GSC realisations using the two-dimensional transform-domain LMS algorithm

J. An
B. Champagne

Indexing terms: GSC realisations, Transform-domain LMS algorithm

Abstract: The authors propose two new realisations of the broadband generalised sidelobe canceller (GSC) which are based on a two-dimensional (2D) extension of the transform-domain LMS algorithm. In the first realisation, the tap-input matrix of the GSC (following the blocking network) is mapped into the transform-domain via a 2D unitary image transform. Following this step, the transformed data are weighted and summed to produce the GSC output and the processor weights are updated via the self-orthogonalising LMS algorithm in the transform-domain. The second realisation uses a separable 2D transform and provides a narrowband decomposition of the GSC. The computational complexity of these new GSC realisations is analysed and shown to be comparable to that of the fast FLMS-GSC recently proposed by Chen and Fang. However, as demonstrated by computer simulations, the new GSC realisations have a much faster convergence rate than the FLMS-GSC when the eigenvalue spread of the tap-input correlation matrix is large.

1 Introduction

A broadband adaptive beamformer consists of a multi-input single-output processor together with an adaptive algorithm. The processor linearly combines the signals received by an array of sensors while the adaptive algorithm recursively adjusts the weights of the processor in real time to respond to a signal coming from a desired direction while discriminating against noises from other directions [1]. Among early works on adaptive beamforming, we mention the so-called 'constrained LMS' algorithm of Frost [2]. In this approach, a lineary-constrained stochastic-gradient least-mean-square (LMS) algorithm with self-correcting capability is used to update the processor weights. In effect, the algorithm attempts to minimise the noise power at the array output while maintaining a chosen frequency response in the direction of interest (look-direction). In Reference 3, Griffiths and Jim describe an alternative beamforming structure known as the generalised sidelobe canceller (GSC) for realising the constrained LMS algorithm. Through the use of a blocking matrix, the GSC allows the constrained LMS problem to be transformed into an unconstrained one.

In both Frost's algorithm and the Griffiths–Jim GSC, a vector of time-domain tap inputs is used to update the processor weights at each iteration. As a result, these algorithms suffer from a major drawback which is common to most time-domain implementations of LMS-type adaptive algorithms, namely: their convergence rate decreases as the condition number (i.e., the ratio of maximum to minimum eigenvalues) of the input correlation matrix increases. This difficulty can be overcome by using adaptive algorithms based on least-squares or Kalman filtering theory [4]. Realisations of the GSC based on adaptive multichannel lattice and escalator filters have also been proposed by Lee *et al.* [5]. However, the gain in convergence rate associated with these algorithms is generally obtained at the expense of a significant increase in computational requirements.

Recently, Chen and Fang [6] have proposed an alternative realisation of the GSC using the frequency-domain LMS algorithm with the self-orthogonalising property, simply called the FLMS algorithm. In their approach, each tapped-delay-line at the output of the blocking matrix is transformed individually into the frequency-domain via a one-dimensional discrete Fourier transform (1D-DFT). The processor weights are applied to the transformed data and are then updated via the self-orthogonalising LMS algorithm in the transform-domain, which is known to converge more rapidly than its time-domain counterpart [7, 8]. The main advantage of the Chen–Fang approach, identified here as the FLMS-GSC, is that it can effectively accelerate the convergence rate of the GSC, as demonstrated by computer simulations, without a major increase in computational requirements. This is particularly important for real-time processing of non-stationary array signals.

In this paper, we present two new realisations of the GSC based on a two-dimensional (2D) extension of the transform-domain LMS algorithm. In both realisations, the set of tap inputs following the blocking network of the GSC is viewed as an image which is mapped into the transform-domain via a 2D unitary transform. The transformed data are then linearly combined to produce the GSC output and the processor weights are updated using the self-orthogonalising LMS algorithm in the transform-

Support for this work was provided by NSERC operating grant OGP0105533. Part of this work was presented at the IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, Victoria, Canada, May 1993.

Paper 1411F (E5), first received 5th November 1993 and in revised form 21st June 1994

The authors are with INRS-Télécommunications, Université du Québec, 16 Place du Commerce, Verdun, Québec, Canada H3E 1H6

domain. The second realisation, restricted to separable 2D transforms, provides a narrowband decomposition of the GSC. The two realisations offer a trade-off between computational complexity and convergence rate. Computer simulations show that these new GSC realisations have faster convergence rates than the Chen–Fang FLMS-GSC.

2 Chen–Fang FLMS-GSC

In the Chen–Fang FLMS-GSC (Fig. 1), the output signals from an array of K sensors are fed to delay elements needed to steer the array in the desired look-direction. Two different processing paths are then taken. The upper path consists of a conventional beamformer followed by a fixed 'target signal filter' used to control the frequency response of the beamformer in the look-direction. The lower path is the sidelobe canceller. It consists of $K - 1$ subtractors followed by a set of $K - 1$ tapped-delay-lines (TDL), each with $L - 1$ unit-delay elements.

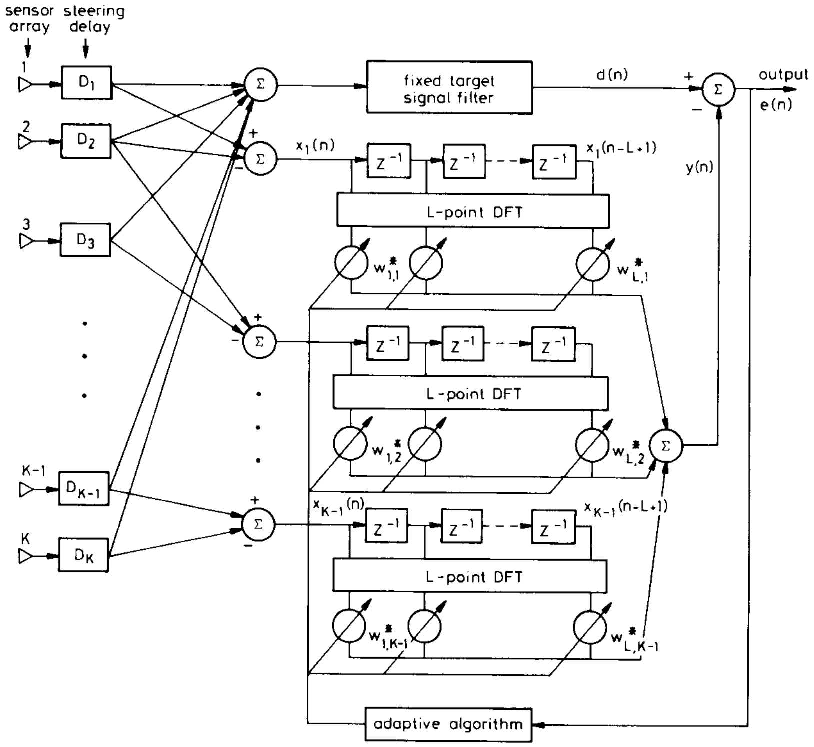

Fig. 1 *The Chen–Fang FLMS-GSC*

Let $x_i(n)$, $i = 1, \ldots, K - 1$, denote the output of the ith subtractor at time n. In the FLMS-GSC, the time-domain tap-input vector of the ith TDL, $X_i(n)$, is transformed into the frequency-domain vector $U_i(n)$ by means of the L-point 1D-DFT, represented here by the matrix $\mathbf{D}$:

