(\mathbf{g}_{i,1}(n)+\mathbf{g}_{i,3}(n))$	0	$2P$
14	$\mathbf{d}_i(n)=\mathbf{d}_i(n-1)+(\mathbf{g}_{i,2}(n)+\mathbf{g}_{i,3}(n))$	0	$2P$
	END i		
	TOTAL COST	$6(K-1)P$	$8(K-1)P+2K-1$

implementation imposed by Eqn. (21). Thus, an auxiliary signal is introduced, as defined by

$$\hat{f}_{i,m+1}(n)=f^{\Re}_{i,m+1}(n)-f^{\Im}_{i,m+1}(n). \tag{51}$$

The complex regressor vector $\mathbf{f}_i(n)$ is expressed in terms of its real and imaginary parts as

$$\mathbf{f}_i(n)=\mathbf{f}_{\Re,i}(n)+\jmath\mathbf{f}_{\Im,i}(n), \tag{52}$$

where

$$\begin{aligned}\mathbf{f}_{\Re,i}(n)\; &= \left[f^{\Re}_{1,i}(n)\; f^{\Re}_{2,i}(n)\dots f^{\Re}_{P,i}(n)\right]^T,\\ \mathbf{f}_{\Im,i}(n)\; &= \left[f^{\Im}_{1,i}(n)\; f^{\Im}_{2,i}(n)\dots f^{\Im}_{P,i}(n)\right]^T.\end{aligned} \tag{53}$$

Similarly, the vector of transformed complex coefficients $\mathbf{W}_i(n)$ is written as

$$\mathbf{W}_i(n)=\mathbf{W}_{\Re,i}(n)+\jmath\mathbf{W}_{\Im,i}(n). \tag{54}$$

Equation (5) of Table 2 is equivalently written as

$$y_i(n)=\mathbf{f}_i^T(n)\mathbf{W}_i^*(n-1).$$

The subsequent application of the strength reduction transform, Eqn. (24), results in

$$y_i(n)=y_{\Re,i}(n)+\jmath y_{\Im,i}(n), \tag{55}$$

where

$$\begin{aligned}y_{\Re,i}(n)\; &= Y_{i,1}(n)+0.5Y_{i,3}(n),\\ y_{\Im,i}(n)\; &= Y_{i,2}(n)+0.5Y_{i,3}(n)\end{aligned} \tag{56}$$

and

$$\begin{aligned}Y_{i,1}(n)\; &= \mathbf{C}_i^T(n-1)\mathbf{f}_{\Re,i}(n),\\ Y_{i,2}(n)\; &= \mathbf{D}_i^T(n-1)\mathbf{f}_{\Im,i}(n),\end{aligned} \tag{57}$$

$$\begin{aligned}Y_{i,3}(n)\; &= -2\mathbf{W}^T_{\Im,i}(n-1)\widehat{\mathbf{f}}_i(n)\\ &= -\left(\mathbf{C}_i(n-1)+\mathbf{D}_i(n-1)\right)^T.\end{aligned} \tag{58}$$

The auxiliary parameters $\mathbf{C}_i(n)$ and $\mathbf{D}_i(n)$ are defined as

$$\begin{aligned}\mathbf{C}_i(n)\; &= \mathbf{W}_{\Re,i}(n)+\mathbf{W}_{\Im,i}(n),\\ \mathbf{D}_i(n)\; &= \mathbf{W}_{\Re,i}(n)-\mathbf{W}_{\Im,i}(n).\end{aligned} \tag{59}$$

The regressor vector $\hat{\mathbf{f}}_i(n)$ introduced in Eqn. (59) is defined as

$$\widehat{\mathbf{f}}_i(n) = \mathbf{f}_{\Re,i}(n) - \mathbf{f}_{\Im,i}(n). \tag{60}$$

Obviously, the original parameters can be recovered as

$$\begin{aligned} \mathbf{W}_{\Re,i}(n) &= 0.5\left(\mathbf{C}_i(n)+\mathbf{D}_i(n)\right), \\ \mathbf{W}_{\Re,i}(n) &= 0.5\left(\mathbf{C}_i(n)-\mathbf{D}_i(n)\right). \end{aligned} \tag{61}$$

Finally, the estimation error $e(n)$ is computed as

$$e(n) = e_{\Re}(n) + \jmath e_{\Im}(n), \tag{62}$$

$$e_{\Re}(n) = z_{\Re}(n) - y_{\Re}(n), \quad e_{\Im}(n) = z_{\Im}(n) - y_{\Im}(n). \tag{63}$$

Thus, we may adapt parameters $\mathbf{C}_i(n)$ and $\mathbf{D}_i(n)$ instead of using $\mathbf{W}_i(n)$. Applying the above analysis, we get

$$\begin{aligned} \mathbf{C}_i(n) &= \mathbf{C}_i(n-1) \\ &\quad + \mu \mathbf{P}_i^{-1}\big(e_{\Re}(n)\tilde{\mathbf{f}}_i(n) - e_{\Im}(n)\widehat{\mathbf{f}}_i(n)\big), \end{aligned} \tag{64}$$

$$\begin{aligned} \mathbf{D}_i(n) &= \mathbf{D}_i(n-1) \\ &\quad + \mu \mathbf{P}_i^{-1}\big(e_{\Re}(n)\widehat{\mathbf{f}}_i(n) + e_{\Im}(n)\tilde{\mathbf{f}}_i(n)\big), \end{aligned} \tag{65}$$

where $\tilde{\mathbf{f}}_i(n) = \mathbf{f}_{\Re,i}(n) + \mathbf{f}_{\Im,i}(n)$. Grouping together the above recursions, a single complex valued recursion is obtained:

$$\mathbf{V}_i(n) = \mathbf{V}_i(n-1) + \mu e(n)\mathbf{P}_i^{-1}\big(\tilde{\mathbf{f}}_i(n) + \jmath\widehat{\mathbf{f}}_i(n)\big), \tag{66}$$

where $\mathbf{V}_i(n) = \mathbf{C}_i(n) + \jmath \mathbf{D}_i(n)$. Finally, the application of the algorithmic strength reduction transform, Eqn. (24), results in

$$\mathbf{C}_i(n) = \mathbf{C}_i(n-1) + \mu \mathbf{P}_i^{-1}\left(\mathbf{G}_{i,1}(n) + \mathbf{G}_{i,3}(n)\right), \tag{67}$$

$$\mathbf{D}_i(n) = \mathbf{D}_i(n-1) + \mu \mathbf{P}_i^{-1}\left(\mathbf{G}_{i,2}(n) + \mathbf{G}_{i,3}(n)\right), \tag{68}$$

where

$$\begin{aligned} \mathbf{G}_{i,1}(n) &= 2e_{\Re}(n)\mathbf{f}_{\Re,i}(n), \\ \mathbf{G}_{i,2}(n) &= 2e_{\Im}(n)\mathbf{f}_{\Re,i}(n) \end{aligned} \tag{69}$$

$$\begin{aligned} \mathbf{G}_{i,3}(n) &= \hat{e}(n)\widehat{\mathbf{f}}_i(n), \\ \hat{e}(n) &= e_{\Re}(n) - e_{\Im}(n). \end{aligned} \tag{70}$$

The SR TD-LMS GSC adaptive algorithm is outlined in Table 4.