$$U_i(n) = \mathbf{D}X_i(n) \tag{1}$$

where

$$X_i(n) \triangleq [x_i(n), x_i(n-1), \ldots, x_i(n-L+1)]^T \tag{2}$$

$$U_i(n) \triangleq [u_{0,i}(n), \mu_{1,i}(n), \ldots, u_{L-1,i}(n)]^T \tag{3}$$

Let $X(n)$ and $U(n)$ respectively denote the $(K-1)L$-dimensional time-domain and frequency-domain tap-input vectors obtained by concatenation of the $X_i(n)$ and $U_i(n)$ as follows:

$$X(n) \triangleq [X_1^T(n), X_2^T(n), \ldots, X_{K-1}^T(n)]^T \tag{4}$$

$$U(n) \triangleq [U_1^T(n), U_2^T(n), \ldots, U_{K-1}^T(n)]^T \tag{5}$$

Then, from eqns. 1, 4 and 5, it is clear that

$$U(n) = \mathscr{D}X(n) \tag{6}$$

where

$$\mathscr{D} \triangleq \begin{bmatrix} \mathbf{D} & \mathbf{0} & & \mathbf{0} \\ \mathbf{0} & \mathbf{D} & & \mathbf{0} \\ & & \ddots & \\ \mathbf{0} & \mathbf{0} & & \mathbf{D} \end{bmatrix} \tag{7}$$

Following the transformation (6), the output of the adaptive sidelobe canceller, $y(n)$, is obtained as a linear combination of the elements of the vector $U(n)$. Let $W(n)$ denote the $(K-1)L$-dimensional weight vector of all the TDLs at the nth iteration, i.e.,

$$W(n) \triangleq [W_1^T(n), W_2^T(n), \ldots, W_{K-1}^T(n)]^T \tag{8}$$

$$W_i(n) \triangleq [w_{0,i}(n), w_{1,i}(n), \ldots, w_{L-1,i}(n)]^T \tag{9}$$

Then

$$y(n) = W^H(n)U(n) \tag{10}$$

The weight vector $W(n)$ is continually updated so as to minimise the power of the error signal,

$$e(n) = d(n) - y(n) \tag{11}$$

between the output of the conventional beamformer, $d(n)$, and that of the sidelobe canceller, $y(n)$. To this end, a self-orthogonalising LMS algorithm with accelerated convergence rate is used [9]. In this algorithm, the gradient estimate is premultiplied by the inverse correlation matrix of $U(n)$, so that the weights are updated as follows:

$$W(n+1) = W(n) + 2\gamma \mathbf{R}_{UU}^{-1} U(n) e^{*}(n) \tag{12}$$

where

$$\mathbf{R}_{UU} \triangleq E[U(n)U^{H}(n)] \tag{13}$$

is the correlation matrix of $U(n)$ and γ is the step-size parameter, which must satisfy the condition $0 < \gamma < 1/(K-1)L$ to ensure convergence of the algorithm.

The role of $\mathbf{R}_{UU}^{-1}$ in eqn. 12 is to reduce the eigenvalue spread of the matrix governing the adaptation process. In practice, $\mathbf{R}_{UU}^{-1}$ is unknown and must be estimated from the data. In the FLMS-GSC, the following approach is used (see also Reference 8). First, $\mathbf{R}_{UU}$ is approximated by a diagonal matrix, i.e.,

$$\mathbf{R}_{UU} \approx \operatorname{diag}(r_{0,1}, \ldots, r_{L-1,1}, \ldots, r_{0,K-1}, \ldots, r_{L-1,K-1}) \tag{14}$$

where $r_{l,k} \triangleq E[|u_{l,k}(n)|^2]$ is the power of the lth frequency component of the kth TDL. The diagonal elements $r_{l,k}$ are then estimated recursively through the equation

$$\hat{r}_{l,k}(n) = \beta \hat{r}_{l,k}(n-1) + (1-\beta)|\mu_{l,k}(n)|^2 \tag{15}$$

where β $(0 \leqslant \beta \leqslant 1)$ is a smoothing constant. Finally, the desired estimate of $\mathbf{R}_{UU}^{-1}$ is given by

$$\hat{\mathbf{R}}_{UU}^{-1} = \operatorname{diag}(1/\hat{r}_{0,1}, \ldots, 1/\hat{r}_{L-1,K-1}) \tag{16}$$

The purpose of the Chen–Fang approach is to accelerate the convergence of the conventional GSC by first mapping the tap-input vector of each TDL into the frequency-domain via an L-point 1D-DFT and then using the self-orthogonalising LMS adaptive algorithm. In this respect, the role of the 1D-DFT is to remove the temporal correlation between the tap-inputs of each TDL. The effectiveness of this approach is confirmed by computer simulations in Reference 6.

In almost all GSC applications, there is also spatial correlation between adjacent TDL channels because of the propagating nature of the received signals. We note, however, that the FLMS-GSC cannot remove this spatial correlation. Indeed, each TDL in Fig. 1 is transformed individually with a 1D-DFT, which amounts to (approximately) diagonalising only the principal submatrices of the correlation matrix of the tap-input vector $X(n)$(eqn. 4). Accordingly, the spatial correlation between adjacent channels is not removed and the frequency-domain correlation matrix $\mathbf{R}_{UU}$ is not block diagonal in general, contrary to the assumption in eqn. 14. Under these conditions, the use of the estimate (eqn. 16) in the weight update eqn. 12 is not 'optimal' and the potential improvement in convergence rate offered by the self-orthogonalising transform-domain LMS algorithm is not fully achieved. This represents a fundamental limitation of the FLMS-GSC.

3 New GSC realisations using 2D transform-domain LMS algorithms

In the light of the above discussion, it appears possible to increase the convergence rate of the GSC further by removing both spatial and temporal correlations between the tap-inputs of all the TDLs prior to the application of the self-orthogonalising LMS algorithm. In this Section, we propose two new realisations of the GSC based on 2D extensions of the transform-domain LMS algorithm, in which both types of correlations are removed (at least partially). These realisations are described separately in Sections 3.1 and 3.2. Related computational issues are discussed in Section 3.3.

3.1 The 2D-TDLMS-GSC

A block diagram of the proposed 2D-TDLMS-GSC (where TD stands for transform-domain) is shown in Fig. 2. Its basic principles of operation are similar to those of the FLMS-GSC. However, instead of transforming the tap-input vectors of each TDL individually with a 1D-DFT, the set of all tap-input vectors is viewed as an image which is mapped into the transform-domain via a 2D unitary transform, whose purpose is to remove both temporal and spatial correlations between the tap inputs of all the TDLs. Following this step, the transformed data are weighted and summed as in the conventional GSC. Finally, the self-orthogonalising LMS algorithm in the transform-domain is used to update the processor weights. Additional details are provided below.

To unify the presentation, we use the same vector notation as in Section 2. Hence, the images formed by the tap-inputs, the transform coefficients and the corresponding weights are represented by $X(n)$ (eqn. 4), $U(n)$ (eqn. 5) and $W(n)$ (eqn. 8), respectively. In image terminology, this corresponds to a lexicographic line-ordering [10]. Using this notation, the relationship between the tap-input image and its transform can be expressed as

$$U(n) = \mathscr{T} X(n) \tag{17}$$

where $\mathscr{T}$ is the $(K-1)L \times (K-1)L$ matrix of the 2D unitary transform under consideration.

Since the purpose of eqn. 17 is to remove both temporal and spatial correlations between the elements of $X(n)$, the Karhunen–Loeve transform (KLT) is, at least in theory, the most appropriate choice for $\mathscr{T}$. To see this, let us briefly review some relevant properties of the KLT [10]. Let $R_{XX} = E[X(n)X^{H}(n)]$ denote the correlation matrix of $X(n)$, which is assumed to have zero mean. Let $\{\psi_{l,k}\}$ be a complete orthonormal set of eigenvectors of R_{XX} and let $\{\lambda_{l,k}\}$ denote the corresponding eigenvalues. That is,

$$R_{XX}\psi_{l,k} = \lambda_{l,k}\psi_{l,k}, \quad \psi_{l,k}^{H}\psi_{l',k'} = \delta_{l,l'}\delta_{k,k'} \tag{18}$$

where $\lambda_{l,k} > 0$ if R_{XX} is positive-definite. By definition, the KLT for the image $X(n)$ is given by the unitary matrix

$$\mathscr{K} = [\psi_{0,1}, \ldots, \psi_{L-1,1}, \ldots, \psi_{0,K-1}, \ldots, \psi_{L-1,K-1}]^{H} \tag{19}$$

Now, if the KLT is used in eqn. 17, i.e. if $\mathscr{T} = \mathscr{K}$, it follows from eqn. 18 that

$$\mathbf{R}_{UU} = \operatorname{diag}[\lambda_{0,1}, \ldots, \lambda_{L-1,K-1}] \tag{20}$$

This result, referred to as the decorrelation property of the KLT, signifies that the transformed coefficients $\{u_{l,k}(n)\}$ are uncorrelated, which is the desired property.

Except for the use of different transform matrices (i.e., eqn. 17 instead of eqn. 6), the 2D-TDLMS-GSC is similar to the FLMS-GSC. In particular, the output of the adaptive sidelobe canceller, $y(n)$, is given by eqn. 10 and the self-orthogonalising transform-domain LMS algorithm, eqns. 11 and 12, is used to update the weight vector

$W(n)$. Moreover, the range of permissible values for the step-size parameter γ is unchanged (i.e. $0 < \gamma < 1/(K-1)L$). To estimate the inverse matrix $\mathbf{R}_{UU}^{-1}$ needed in eqn. 12, we follow the same procedure as in Section 2. This time, however, the use of a 2D transform provides a much sounder foundation for the approximation made in eqn. 14. In particular, when the KLT is used in eqn. 17, i.e. when $\mathscr{T} = \mathscr{K}$, eqn. 14 is exact and the matrix controlling the convergence of the adaptive process is equal to the identity matrix, which is consistent with the correct application of the self-orthogonalising technique [9].

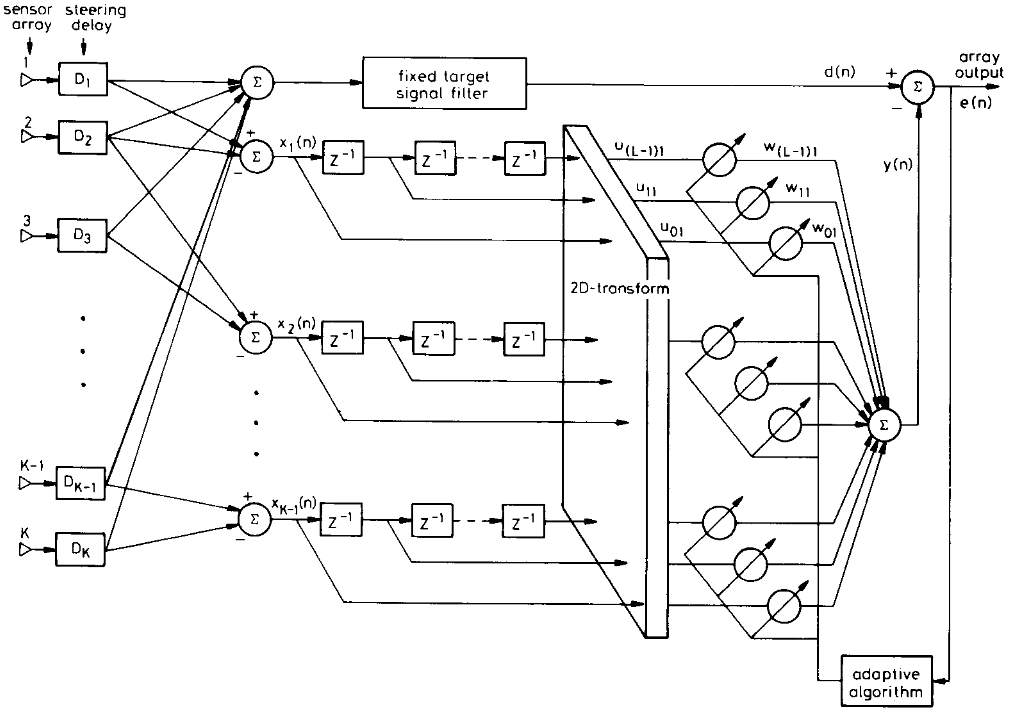

Fig. 2 *The 2D-TDLMS-GSC*

We note, however, that the KLT is signal-dependent and does not have a fast algorithm in general. Hence, its implementation in real time poses serious practical problems. In such cases, it is necessary to use suboptimal transforms that are signal-independent and for which fast algorithms exist. Typical candidates include the 2D discrete cosine transform (2D-DCT), the 2D-DFT and several others [10]. When suboptimal transforms such as the 2D-DCT or the 2D-DFT are used in the 2D-TDLMS-GSC, the matrix $\mathbf{R}_{UU}$ (eqn. 13) is only approximately diagonal. Nevertheless, the advantages offered by the 2D-TDLMS-GSC in terms of accelerated convergence of the adaptive algorithm are preserved. This is confirmed by computer simulations in Section 4.

3.2 The 2D-NDLMS-GSC

Most of the 2D unitary transforms commonly used in image processing applications are separable. Here, we present an alternative realisation of the GSC which is obtained from the 2D-TDLMS-GSC by using a separable 2D transform and by making additional modifications. We begin with a brief review of separable 2D transforms.