The GSC adaptive beamforming scheme discussed so far assumes perfect knowledge of the steering vector parameters. However, the performance of the GSC may be substantially deteriorated if the steering vector specification does not match the true signal environments (Cox *et al.*, 1987; Feldman and Griffiths, 1994). This type of imperfection occurs either as a result of *calibration errors*, or due to *pointing errors*, or a combination of both of them. A remedy to the aforementioned problem is the utilization of adaptation methods that improve the robustness to perturbations in the covariance matrix and/or the steering vector. Several robust algorithmic schemes were proposed that improve the robustness of the GSC adaptive beamformer, applying techniques such as regularization (Cox *et al.*, 1987; Gershman, 1999), iterative projection (Feldman and Griffiths, 1994), quadratic constraints optimization (Tian *et al.*, 2001), H_∞ minimum estimation criterion (Chang and Chiang, 2002), joint adaptive estimation-calibration methods (Fudge and Linebarger, 1994; Hoshuyama *et al.*, 1999; Gannot *et al.*, 2001; Ye *et al.*, 1997), etc. Since robust adaptive GSC algorithms have a structure similar to that of the standard LMS schemes of Tables 1 and 2, the application of the proposed strength reduction technique to robust GSC methods is feasible, which results in a reduction in the arithmetic complexity of the original schemes.

3.3. Complexity Assessment

The computational complexity of the LMS GSC algorithm of Table 2 is given by

$$C_{LMS}^{\text{classical}} = 2(K-1)P\,\text{CMUL} + 2(K-1)P\,\text{CADD}, \tag{71}$$

where CMUL and CADD denote the complex valued multiplication and the complex valued addition, respectively. Multiplication by μ can be realized by a digit shift operator, since μ can be chosen to be a power of two.

The computational complexity of the TD-LMS GSC scheme listed in Table 2 is

$$\begin{aligned} C_{TD-LMS}^{\text{classical}} = 3(K-1)P\,&\text{CMUL} \\ + (K-1)(3P+1)\,&\text{CADD} \\ + 4(K-1)P\,&\text{RMUL} \\ + 2(K-1)P\,&\text{RADD}, \end{aligned} \tag{72}$$

where RMUL and RADD denote the real valued multiplication and the real valued addition, respectively. Multiplication by μ and λ is ignored, since they can be realized by a digit shift operator (both of μ and λ can be chosen to be a power of two).

The direct implementation of the fast complex valued multiplication method in the LMS GSC and TD-LMS GSC schemes can reduce the number of real multiplications, at the expense of an increased number of adders, cf. Eqn. (71) that describes the complexity of the original LMS GSC adaptive algorithm. The equivalent algorithm complexity in terms of real valued computations is estimated using the fact that the classical complex valued

Table 4. Computation flow chart of the proposed SR TD LMS GSC algorithm.

	SR TD LMS GSC algorithm		
	Function	RMUL	RADD
	FOR $i = 1$ TO $K-1$		
	$\widehat{K}_m = \rho\left(\cos(\frac{2\pi m}{P}) - \sin(\frac{2\pi m}{P})\right), \quad \tilde{K}_m = \rho\left(\cos(\frac{2\pi m}{P}) + \sin(\frac{2\pi m}{P})\right)$		
1	$u_{\Re,i}(n) = x_{\Re,i}(n) - \rho^P x_{\Re,i}(n-P)$	0	1
2	$u_{\Im,i}(n) = x_{\Im,i}(n) - \rho^P x_{\Im,i}(n-P)$	0	1
	FOR $m = 0$ TO $P-1$		
3	$\widehat{f}_{i,m+1}(n) = f^{\Re}_{i,m+1}(n) - f^{\Im}_{i,m+1}(n)$	0	1
4	$f_{i,m+1,1}(n) = \tilde{K}_{m+1}\widehat{f}_{i,m+1}(n-1) + u_{\Re,i}(n)$	1	1
5	$f_{i,m+1,2}(n) = \rho\cos(\frac{2\pi m}{P})\widehat{f}_{i,m+1}(n-1) + u_{\Im,i}(n)$	1	1
6	$f_{i,m+1,3}(n) = \widehat{K}_m f^{\Im}_{i,m+1}(n-1)$	1	0
7	$f^{\Re}_{i,m+1}(n) = f_{i,m+1,1}(n) + f_{i,m+1,3}(n)$	0	1
8	$f^{\Im}_{i,m+1}(n) = f_{i,m+1,2}(n) - f_{i,m+1,3}(n)$	0	1
9	$p_{i,m+1}(n) = \lambda p_{i,m+1}(n-1) + (1-\lambda)\left((f^{\Re}_{i,m+1}(n))^2 + (f^{\Im}_{i,m+1}(n))^2\right)$	2	2
10	$F^{\Re}_{i,m+1}(n) = \dfrac{f^{\Re}_{i,m+1}(n)}{p_{i,m+1}(n)}, \quad F^{\Im}_{i,m+1}(n) = \dfrac{f^{\Im}_{i,m+1}(n)}{p_{i,m+1}(n)}$	2	0
11	$\widehat{F}^{\Re}_{i,m+1}(n) = F^{\Re}_{i,m+1}(n) - F^{\Im}_{i,m+1}(n)$	0	1
	END m		
	$\mathbf{f}_{\Re,i}(n) = [f^{\Re}_{i,1}(n)\ f^{\Re}_{i,2}(n) \ldots f^{\Re}_{i,P}(n)]^T$		
	$\mathbf{f}_{\Im,i}(n) = [f^{\Im}_{i,1}(n)\ f^{\Im}_{i,2}(n) \ldots f^{\Im}_{i,P}(n)]^T$		
	$\widehat{\mathbf{f}}_i(n) = [\widehat{f}_{i,1}(n)\ \widehat{f}_{i,2}(n) \ldots \widehat{f}_{i,P}(n)]^T$		
	$\mathbf{F}_{\Re,i}(n) = [F^{\Re}_{i,1}(n)\ F^{\Re}_{i,2}(n) \ldots F^{\Re}_{i,P}(n)]^T$		
	$\mathbf{F}_{\Im,i}(n) = [F^{\Im}_{i,1}(n)\ F^{\Im}_{i,2}(n) \ldots F^{\Im}_{i,P}(n)]^T$		
	$\widehat{\mathbf{F}}_i(n) = [\widehat{F}_{i,1}(n)\ \widehat{F}_{i,2}(n) \ldots \widehat{F}_{i,P}(n)]^T$		
12	$Y_{i,1}(n) = \mathbf{C}_i^T(n-1)\mathbf{f}_{\Re,i}(n)$	P	$P-1$
13	$Y_{i,2}(n) = \mathbf{D}_i^T(n-1)\mathbf{f}_{\Im,i}(n)$	P	$P-1$
14	$Y_{i,3}(n) = -\left(\mathbf{C}_i(n-1) + \mathbf{D}_i(n-1)\right)^T \widehat{\mathbf{f}}_i(n)$	P	$2P-1$
15	$y_{\Re,i}(n) = Y_{i,1}(n) + Y_{i,3}(n), \quad y_{\Im,i}(n) = Y_{i,2}(n) + Y_{i,3}(n)$	0	2
	END i		
16	$y_{\Re}(n) = \sum_{i=1}^{K-1} y_{\Re,i}(n), \quad y_{\Im}(n) = \sum_{i=1}^{K-1} y_{\Im,i}(n)$	0	$2(K-2)$
17	$e_{\Re}(n) = d_{\Re}(n) - y_{\Re}(n), \quad e_{\Im}(n) = d_{\Im}(n) - y_{\Im}(n)$	0	2
18	$\hat{e}(n) = e_{\Re}(n) - e_{\Im}(n)$	0	1
	FOR $i = 1$ TO $K-1$		
19	$\mathbf{G}_{i,1}(n) = 2e_{\Re}(n)\mathbf{F}_{\Re,i}(n), \quad \mathbf{G}_{i,2}(n) = 2e_{\Im}(n)\mathbf{F}_{\Re,i}(n)$	$2P$	0
20	$\mathbf{G}_{i,3}(n) = \hat{e}(n)\widehat{\mathbf{F}}_i(n)$	P	0
21	$\mathbf{C}_i(n) = \mathbf{C}_i(n-1) + \mu\left(\mathbf{G}_{i,1}(n) + \mathbf{G}_{i,3}(n)\right)$	0	$2P$
22	$\mathbf{D}_i(n) = \mathbf{D}_i(n-1) + \mu\left(\mathbf{G}_{i,2}(n) + \mathbf{G}_{i,3}(n)\right)$	0	$2P$
	END i		
	TOTAL COST	$13P(K-1)$	$16P(K-1) + 3(K-1) + 1$