An L-point unitary transform of a 1D data sequence $x(n)$ can be written in the form

$$u_r(n) = \alpha_L(r) \sum_{l=0}^{L-1} x(n-1)\mathscr{W}_L(r, l), \qquad r = 0, 1, \ldots, L-1 \quad (21)$$

where $\alpha_L(r)$ is the normalisation coefficient and $\mathscr{W}_L(r, l)$ is the kernel of the transform. In the case of the normalised 1D-DFT, we have

$$\alpha_L(r) = 1/\sqrt{L}$$
$$\mathscr{W}_L(r, l) = e^{-j2\pi rl/L} \quad (22)$$

For the 1D-DCT, we have

$$\alpha_L(r) = \begin{cases} 1/\sqrt{L}, & r = 0 \\ \sqrt{(2/L)}, & r = 1, \ldots, L-1 \end{cases}$$
$$\mathscr{W}_L(r, l) = \cos\,[(2l+1)r\pi/(2L)] \quad (23)$$

Now, consider the $(K-1) \times L$ image formed (at time n) by the tap-inputs $\{x_k(n-l),\ k = 1, \ldots, K,\ l = 0, \ldots, L-1\}$ in the 2D-TDLMS-GSC of Fig. 2. For this image, a possible 2D extension of the 1D transform (eqn. 21) is given by

$$u_{r,p}(n) = \alpha_L(r)\alpha_{K-1}(p-1) \sum_{k=1}^{K-1} \sum_{l=0}^{L-1} x_k(n-l) \times \mathscr{W}_L(r, l)\mathscr{W}_{K-1}(p-1, k-1) \quad (24)$$

for $r = 0, 1, \ldots, L-1$ and $p = 1, 2, \ldots, K-1$. This 2D unitary transform is said to be separable, for both its normalisation coefficient and kernel can be expressed as the products of two 1D normalisation coefficients and kernels, respectively. The 2D-DFT and 2D-DCT are obtained as particular cases of eqn. 24 when eqns. 22 and 23 are used, respectively.

Being separable, the 2D transform (eqn. 24) can be realised as a succession of two sets of 1D tranforms. To illustrate this point, suppose that eqn. 21 represents a 1D-DFT. According to eqn. 24, the tap-input vectors of each TDL are first transformed into the frequency-domain via $K-1$ parallel L-point 1D-DFTs, as in the Chen–Fang approach. Then, the spatial sequences of 1D-DFT coefficients obtained from successive TDLs for each frequency bin are transformed individually into the spatial frequency-domain via L parallel $(K-1)$-point 1D-DFTs.

Following this interpretation of eqn. 24, the use of a separable 2D transform in the 2D-TDLMS-GSC can be viewed as a narrowband decomposition of the GSC in which the first set of 1D transforms provides the desired narrowband decomposition of the tap-input data and the second set of 1D transforms remove the remaining spatial correlation between the transformed coefficients so as to accelerate the convergence of the adaptive process. We note, however, that the reference signal $d(n)$ and the sidelobe canceller output $y(n)$ in the 2D-TDLMS-GSC are time-domain signals. Hence, even when a separable 2D transform is used, the 2D-TDLMS-GSC does not have a pure narrowband decomposition structure because the adaptation process is based on temporal information.

In Fig. 3, we present an alternative GSC realisation, called the 2D-NDLMS-GSC (where ND stands for narrowband decomposition), which has a genuine narrowband decomposition structure. The 2D-DFT is assumed in Fig. 3 to simplify the presentation; however, any separable 2D unitary transform of the type (eqn. 24) can be used. The mathematical operations involved in the 2D-NDLMS-GSC are specified below.

Step 1: Map the reference signal $d(n)$ into the transform-domain via an L-point unitary transform as in eqn. 21

$$d_r(n) = \alpha_L(r) \sum_{l=0}^{L-1} d(n-l)\mathscr{W}_L(r, l),$$
$$r = 0, 1, \ldots, L-1 \quad (25)$$

Note that the fixed target signal filter in Fig. 3 can be implemented in the transform-domain following the transformation (eqn. 25).

Step 2: Map the tap-inputs $\{x_k(n-l)\}$ into the transform-domain using the corresponding $(K-1) \times L$-point separable 2D transform, as given by eqn. 24.

Step 3: For $r = 0, 1, \ldots, L-1$, perform sidelobe cancellation independently:

$$y_r(n) = \sum_{p=1}^{K-1} w^*_{r,p}(n) u_{r,p}(n) \quad (26)$$

$$e_r(n) = d_r(n) - y_r(n) \quad (27)$$

Step 4: For $r = 0, 1, \ldots, L-1$, update the weight vector $[w_{r,1}(n), \ldots, w_{r,K-1}(n)]^T$ using a different version of the transform-domain self-orthogonalising LMS algorithm:

$$w_{r,p}(n+1) = w_{r,p}(n) + 2\gamma \hat{r}^{-1}_{r,p} u_{r,p}(n) e^*_r(n) \quad (28)$$

where $\hat{r}_{rp}$ is obtained as in eqn. 15 and $0 < \gamma < 1/(K-1)$ to ensure the convergence of the algorithm. Note that eqn. 28 is different from the weight update eqn. 12 used in the FLMS-GSC and the 2D-TSLMS-GSC, since a different error signal is used for each frequency bin.

Step 5: Obtain the time-domain output $e(n)$ of the GSC via an inverse L-point transform. Note that only

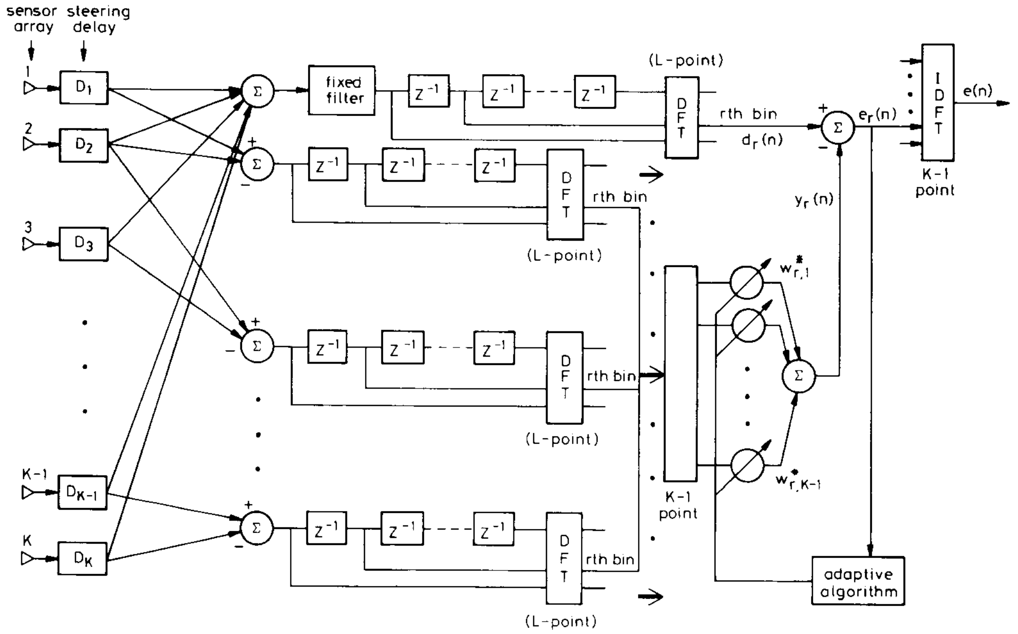

Fig. 3 *The 2D-NDLMS-GSC*
Realisation based on the 2D-DFT

the first component of the inverse transform is needed. For instance, if eqn. 25 represents an L-point 1D-DFT, then $e(n)$ can be recovered simply as

$$e(n) = \sqrt{\left(\frac{1}{L}\right)} \sum_{r=0}^{L-1} e_r(n) \qquad (29)$$

Because of their ability to remove the spatial correlation between adjacent TDL channels, both the 2D-TDLMS-GSC and the 2D-NDLMS-GSC are expected to have faster convergence rates than the Chen–Fang FLMS-GSC. However, it is more difficult to predict which one of these new methods has the fastest convergence rate. In the Appendix, we show that the matrices governing the adaptation of each narrowband component of the 2D-NDLMS-GSC have a smaller eigenvalue spread than the matrix governing the adaptation of the 2D-TDLMS-GSC. It is therefore reasonable to believe that the 2D-NDLMS-GSC has a faster convergence rate than the corresponding 2D-TDLMS-GSC. This is confirmed by simulations in Section 4.

3.3 Computational complexity

In this Section, the new GSC realisations proposed above are compared with the Chen–Fang FLMS-GSC [6] and the Griffith–Jim GSC [3] in terms of the number of complex multiplications required per adaptation cycle. To simplify the presentation, we assume that the 2D-DFT is used in both the 2D-TDLMS-GSC and the 2D-NDLMS-GSC. However, a similar comparison of the computational complexity can be made for other types of separable 2D tranforms. Related issues pertaining to the recursive implementation of the 1D and 2D-DFT are also addressed.

The original GSC realisation proposed by Griffith and Jim is based on plain time-domain LMS adaptation. Hence, it requires on the order of $2L(K-1)$ complex multiplications per cycle (even if different normalised convergence factors are used for each TDL, as is the case in our simulations).

In the FLMS-GSC, the tap-input vector of each TDL is mapped into the frequency-domain via an L-point 1D-DFT. Assuming that L is a power of 2 and that the radix-2 FFT algorithm is used, each L-point 1D-DFT requires approximately $L \log_2 L$ complex multiplications (and $L \log_2 L$ complex additions) [11]. Since there are $K-1$ TDLs, a total of $(K-1)L \log_2 L$ complex multiplications are required. Following this transformation step, each iteration of the self-orthogonalising LMS algorithm (eqns. 10–16) requires $3.5L(K-1)$ complex multiplications.

Now consider the 2D-TDLMS-GSC based on the 2D-DFT. Being separable, the 2D-DFT can be realised as a succession of two sets of 1D-DFTs. Moreover the sequence of 1D-DFT operations is reversible. In this case, the most efficient approach is first to transform each column of tap-inputs in Fig. 2 using $(K-1)$-point 1D-DFTs, and then transform each line of the resulting image using L-point 1D-DFTs. Indeed, one will note that the columns of tap-inputs in Fig. 2 are shifted down to the right by one position every iteration. Hence, only one $(K-1)$-point 1D-DFT is actually needed for the first column of tap-inputs entering the TDLs, which correspond the new data; the 1D-DFTs of the other columns are available from previous iterations. Accordingly, the 2D-DFT requires a total of $(L-1)\log_2(K-1) + (K-1)L \log_2 L$ complex multiplications. An additional $3.5L(K-1)$ complex multiplications per cycle are required for the self-othogonalising LMS algorithm.

The computational requirements of the 2D-NDLMS-GSC are similar to those of the 2D-TDLMS-GSC, except that an additional $L \log_2 L$ complex multiplications are required to implement the 1D-DFT of the reference signal $d(n)$ in eqn. 25. The results of this discussion are summarised in Table 1. Based on these results, we conclude that the computational complexity of the new GSC realisations proposed in this paper only marginally exceeds that of the Chen–Fang FLMS-GSC.

More efficient implementations of the proposed GSC realisations are possible based on recursive forms of the 1D-DFT and the 1D-DCT [6, 7]. For instance, the 1D-DFT of the tap-inputs of the kth TDL, i.e. $\{x_k(n-l), l = 0, 1, \ldots, L-1\}$, can be evaluated recursively over time as follows:

$$u_{r,k}(n) = u_{r,k}(n-1)e^{-j2\pi r/L} + L^{-1/2}[x_k(n) - x_k(n-L)] \qquad (30)$$

where $0 \leqslant r \leqslant L-1$ and $1 \leqslant k \leqslant K-1$. To reduce the computation time of the FLMS-GSC, Chen and Fang [6] have suggested the use of eqn. 30 for each of the $K-1$ TDLs in Fig. 1. The possibility of using a similar recursion with the new GSC realisations proposed above is also obvious.

However, simulation experiments with the FLMS-GSC have revealed some stability difficulties associated with the recursion (eqn. 30). These difficulties, similar to those encountered with the frequency-sampling structure for FIR filters [11], are due to the pole of eqn. 30 at $z = \exp(-j2\pi r/L)$ on the unit circle. As a result, small round-off errors in the calculation or incorrect initial conditions generate oscillations in the transform-domain coefficients, which also propagate to the output of the beamformer. This type of behaviour would pose particular problems in a low-resolution fixed-point implementation.

One way to overcome these difficulties is to introduce an exponential window on the tap-input signals $x_k(n-l)$. The corresponding recursion is then given by

$$u_{r,k}(n) = \alpha u_{r,k}(n-1)e^{-j2\pi r/L} + L^{-1/2}[x_k(n) - \alpha^L x_k(n-L)] \qquad (31)$$

where $0 < \alpha < 1$. The effectiveness of this approach in the FLMS-GSC was verified by computer simulations. In our experiments, a value of α slightly less than one (typically 0.95) was sufficient to eliminate the stability

Table 1: Number of complex multiplications per cycle for various GSC realisations (figures based on the use of FFT algorithm)

GSC realisations	Complex multiplications per cycle
Griffiths–Jim GSC	$2L(K-1)$
FLMS-GSC	$(K-1)L \log_2 L + 3.5(K-1)L$
2D-TDLMS-GSC	$(K-1)L \log_2 L + 3.5(K-1)L + (K-1) \log_2(K-1)$
2D-NDLMS-GSC	$KL \log_2 L + 3.5(K-1)L + (K-1) \log_2(K-1)$

difficulties associated with eqn. 30, without affecting the overall performance of the system.

The recursion (eqn. 31) can be used to reduce the computational load of the proposed GSC realisations in a significant way. For instance, when eqn. 31 is used following the $(K - 1)$-point 1D-DFT on the first column of tap-inputs, the number of complex multiplications per cycle needed for the 2D-DFT is reduced to $(K - 1) \log_2(K - 1) + 1.5(K - 1)L$.