multiplication method requires 4 RMULs and 2 RADDs, while the complex valued addition can be performed by 2 RADDs. Thus, we get

$$C_{LMS}^{\text{classical}} = 8(K-1)P\,\text{RMUL} + 8(K-1)P\,\text{RADD}. \tag{73}$$

On the other hand, if the direct fast complex valued multiplication is applied (3 RMULs and 5 RADDs), we get

$$\begin{aligned} C_{LMS}^{\text{direct}\,SR} &= 6(K-1)P\,\text{RMUL} \\ &\quad + 14(K-1)P\,\text{RADD}. \end{aligned} \tag{74}$$

Comparing Eqns. (73) and (74), we conclude that the direct application of the fast complex valued multiplication reduces the number of real valued multiplications by 25%, at the expense of 50% of extra real valued adders.

The computational complexity of the proposed SR LMS GSC adaptive algorithm of Table 3 is

$$\begin{aligned} C_{LMS}^{\text{proposed}\,SR} &= 6(K-1)P\ \text{RMUL} \\ &\quad + (8(K-1)P+2K-1)\ \text{RADD}. \end{aligned} \tag{75}$$

Comparing Eqns. (75) and (72) giving the equivalent real valued arithmetic complexity of the LMS GSC algorithm, it is established that the proposed SR LMS GSC adaptive algorithm requires 25% fewer real valued multiplications. Moreover, the number of real valued additions remains approximately the same as the original scheme (this follows from the fact that $(2K-1) \ll 8(K-1)P$).

In a similar way, the TD-LMS GSC algorithm has an equivalent real valued arithmetic complexity using the classical or direct fast complex valued multiplication methods respectively given by

$$\begin{aligned} C_{TD-LMS}^{\text{classical}} &= 16(K-1)P\ \text{RMUL} \\ &\quad + (14P+2)(K-1)\ \text{RADD} \end{aligned} \tag{76}$$

and

$$\begin{aligned} C_{TD-LMS}^{\text{direct}\,SR} &= 13(K-1)P\,\text{RMUL} \\ &\quad + (23P+2)(K-1)\,\text{RADD}. \end{aligned} \tag{77}$$

In this case, the direct use of the fast complex valued multiplication method results in a 20% reduction in the number of real valued multipliers, at the expense of about 80% of extra real valued additions.

The computational complexity of the proposed SR TD-LMS GSC adaptive algorithm of Table 4 is given by

$$\begin{aligned} C_{TD-LMS}^{\text{proposed}\,SR} &= 13(K-1)P\ \text{RMUL} \\ &\quad + (16(K-1)P+3K+1)\ \text{RADD}. \end{aligned} \tag{78}$$

Comparing Eqns. (78) and (72) of the LMS GSC algorithm, it is clear that the proposed SR TD-LMS GSC adaptive algorithm requires fewer 20% real valued multiplications at the expense of 20% of extra real valued additions.

A comparison of the relative computational requirements among all LMS GSC type algorithms discussed above is provided below:

Realization of the LMS GSC

Algorithm	Relative RMUL	Relative RADD
Classic	100%	100%
Direct SR	75%	120%
Proposed SR	75%	100%

Realization of the TD-LMS GSC

Algorithm	Relative RMUL	Relative RADD
Classic	100%	100%
Direct SR	80%	180%
Proposed SR	80%	120%

3.4. Pipelined Implementation Aspects

In many beamforming applications, very high sample rates and/or multi-input linear combiners of large size are required. Electronically phased-array radars and digital beamforming in wireless communications are typical examples where very high sampling rates are applied. The computational burden of the adaptation mechanism of the GSC beamformer becomes extremely high; the processing power of several GOPS (Giga operations per second) is needed (Ahlander *et al.*, 1996; Boukalov and Haggman, 2000; Martinez, 1999; Taveniku and Ahlander, 1997). This means that for a real-time implementation, pipelined and/or parallel computational architectures should be developed. Moreover, pipelining and parallelism in the computation mechanism can be used to design low power implementations, which are necessary for power constrained applications (Parhi, 1999).