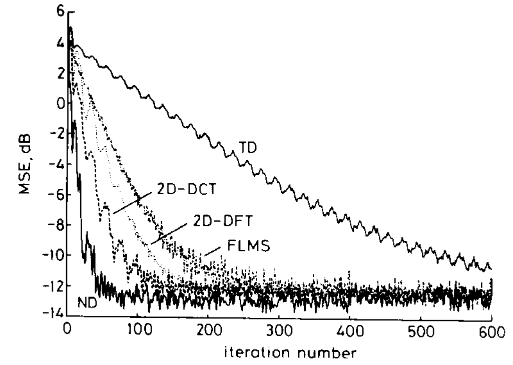

Fig. 4 *Learning curves of various GSC realisations*

Estimated by averaging 200 runs

TD	Griffith-Jim's GSC
FLMS	FLMS-GSC
2D-DFT	2D-DFT-based 2D-TDLMS-GSC
2D-DCT	2D-DCT-based 2D-TDLMS-GSC
ND	2D-NDLMS-GSC

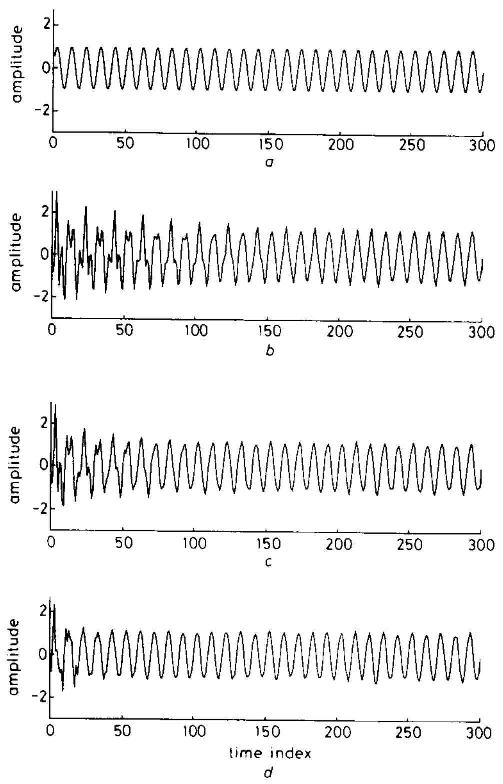

Fig. 5 *Comparison of the output waveforms of the FLMS-GSC (FLMS), 2D-DCT-based 2D-TDLMS-GSC (2D-DCT) and the 2D-NDLMS-GSC (ND) with the original signal*

a Original signal
b FLMS
c 2D-DCT
d ND

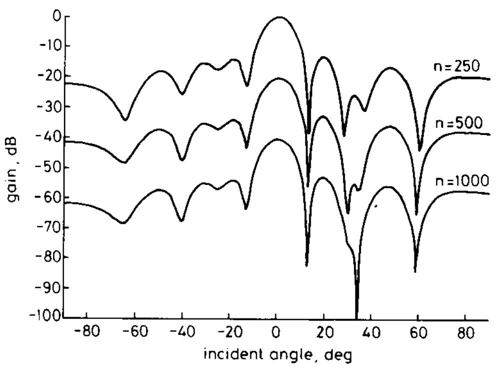

Fig. 6 *Beampatterns of FLMS-GSC at frequency $f_1 = 0.3$*

After 250, 500 and 1000 iterations

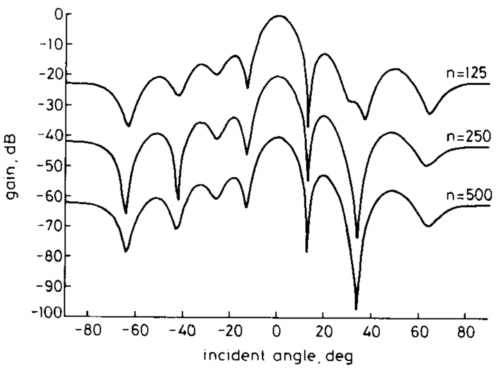

Fig. 7 *Beampatterns of 2D-DCT-based 2D-TDLMS-GSC at frequency $f_1 = 0.3$*

After 125, 250 and 500 iterations

4 Simulation results

Computer simulations were used to study the convergence performance of the new GSC realisations proposed above, in relation to that of the FLMS-GSC and the Griffith-Jim GSC. Note that several different implementations of the 'transform-domain GSCs' are possible depending on the type of unitary transform being used. In the following discussion we consider four practical and computationally efficient combinations based on the DFT and the DCT families, namely: DFT-based FLMS-GSC, two different versions of the 2D-TDLMS-GSC based on the 2D-DFT and the 2D-DCT, and DFT-based 2D-NDLMS-GSC.

The simulation scenario is similar to that of Reference 6. The wave field consists of one target signal (the 'desired' signal) and three jammers in additive uncorrelated background noise. The target signal and the jammers are stationary narrowband plane waves with different directions of arrival. The background noise is a zero-mean Gaussian white noise process. Specific values of the simulation parameters are given below, where f_i and θ_i respectively denote the normalised frequency and incident angle (relative to broadside) of the plane wave signals and where SNR and JNR respectively denote the signal-to-noise and jammer-to-noise ratios:

Target signal: $f_0 = 0.1, \theta_0 = 0°, SNR = 10$ dB.
Jammer 1: $f_1 = 0.3, \theta_1 = 34°, JNR1 = 20$ dB.
Jammer 2: $f_2 = 0.4, \theta_2 = -49°, JNR2 = 40$ dB.
Jammer 3: $f_3 = 0.25, \theta_3 = -24°, JNR3 = 30$ dB.

The wave field is spatially sampled with a uniform linear array of $K = 17$ sensors, which is steered in the direction of the target signal. The sensor spacing is equal to half the wavelength at some prescribed maximum frequency of interest, say f_{max}, and the sensor outputs are temporally sampled at the corresponding Nyquist rate, i.e., $2f_{max}$. In all the GSC realisations, the number of taps per TDL in the sidelobe canceller is set to $L = 8$, with each unit delay element introducing a physical delay of $T_s = 1/(2f_{max})$. Moreover, the fixed target signal filter is an all-pass, linear phase (i.e., distortionless) filter with a gain $1/K$ and a pure delay of $L - 1$ time samples. The following values of the step-size parameter were used initially: Griffith–Jim GSC: $\gamma = 1.21 \times 10^{-3}$; FLMS-GSC: $\gamma = 1.2 \times 10^{-3}$; 2D-TDLMS-GSC: $\gamma = 1.116 \times 10^{-3}$; 2D-NDLMS-GSC: $\gamma = 7.1 \times 10^{-3}$. The value of γ for the FLMS-GSC is taken within the interval $0 < \gamma < 1/(K - 1)L = 7.81 \times 10^{-3}$ to ensure the converge of the algorithm. The other values of γ are chosen empirically to achieve the same steady-state value of the mean-squared error (MSE), which is defined here as $E[|e(n) - s(n)|^2]$, where $e(n)$ is the GSC output and $s(n)$ is the desired target signal. In all cases, the smoothing constant β occuring in eqn. 15 is set to 0.7.