The inner product computations involved in the error feedback loop of both SR LMS GSC and SR TD-LMS GSC algorithms, i.e., Eqns. (2)–(4) and (12)–(14) of Tables 3 and 4, respectively, prohibit the full pipelining and/or parallelism of the pertinent algorithms. A remedy to this bottleneck is the introduction of an adaptation delay (Long *et al.*, 1989; 1992) resulting in delayed LMS GSC schemes. The development of pipelined adaptive LMS and TD-LMS GSC schemes is accomplished by introducing a certain amount of delay in the adaptation mechanism. Thus, the filter updating equations, i.e., Eqns. (13)

and (14) of Table 3, and Eqns. (21) and (22) of Table 4, are replaced by the delayed updating versions as

$$\begin{aligned}\mathbf{c}_i(n) &= \mathbf{c}_i(n-1) \\ &\quad + \mu\left(\mathbf{g}_{i,1}(n-\Delta_1) + \mathbf{g}_{i,3}(n-\Delta_1)\right),\end{aligned} \tag{79}$$

$$\begin{aligned}\mathbf{d}_i(n) &= \mathbf{d}_i(n-1) \\ &\quad + \mu\left(\mathbf{g}_{i,2}(n-\Delta_1) + \mathbf{g}_{i,3}(n-\Delta_1)\right)\end{aligned} \tag{80}$$

and

$$\begin{aligned}\mathbf{C}_i(n) &= \mathbf{C}_i(n-1) \\ &\quad + \mu\big(\mathbf{G}_{i,1}(n-\Delta_2) + \mathbf{G}_{i,3}(n\Delta_2)\big),\end{aligned} \tag{81}$$

$$\begin{aligned}\mathbf{D}_i(n) &= \mathbf{D}_i(n-1) \\ &\quad + \mu\big(\mathbf{G}_{i,2}(n-\Delta_2) + \mathbf{G}_{i,3}(n-\Delta_2)\big),\end{aligned} \tag{82}$$

respectively. A proper retiming of the delays introduced in the error feedback loop allows for the development of high throughput pipelineable and/or parallel schemes for the implementation of both of the proposed LMS GSC algorithms on ASIC VLSI systolic or wavefront array processors. The exact size of the adaptation delays Δ_1 and Δ_2 depends on the pipelined architecture adopted. Roughly speaking, Δ_1 and Δ_2 are $O(\log_2(P) + \log_2(K))$ when binary tree adders are utilized for the estimation of additions involved in Eqns. (2)–(4) and (7) and Eqns. (12)–(14) and (16) of Tables 3 and 4, respecively.

3.5. Power Dissipation and the Silicon Area

The strength reduction transformation applied to the LMS GSC and TD-LMS GSC algorithms results in a lower power dissipation and silicon area characteristics of the fabricated ASIC VLSI or the FPGA implementation of the pertinent adaptive beamforming algorithm for real time applications. The dynamic power dissipation P_D in the CMOS technology depends mainly on the following three factors: (a) the average capacitance being switched, C_L, (b) the frequency of operation, f, and (c) the supply voltage, V_{dd}, (Chandrakasan and Brodersen, 1995; Shanbag, 1998). Indeed, the following equation holds:

$$P_D = C_L f V_{dd}^2. \tag{83}$$

The application of the strength reduction transform at either the algorithmic or the architectural level results in a low power dissipation by reduction of arithmetic operations, which corresponds to the reduction of the average capacitance being switched, C_L.

We will now derive a fairly accurate estimate of the power savings achieved by the proposed strength reduced LMS GSC and TD-LMS CSC algorithms. Let us assume that the effective capacitance, C_L, and the occupied area on silicon, A, of a two-operand multiplier are proportional to that of a two-operand adder, i.e., $C_L^{RMUL} \propto K_P C_L^{RADD}$ and $A^{RMUL} \propto K_A A^{RADD}$ (Shanbhag, 1998). Then the power saving factor, PE, due to the application of the strength reduction transform can be expressed as

$$PE(K_P) = 1 - \frac{P_D^{\text{classical}}(K_P)}{P_D^{SR}(K_P)}, \tag{84}$$

The power savings achieved by the proposed SR LMS GSC algorithm are estimated by taking into account Eqns. (73) and (75) as

$$PE_{LMS}^{\text{proposed}\,SR} = \frac{K_P}{4(K_P+1)}. \tag{85}$$

When the fast complex multiplication is directly applied to the original LMS GSC algorithm, the PE is estimated using Eqns. (73) and (74) as follows:

$$PE_{LMS}^{\text{direct}\,SR} = \frac{K_P - 3}{4(K_P+1)}. \tag{86}$$

Using similar arguments as above, the area savings achieved by both schemes are estimated as

$$\begin{aligned}AE_{LMS}^{\text{proposed}\,SR} &= \frac{K_A}{4(K_A+1)}, \\ AE_{LMS}^{\text{direct}\,SR} &= \frac{K_A - 3}{4(K_A+1)}.\end{aligned} \tag{87}$$

From (85)–(87) it is clear that both schemes result in power and area savings that approach the 25% limit as K_P and K_A become large. If we assume array based multiplier structures, then both K_P and K_A are approximately equal to the number of bits n_b that are used for the digital number representation. The power savings achieved by both schemes are illustrated in Fig. 3.

Clearly, the proposed SR-LMS GSC scheme outperforms the classical fast multiplication LMS GSC implementation, since it approaches the theoretical limit of 25% savings much faster than the classical scheme, even for small values of K_P and K_A.

The power and area saving achieved by the proposed strength reducing transform domain LMS GSC are estimated in a similar way, using Eqns. (76)–(78). Thus we get

$$\begin{aligned}PE_{TD-LMS}^{\text{proposed}\,SR} &= \frac{3K_P - 2}{16K_P + 14}, \\ AE_{TD-LMS}^{\text{proposed}\,SR} &= \frac{3K_A - 2}{16K_A + 14},\end{aligned} \tag{88}$$

$$\begin{aligned}PE_{TD-LMS}^{\text{direct}\,SR} &= \frac{3K_P - 9}{16K_P + 14}, \\ AE_{TD-LMS}^{\text{direct}\,SR} &= \frac{3K_A - 9}{16K_A + 14}.\end{aligned} \tag{89}$$

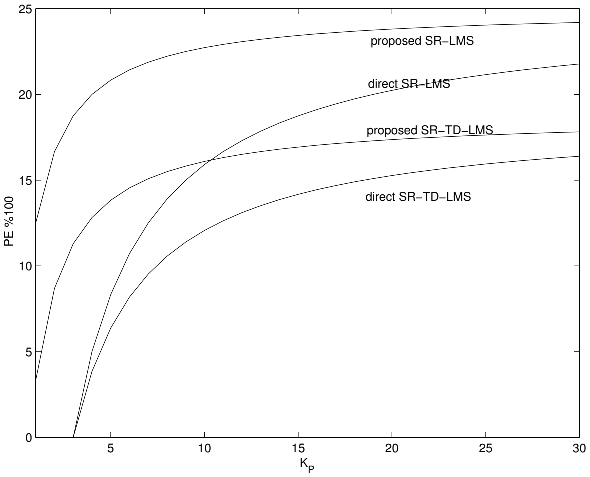


Fig. 2. Power and area savings achieved by the proposed SR LMS GSC and the SR TD LMS GSC algorithms.