The learning curves of the five adaptive beamformers are plotted in Fig. 4. These curves show the MSE at the output of each GSC during the adaptation and are obtained by ensemble averaging over 200 independent runs. As expected, the 2D-TDLMS-GSC based on the 2D-DFT converges more rapidly than the FLMS-GSC. The 2D-TDLMS-GSC based on the 2D-DCT has a still larger convergence rate that its 2D-DFT counterpart, which can be attributed to the better ability of the 2D-DCT to remove correlation between the random elements of the tap-input image. The 2D-NDLMS-GSC provides the largest convergence rate, in agreement with the result of theorem 1 in the Appendix. The waveforms of the error signal $e(n)$ at the output of the FLMS-GSC, 2D-TDLMS-GSC based on the 2D-DCT and the 2D-NDLMS-GSC are shown in Fig. 6, together with the original target signal $s(n)$ (single run). The results are consistent with those of Fig. 5.

Fig. 6 shows the adaptive beampatterns of the FLMS-GSC at frequency $f_1 = 0.3$ after $k = 250$, 500 and 1000 iterations, respectively ($K = 15$, $\gamma = 2.232 \times 10^{-3}$, 100-run average). Fig. 7 shows the adaptive beampatterns of the 2D-DCT-based 2D-TDLMS-GSC under the same conditions but after 125, 250 and 500 iterations, respectively. In both cases, the level of suppression in the direction of Jammer 1 ($\theta_1 = 34°$) increases with time as the filter weights converge to the optimal solution, reaching a steady-state value of approximately 60 dB. However, comparison of Figs. 6 and 7 reveals that it takes considerably less time for the 2D-TDLMS-GSC to achieve a given level of suppression than for the FLMS-GSC. For instance, in this particular example, steady-state is reached at around $k = 500$ iterations for the 2D-TDLMS-GSC compared with $k = 1000$ for the FLMS-GSC. It is interesting to note that here the use of the 2D-DCT results in a reduction of about 12.3 dB in the estimated eigenvalue spread of the correlation matrix of the transformed image $U(n)$ (eqn. 17), compared with the FLMS-GSC. Finally, we observe that despite following different adaptive trajectories, the steady-state beampatterns of the two methods are very similar. The adaptive beampatterns of the 2D-DFT-based 2D-TDLMS-GSC and 2D-NDLMS-GSC (not shown) also converge more rapidly than those of the FLMS-GSC, in agreement with the results in Fig. 5.

5 Conclusion

Several extensions of this work are possible. A first issue of interest is the perfomance prediction of various 2D transforms in the proposed GSC realisations and the selection of the most appropriate transform in a given application. The performance index recently proposed in Reference 12 could be used to investigate this matter. Another issue of interest is that of partially adaptive beamforming [13]. Indeed, whenever the interference field consists of a few strong narrowband jammers in additive noise, many of the transform-domain coefficients $u_{lk}(n)$ in eqn. 17 are relatively small. In this case, partially adaptive versions of the proposed GSC realisations can be obtained by setting the smaller coefficients to zero. In practice, this can be achieved by comparing $\hat{r}_{lk}(n)$ (eqn. 15) with a threshold at repeated time intervals. Finally, we note that the GSC realisations proposed in this paper can be extended to two-dimensional arrays by using 3D unitary transforms.

6 References

1 COMPTON, R.T. Jr.: 'Adaptive antennas' (Prentice-Hall, Englewood Cliffs, NJ, 1988)
2 FROST, III, O.L.: 'An algorithm for linearly-constrained adaptive array processing', *Proc. IEEE*, 1972, **60**, pp. 926–935
3 GRIFFITHS, L.J., and JIM, C.W.: 'An alternative approach to linearly constrained adaptive beamforming', *IEEE Trans. Antennas Propag.*, 1982, **AP-30**, pp. 27-34
4 HAYKIN, S.: 'Adaptive filter theory' (Prentice-Hall, Englewood Cliffs, NJ, 1986)
5 LEE, B.H., CHANG, B.K., CHA, I.W., KIM, W.K., and YOUN, D.H.: 'Realization of a generalized sidelobe canceller', *IEEE Trans. Circuits Syst.*, 1987, **CAS-34**, pp. 759-764
6 CHEN, Y.-H., and FANG, H.-D.: 'Frequency-domain implementation of Griffiths–Jim adaptive beamformer', *J. Acoust. Soc. Am.*, 1992, **91**, pp. 3354–3366
7 NARAYAN, S.S., PETERSON, A.M., and NARASIHMA, M.J.: 'Transform domain LMS algorithm', *IEEE Trans. Acoust., Speech, Signal Process.*, 1983, **ASSP-31**, pp. 609–615
8 LEE, J.C., and UN, C.K.: 'Performance of transform-domain LMS adaptive digital filters', *IEEE Trans. Acoust. Speech Signal Process.*, 1986, **ASSP-34**, pp. 499–510
9 GITLIN, R.D., and MAGEE, F.R. Jr.: 'Self-orthogonalizing adaptive equalization algorithms', *IEEE Trans. Comm.*, 1977, **COM-25**, pp. 666-672
10 JAIN, A.J.: 'Fundamentals of digital image processing' (Prentice-Hall, Englewood Cliffs, NJ, 1989)
11 OPPENHEIM, A.V., and SCHAFER, R.W.: 'Discrete-time signal processing' (Prentice-Hall, Englewood Cliffs, NJ, 1989)
12 FARHANG-BOROUJENY, B., and GAZOR, S.: 'Selection of orthonormal transforms for improving the performance of the transform domain normalised LMS algorithm', *IEE Proc. F Commun. Radar Signal Process.*, 1992, **139**, pp. 327–335
13 VAN VEEN, B.D., and BUCKLEY, K.M.: 'Beamforming: a versatile approach to spatial filtering', *IEEE ASSP Mag.*, 1988, pp. 4–24
14 AN, J.: 'Realization of the generalized sidelobe canceller via two-dimensional orthogonal transformations'. MSc thesis, INRS-Telecommunications, Université du Québec, Montreal, 1993
15 GOLUB, G.H., and VAN LOAN, C.F.: 'Matrix computations' (Johns Hopkins University Press, Baltimore, 1989)

7 Appendix

In this appendix, we show that the matrices governing the adaptation of each narrowband component of the 2D-NDLMS-GSC have a smaller eigenvalue spread than the matrix governing the adaptation of the 2D-TDLMS-GSC. We begin with some related matrix properties. The main result is stated in theorem 1.

Property 1: Let $\mathbf{S}$ be a principal submatrix of a positive-definite Hermitian matrix $\mathbf{R}$. Then,

$$\chi(\mathbf{S}) \leqslant \chi(\mathbf{R}) \tag{32}$$

where $\chi(.)$ denotes the condition number of its matrix argument.