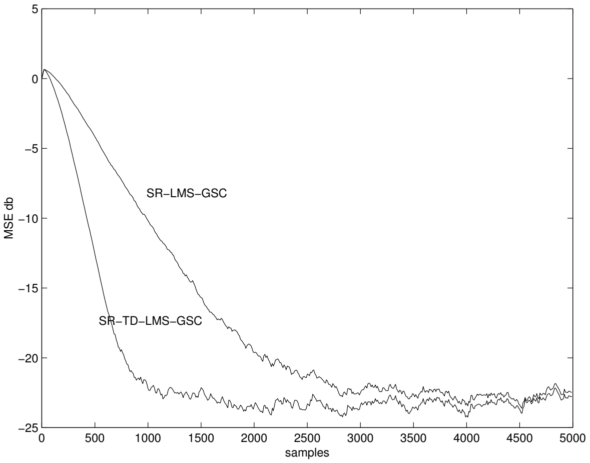


Fig. 3. Experiment 1. Learning curves (MSE): (a) the SR LMS GSC, and (b) the SR TD LMS GSC.

Clearly, both algorithms result in power and area savings that approach the 18.75% limit as K_C and K_A become large. The power savings achieved by both the schemes are illustrated in Fig. 3. The proposed SR-TD LMS GSC algorithm outperforms the classical fast TD LMS GSC counterpart, since it approaches the theoretical limit of 18.75% savings much faster than the classical scheme, even for small values of K_P and K_A (notice that for the direct SR TD LMS GSC scheme savings appear when $K_P > 4$).

4. Simulation

The performance of the proposed SR LMS GSC and SR TD-LMS GSC algorithms is illustrated by two typical adaptive beamforming examples. The simulation scenario for the first experiment is adopted from (An and Champagne, 1994). A stationary narrowband target signal, $s_T(n)$, is mitigated by three stationary narrowband jammers $s_{j1}(n)$, $s_{j2}(n)$ and $s_{j3}(n)$, with different directions of arrival. The background noise, $\eta(n)$, is a zero mean Gaussian white noise signal. Thus

$$v_1(n) = s_T(n) + s_{j1}(n) + s_{j2}(n) + s_{j3}(n) + \eta(n).$$

Specific values of the simulation parameters are given below:

Signal		f	θ	SNR
Target	$s_T(n)$	0.1	0°	10 db
Jammer 1	$s_{j1}(n)$	0.3	34°	20 db
Jammer 2	$s_{j2}(n)$	0.4	-49°	40 db
Jammer 3	$s_{j3}(n)$	0.25	-24°	30 db

The parameters f and θ denote the normalized frequency and the incident angle (relative to the broadside) of the plane wave signals, respectively. The adaptive beamformer consists of $K = 17$ linear array elements, equally spaced at half of the wavelength distance at the maximum frequency of interest, $f_{\max}$, and it is steered in the direction of the target signal. Seven delay elements are associated with each array element, i.e., $P = 8$.

The MSE between the actual target signal, $s_T(n)$, and the output of the beamformer $e(n)$, i.e.,

$$\mathcal{P}_e(n) = \boldsymbol{\mathcal{E}}(|s_T(n) - e(n)|^2)$$

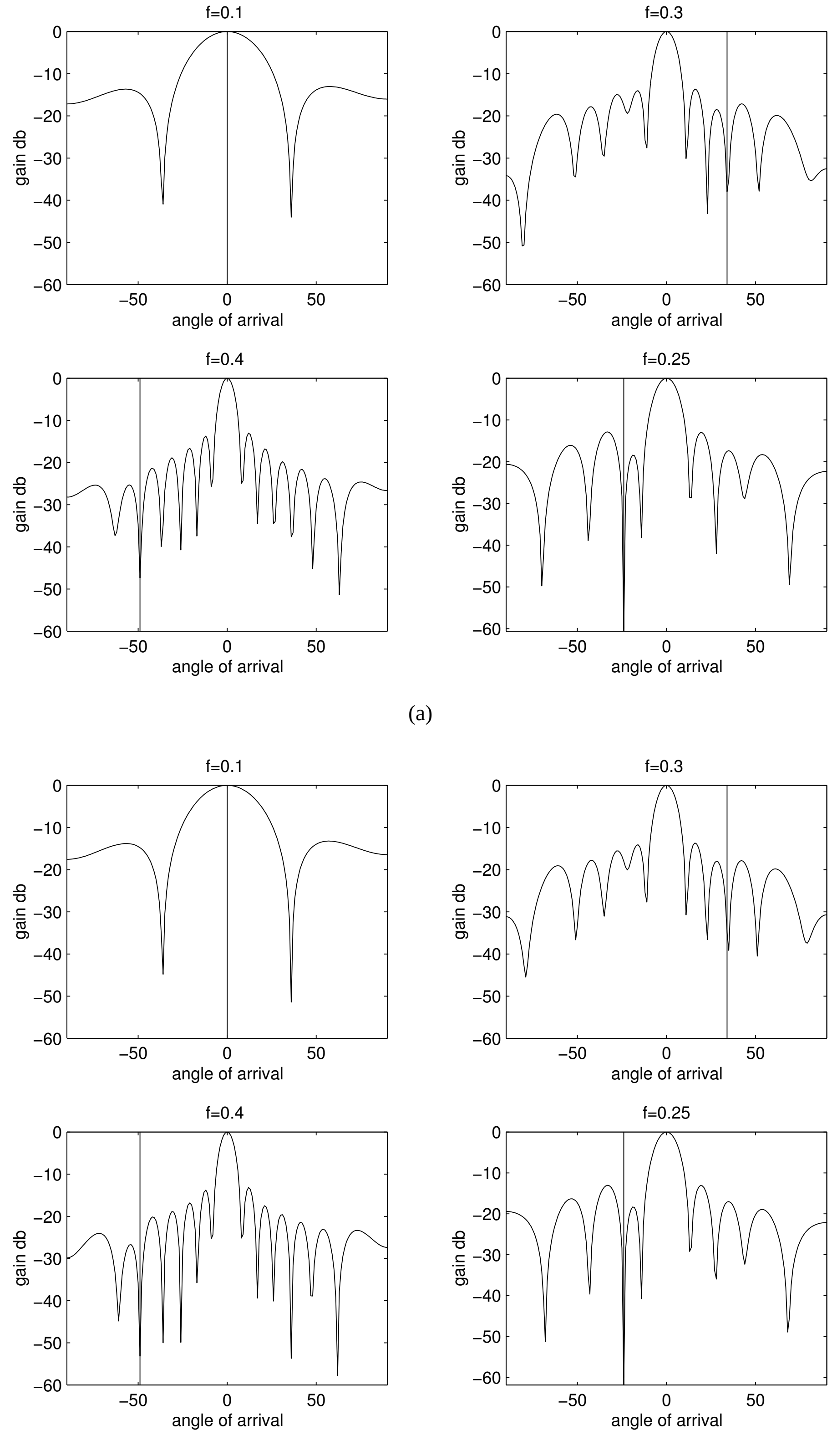
for each case, was computed by averaging the squared instantaneous estimation errors over an exponentially decaying window with the effective memory equal to 100 time instants. The learning curves of the SR LMS GSC and SR TD-LMS GSC algorithms are depicted in Fig. 3. The beampatterns of the proposed method (after convergence) at different frequencies of interest are depicted in Fig. 4.

The simulation scenario for the second experiment is similar to that of (Weiss *et al.*, 1999). A stationary wideband target signal, $s_T(n)$, is mitigated by a stationary wideband jammer, $s_{j1}(n)$. The background noise, $\eta(n)$, is a zero mean Gaussian white noise signal. Thus

$$v_1(n) = s_T(n) + s_{j1}(n) + \eta(n).$$

Specific values of the simulation parameters are given below:

Signal		f	θ	SNR
Target	$s_T(n)$	[0,0.5]	0°	5 db
Jammer 1	$s_{j1}(n)$	[0.1,0.35]	-20°	40 db



(a)

(b)

Fig. 4. Experiment 1. Beampatterns of the SR LMS GSC (a) and the SR TD LMS GSC (b) after convergence at different frequencies of interest.

In this case, the adaptive beamformer consists of $K = 11$ linear array elements, evenly spaced at half of the wavelength distance at the maximum frequency of interest, $f_{\max}$, and it is steered in the direction of the target signal. One hundred delay elements are associated with each array elements, i.e., $P = 100$. The learning curves of the SR LMS GSC and SR TD-LMS GSC algorithms are depicted in Fig. 5. The beampatterns of the proposed methods (after convergence) at different frequencies of interest are depicted in Fig. 6.

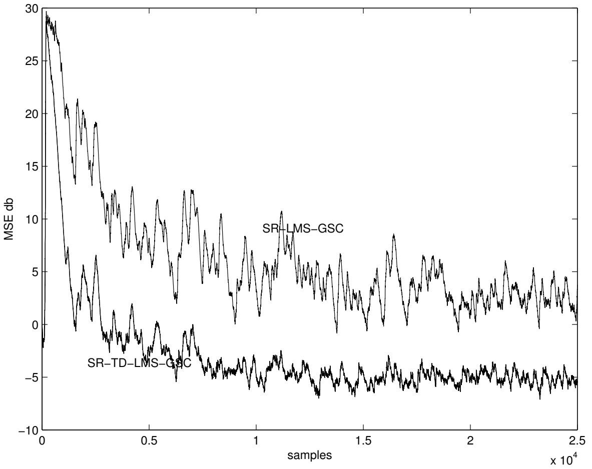
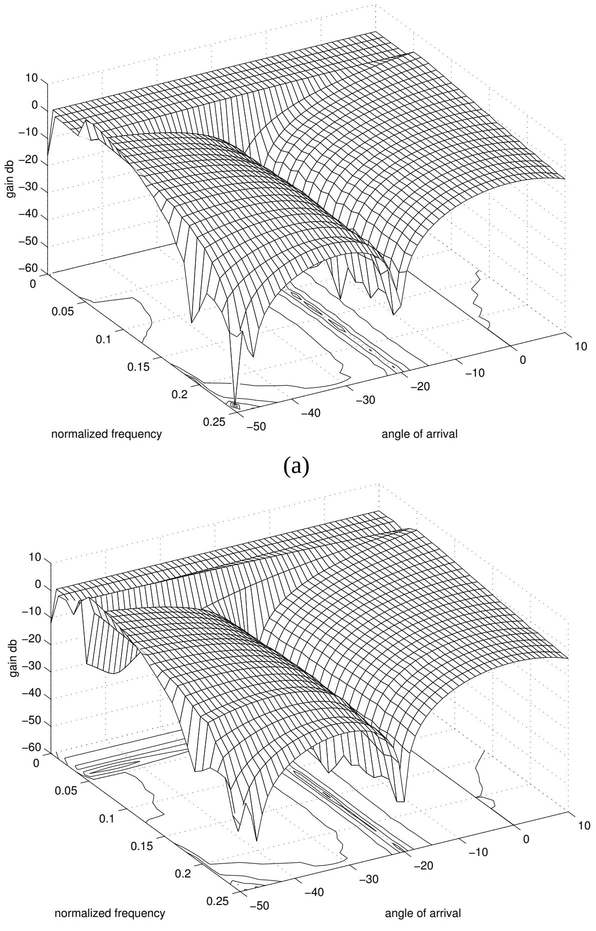


Fig. 5. Experiment 2. Learning curves (MSE): (a) the SR LMS GSC, and (b) the SR TD LMS GSC.

From both experiments it is clear that the proposed SR LMS GSC and SR TD LMS GSC algorithms provide efficient adaptive beamforming with reduced computational complexity, and reduced power dissipation and silicon area requirements.



(a)

(b)

Fig. 6. Experiment 2. Beampatterns of the SR LMS GSC (a) and the SR TD LMS GSC (b) after convergence.

5. Conclusions

In this paper, efficient realizations of least mean squared error, generalized sidelobe canceller algorithms applied to adaptive beamforming have been considered. Two efficient algorithms for adaptive GSCs were developed for implementing the LMS GSC and transform domain LMS GSC schemes. In both cases, the strength reduction transforms that reduce the number of operations were applied without affecting the input-output behavior of the underline algorithms. Using the algorithmic strength reduction, low complexity LMS GSC and TD-LMS GSC algorithms were developed that can be realized using real valued arithmetic only, whilst reducing the number of real valued multipliers by 25% and 20%, respectively, at a zero or a marginal increase in the number or real valued adders. The proposed schemes are suitable for VLSI implementation as they assure a significant reduction in the power dissipation and the silicon area of the fabricated circuit. The performance of the proposed realizations of the LMS based GSC methods was illustrated in the context of beamforming applications.

References

Ahlander A., Taveniku M. and Svensson B. (1996): *A multiple SIMD approach to radar signal processing*. — Proc. 10-th IEEE Region. Conf., Perth, Australia, pp. 852–857.

An J. and Champagne B. (1994): *GSC realization using the two-dimensional transform-domain LMS algorithm*. — IEE Proc. Radar Sonar Navig., Vol. 141, No. 5, pp. 270–278.

Baghaie R. (1999): *Applications of transformation techniques in CDMA receivers*. — Proc. 42-nd IEEE Midwest Symp. *Circuits and Systems'99*, Conf., Las Cruces, Mexico, pp. 905–908.

Baghaie R. and Laakso T. (1998): *Implementation of low-power CDMA RAKE receivers using strength reduction transformation*. — Proc. IEEE Nordic Signal Processing Symp., Aalborg, Denmark, pp. 169–172.

Baillet S., Moscher J. and Leahy R. (2001): *Electromagnetic brain mapping*. — IEEE Signal Process. Mag., Vol. 18, No. 2, pp. 14–30.

Boukalov A. and Haggman S. (2000): *System aspects of smart-antenna technology in cellular wireless communications—An overview*. — IEEE Trans. Micr. Theory Techn., Vol. 48, No. 6, pp. 919–929.

Buckley K. (1986): *Broadband beamforming and the generalized sidelobe canceller*. — IEEE Trans. Acoust. Speech Signal Process., Vol. 34, No. 5, pp. 1322–1323.