Proof: See Reference 14.

Definition: Let $\delta_{ij} = 1$ when $i = j$ and $\delta_{ij} = 0$ when $i \neq j$. Let $\sigma(i)$ denote an arbitrary permutation of the integers $\{1, 2, \ldots, m\}$. The $m \times m$ matrix $\mathbf{P}$ with elements $p_{ij} = \delta_{\sigma(i), \sigma(j)}$ is called a permutation matrix.

Property 2: Let $\mathbf{P}$ be a permutation matrix. The following properties hold: (*i*) $\mathbf{P}$ is unitary, i.e. $\mathbf{P}^{-1} = \mathbf{P}^{\mathrm{H}}$, (ii) premultiplying a matrix $\mathbf{R}$ by $\mathbf{P}$ has the effect of transferring the initial row with index $\sigma(i)$ into the new row with index i; (iii) post-multiplying $\mathbf{R}$ by $\mathbf{P}^{\mathrm{T}}$ produces a similar exchange of the columns.

Proof: See Reference 15.

Property 3: The simultaneous permutation of both lines and columns of a Hermitian matrix $\mathbf{R}$, with the same permutation, does not change its condition number.

Proof: According to property 2, such a rearrangement can be expressed as $\mathbf{PRP}^{\mathrm{T}}$, where $\mathbf{P}$ is a permutation matrix. Since $\mathbf{P}$ is unitary, it follows that $\mathbf{PRP}^{\mathrm{T}}$ is a unitary similarity transformation, which does not modify the eigenvalues of $\mathbf{R}$ [15]. Hence $\chi(\mathbf{PRP}^{\mathrm{T}}) = \chi(\mathbf{R})$.

Property 4: Let $\mathbf{Q}$ and $\mathbf{R}$ be $m \times m$ positive-definite Hermitian matrices. Then

$$\chi(\mathbf{Q}^{-1}\mathbf{R}) = \chi(\mathbf{Q}^{-1/2}\mathbf{R}\mathbf{Q}^{-1/2}) \tag{33}$$

Proof: It can be verified easily that $\mathbf{Q}^{-1}\mathbf{R}$ and $\mathbf{Q}^{-1/2}\mathbf{R}\mathbf{Q}^{-1/2}$ have the same characteristic polynomial, i.e., $\det(\mathbf{Q}^{-1}\mathbf{R} - \lambda\mathbf{I}) = \det(\mathbf{Q}^{-1/2}\mathbf{R}\mathbf{Q}^{-1/2} - \lambda\mathbf{I})$. Hence, $\mathbf{Q}^{-1}\mathbf{R}$ and $\mathbf{Q}^{-1/2}\mathbf{R}\mathbf{Q}^{-1/2}$ have the same eigenvalues and thus the same condition number. This completes the proof.

Now, consider the 2D-TDLMS-GSC and the 2D-NDLMS-GSC proposed in Section 3. To compare their convergence rates, suppose that the same separable 2D unitary transform is used in both cases to map the tap-input vector $X(n)$ (eqn. 4) into the transform-domain. Under this condition, the transform-domain coefficients $u_{r,p}(n)$ are given by eqn. 24 and are identical for both realisations. These coefficients are used to update the adaptive processor weights.

In the 2D-TDLMS-GSC, the processor weights are updated via eqn. 12 so that the convergence rate of the adaptive process is governed by the matrix $\hat{\mathbf{R}}_{UU}^{-1}\mathbf{R}_{UU}$, where $\mathbf{R}_{UU}$ is defined in eqn. 13 and $\hat{\boldsymbol{R}}_{UU}^{-1}$ is given by eqns. 15 and 16. In the 2D-NDLMS-GSC, a separate self-orthogonalising LMS adaptive algorithm is used for each value of the index r. According to eqn. 28, the convergence rate of the rth (narrowband) adaptive process is governed by the matrix $\hat{\mathbf{R}}_{U_r}^{-1}{}_{U_r}\mathbf{R}_{U_rU_r}$, where here,

$$\mathbf{R}_{U_rU_r} = E[U_r(n)U_r^{\mathrm{H}}(n)] \tag{34}$$

$$U_r(n) \triangleq [u_{r,1}(n), u_{r,2}(n), \ldots, u_{r,K-1}(n)]^{\mathrm{T}} \tag{35}$$

$$\hat{\mathbf{R}}_{U_r}^{-1}{}_{U_r} = \mathrm{diag}\,(1/\hat{r}_{r,1}, \ldots, 1/\hat{r}_{r,K-1}) \tag{36}$$

and $\hat{r}_{r,k}(n)$ is given by eqn. 15.

Theorem 1: For $r = 0, 1, \ldots, L-1$, there holds the inequality

$$\chi(\hat{\mathbf{R}}_{U_r}^{-1}{}_{U_r}\mathbf{R}_{U_rU_r}) \leqslant \chi(\hat{\mathbf{R}}_{UU}^{-1}\mathbf{R}_{UU}) \tag{37}$$

Proof: We note that $U_r(n)$ (eqn. 35) can be obtained by first permuting the elements of $U(n)$ (eqn. 5) and then retaining only the first $K-1$ elements of the resulting vector. Hence

$$U_r(n) = [\mathbf{I}_{K-1}, 0]\mathbf{P}U(n) \tag{38}$$

where $\mathbf{P}$ is an appropriate permutation matrix and $\mathbf{0}$ is a zero matrix of proper dimension. Using this decomposition, it is not difficult to verify that $\hat{\mathbf{R}}_{U_rU_r}^{-1/2}\mathbf{R}_{U_rU_r}\hat{\mathbf{R}}_{U_rU_r}^{-1/2}$ is a principal submatrix of $\mathbf{P}\hat{\mathbf{R}}_{UU}^{-1/2}\mathbf{R}_{UU}\hat{\mathbf{R}}_{UU}^{-1/2}\mathbf{P}^{\mathrm{T}}$. Now, using properties 4, 1, 3 and 4, in that order, we have

$$\begin{aligned}\chi(\hat{\mathbf{R}}_{U_rU_r}^{-1}\mathbf{R}_{U_rU_r}) &= \chi(\hat{\mathbf{R}}_{U_rU_r}^{-1/2}\mathbf{R}_{U_rU_r}\hat{\mathbf{R}}_{U_rU_r}^{-1/2}) \\ &\leqslant \chi(\mathbf{P}\hat{\mathbf{R}}_{UU}^{-1/2}\mathbf{R}_{UU}\hat{\mathbf{R}}_{UU}^{-1/2}\mathbf{P}^{\mathrm{T}}) \\ &= \chi(\hat{\mathbf{R}}_{UU}^{-1/2}\mathbf{R}_{UU}\hat{\mathbf{R}}_{UU}^{-1/2}) \\ &= \chi(\hat{\mathbf{R}}_{UU}^{-1}\mathbf{R}_{UU})\end{aligned} \tag{39}$$

This completes the proof.