Buckley K. (1987): *Spacial/spectral filtering with linearly constrained minimum variance beamformers*. — IEEE Trans. Acoust. Speech Signal Process., Vol. 35, No. 3, pp. 249–266.

Chang A. and Chiang C. (2002): *Adaptive H_∞ robust beamforming for imperfect antenna array*. — Signal Process., Vol. 82, No. 8, pp. 1183–1188.

Chandrakasan A. and Brodersen R. (1995): *Minimizing power consumption in digital CMOS circuits*. — Proc. IEEE, Vol. 83, No. 4, pp. 498–523, 12–31.

Chen Y.H. and Fang H.D. (1992): *Frequency domain implementation of the Griffiths-Jim adaptive beamformer*. — J. Acoust. Soc. Am., Vol. 91, No. 6, pp. 3354–3366.

Chryssomallis M. (2000): *Smart antennas*. — IEEE Antenn. Propag. Mag., Vol. 42, No. 3, pp. 129–136.

Chu Y. and Fang W. (1999): *A novel wavelet-based generalized sidelobe canceller*. — IEEE Antenn. Propag., Vol. 47, No. 9, pp. 1485–1494.

Compton R. (1988): *Adaptive Antennas: Concepts and Applications*. — Englewood Cliffs, NJ: Prentice-Hall.

Cox H., Zeskind R. and Owen M. (1987): *Robust adaptive beamforming*. — IEEE Trans. Acoust. Speech Sigal. Process., Vol. 35, No. 9, pp. 1365–1376.

Drabowitch S., Papiernik A., Griffiths H., Encinas J. and Smith B. (1998): *Modern Antennas*. — Chapman & Hall.

Farina R. (1992): *Antenna-Based Signal Processing Techniques for Radar Systems*. — Norwood, MA: Artech House.

Farina A., Saverione A. and Timmoneri, L., (1996): *MVDR vectorial lattice applied to space-time processing for AEW radar with large instantaneous bandwidth*. — IEE Proc. Radar Sonar Navig., Vol. 143, No. 1, pp. 41–46.

Farina A. and Timmoneri L. (1999): *Real-time STAP techniques*. — Electron. Comm. Eng. J., Vol. 11, No. 1, pp. 13–22.

Feldman D. and Griffiths L. (1994): *A projection approach for robust adaptive beamforming*. — IEEE Trans. Signal Process., Vol. 42, No. 4, pp. 867–876.

Frost O. (1972): *An algorithm for linearly constrained adaptive array processing*. — Proc. IEEE, Vol. 60, No. 8, pp. 926–935.

Fudge G. and Linebarger D. (1994): *A calibrated generalized sidelobe canceller for wideband beamforming*. — IEEE Trans. Signal Process., Vol. 42, No. 10, pp. 2871–2875.

Haykin S. (1996): *Adaptive Filter Theory, 3rd Ed.* — Englewood Cliffs, NJ: Prentice Hall.

Hendon E. and Reed I. (1990): *A new CFAR sidelobe canceller algorithm for radar*. — IEEE Trans. Aerospace Electr. Syst., Vol. 26, No. 5, pp. 792–803.

Herbordt W. and Kellermann W. (2001): *Efficient frequency domain realization of robust generalized sidelobe cancellers*. — IEEE Conf. Multimedia Signal Process., Cannes, France, pp. 377–382.

Honig M. and Tsatsanis M. (2000): *Multiuser CDMA receivers*. — IEEE Signal Process. Mag., Vol. 17, No. 3, pp. 49–61.

Hoshuyama O., Sugiyama A. and Hirano A. (1999): *A robust adaptive beamformer with a blocking matrix using coefficient constrained adaptive filters*. — IEICE Trans. Fundament., Vol. E82-A, No. 4, pp. 640–647.

Hudson J. (1991): *Adaptive Array Principles*. — London, UK: IEE Press.

Huard K. and Yeh C. (1994): *Gram-Schmidt forward-backward Generalized Sidelobe Canceller*. — IEEE Trans. Aerospace Electron. Syst., Vol. 30, No. 1, pp. 151–160.

Gannot S., Burshtein D. and Weinstein E. (2001): *Signal enchancement using beamforming and nonstationarity with applications to speech*. — IEEE Trans. Signal Process., Vol. 49, No. 8, pp. 1614–1626.

Gershman A. (1999): *Robust adaptive beamforming in sensor arrays*. — AEU-Int. J. Electron. Comm., Vol. 53, No. 6, pp. 305–314.

Ghorayeb S., Lord W. and Udpa S. (1994): *Application of a beamforming technique to ultrasound imaging in nondestructive test*. — IEEE Trans. Ultasonics Ferroel. Freq. Contr., Vol. 41, No. 2, pp. 199–208.

Glentis G., Berberidis K. and Theodoridis S. (1999): *Efficient least squares adaptive algorithms for FIR transversal filtering: a unified view*. — IEEE Signal Process. Mag., Vol. 16, No. 4, pp. 13–42.

Godara L. (1997): *Applications of antenna arrays to mobile communications, Part II: Beamforming and direction-of-arrival considerations*. — Proc. IEEE, Vol. 85, No. 8, pp. 1195–1245.

Goldstein J., Williams D., Merserau R. and Holder E. (1994): *Inter-space and intra-space transformation for sensor array processing*. — Proc. Asilomar Conf. *Signals, Systems, Computers*, Pacific Grove, CA, pp. 638–642.

Griffiths L.J. and Jim C.W. (1982): *An alternative approach to linearly constrained adaptive beamforming*. — IEEE Trans. Antennas Propag., Vol. 30, No. 1, pp. 27–34.

Johnson D. and Dudgeon D. (1993): *Array Signal Processing: Concepts and Techniques*. — Prentice-Hall.

Joho M. and Moschytz G. (1997): *Adaptive beamforming with partitioned frequency-domain filters*. — Proc. IEEE Workshop *Applications of Signal Processing to Audio and Acoustics*, NY, USA.

Kalouptsidis N. and Theodoridis S. (Eds.) (1993): *Adaptive System Identification and Signal Processing Algorithms*. — Prentice Hall.

Kim K., Park Y., Cha I. and Youn D. (1992): *Adaptive multichannel lattice-escalator filter structure: An application to generalized sidelobe canceller*. — IEEE Trans. Signal Process., Vol. 40, No. 7, pp. 1816–1819.

Kohno R. (1998): *Spacial and temporal communication theory using adaptive antenna array*. — IEEE Personal Comm., Vol. 5, No. 1, pp. 28–36.

Kogon S., Williams D. and Holder E. (1996): *Beamspace techniques for hot clutter cancellation*. — IEEE Int. Conf. *Acoust. Speech Signal Process*, Vol. 2, Atlanta, USA, pp. 1177–1180.

Krim H. and Viberg M. (1996): *Two decades or array signal processing research: the parametric approach*. — IEEE Signal Process. Mag., Vol. 13, No. 4, pp. 67–94.

Lamagna A. (1982): *Fast computer algebra*. — IEEE Comp., Vol. 15, No. 9, pp. 43–56.

Lee B., Chang B., Cha I., Kim W. and Youn D. (1987): *Realization of a generalized sidelobe canceller*. — IEEE Trans. Circ. Syst., Vol. 34, No. 7, pp. 759–764.

Lee T. and Tsai T. (2001): *A beamspace-time interference canceling CDMA receiver for sectored communications in multipath environment*. — IEEE J. Select. Areas Comm., Vol. 19, No. 7, pp. 1374–1384.

Leshem A., van der Veen A. and Boonstra A. (2000): *Multichannel interference mitigation techniques in radio astronomy*. — Astrophys. J. Suppl., Vol. 131, pp. 355–374.

Li L., Jeffs B., Poulsen A. and Warnick K. (2002): *Analysis of adaptive array algorithm performance for satellite interference cancellation in radio astronomy*. — Proc. XXVII URSI General Assembly, Maastricht, The Netherlands.

Li X. and Gaillard P. (1988): *Broadband generalized sidelobe canceller using multichannel least squares lattice structure*. — Proc. IEEE Conf. *Acoust. Speach Signal Process*, pp. 1248–1251.

Litva J. and Lo T. (1996): *Digital Beamforming in Wireless Communications*. — Norwood, MA: Artech House.

Long G., Ling F. and Proakis J. (1989): *The LMS algorithm with delayed coefficients adaptation*. — IEEE Trans. Acoust. Speech Sign. Process., Vol. 37, No. 9, pp. 1397–1405.

Long G., Ling F. and Proakis J. (1992): *Corrections to 'The LMS algorithm with delayed coefficients adaptation'*. — IEEE Trans. Acoust. Speech Sign. Process., Vol. 40, No. 1, pp. 230–232.

Martinez D. (1999): *Application of parallel processors to real-time sensor array processing*. — Proc. 13th Int. *Parallel Processing* Symp. and 10th Symp. *Parallel and Distrib. Process.*, San Juan, Puerto Rico, pp. 463–469.

Moon S., Han D. and Cho M., (2001): *Frequency domain partially adaptive array algorithm conbined with CFAR technique*. — Signal Process., Vol. 81, No. 9, pp. 1927–1934.

Monzingo R. and Miller T. (1980): *Introduction to Adaptive Aarrays*. — New York: Wiley.

Mucci R. (1984): *A comparison of efficient beamforming algorithms*, IEEE Trans. Acoust. Speech Signal Process., Vol. 32, No. 3, pp. 548–558.

Narayan S., Peterson A.M. and Narasimba M.J. (1983): *Transform domain LMS algorithm*. — IEEE Trans. Acoust. Speech Signal Process., Vol. 31, No. 3, pp. 609–615.

Nitzberg R. (1999): *Radar Signal Processing and Adaptive Systems*. — Norwood, MA: Artech House.

Parhi K. (1999): *VLSI Digital Signal Processing Systems, Design and Implementation*. — New York: Wiley.

Paulraj A. and Papadias C. (1997): *Space-time processing for wireless communications*. — IEEE Signal Process. Mag., Vol. 14, No. 5, pp. 49–83.

Perry R., Bull D. and Nix A. (1999): *Efficient adaptive complex filtering algorithm with application to channel equalization*. — IEE Proc.-Vis. Image Signal Process., Vol. 146, No. 2, pp. 57–64.

Pillai S. (1989): *Array Signal Processing*. — New York: Springer.

Rappaport T. (1998): *Smart Antennas: Adaptive Arrays, Algorithms, and Wireless Position Location*. — IEEE Press.

Scott I. and Mulgrew B. (1995): *Sparse LCMV beamforming design for suppression of ground clutter in airborn radar*. — IEEE Trans. Signal Process., Vol. 43, No. 12, pp. 2843–2852.

Shanbhag N. (1998): *Algorithmic transformation techniques for low-power wireless VLSI systems design*. — Int. J. Wireless Inf. Netw., Vol. 5, No. 2, pp. 147–171.

Shynk J. (1992): *Frequency-domain and multirate adaptive filtering*. — IEEE Signal Process. Mag., Vol. 9, No. 1, pp. 14–39.

Taveniku M. and Ahlander A. (1997): *Instruction statistics in array signal processing*. — Res. Rep., Halmstad Univ.

Tian Z., Bell K. and van Trees H. (2001): *A recursive least squares implementation of the LCMP beamformer under quadratic constraint*. — IEEE Trans. Signal Process., Vol. 49, No. 6, pp. 1510–1522.

Timmoneri L., Proudler I., Farina A. and McWhirter J. (1994): *QRD-based MVDR algorithm for adaptive multipulse array signal processing*. — IEE Proc.-Radar, Sonar Navigat., Vol. 141, No. 2, pp. 93–102.

van Veen B. and Buckley K. (1988): *Beamforming: A versatile approach to spatial filtering*. — IEEE ASSP Mag., Vol. 5, No. 4, pp. 4–24.

Weiss S., Stewart R., Schaber M., Proudler I. and Hoffman M. (1999): *An efficient scheme for broadband adaptive beamforming*. — Proc. 33rd Asilomar Conf. *Signals*, *Sysems*, *Computers*, Monterey, CA, USA, Vol. I, pp. 495–500.

Wenzler A. and Luder E. (1995): *New structures for compex multipliers and their noise analysis*. — Proc. IEEE ISCAS, Seattle, USA, pp. 1431–1435.

Widrow B. and Stearns S. (1985): *Adaptive Signal Processing*. — Englewood Cliffs, NJ: Prentice Hall.

Winograd S. (1980): *Arithmetic Complexity of Computations*. — Philadelphia: SIAM.

Winters J. (1998): *Smart antennas for wireless systems*. — IEEE Person. Comm., Vol. 5, No. 1, pp. 23–27.

Xu Z. and Tsatsanis M. (1999): *Adaptive minimum variance methods for direct blind multichannel equalization*. — Signal Process., Vol. 73, pp. 125–138.

Ye W., Bar-Nes W. and Haimovich A. (1997): *A self-correcting loop for joint estimation-calibration in adaptive radar*. — IEEE Nat. Radar Conf., Syracuse, NY, pp. 320–324.

Yu J. and Leou M. (2000): *Transformation-based adaptive array beamforming*. — Sign. Process., Vol. 80, No. 2, pp. 231–241.

Yu S. and Ueng F. (2000): *Blind adaptive beamforming based on generalized sidelobe canceller*. — Signal Process., Vol. 80, No. 12, pp. 2497–2506.

Yuen S. (1991): *Exact Least-Squares adaptive beamforming using orthogonalization network*. — IEEE Trans. Aerospace Electr. Sys., Vol. 27, No. 2, pp. 311–330.

Received: 4 April 2002
Revised: 12 March 2003