
Martin Bouchard, Member, IEEE, and Stephan Quednau

Abstract—In the last ten years, there has been much research on active noise control (ANC) systems and transaural sound reproduction (TSR) systems. In these fields, multichannel FIR adaptive filters are extensively used. For the learning of FIR adaptive filters, recursive-least-squares (RLS) algorithms are known to produce a faster convergence speed than stochastic gradient descent techniques, such as the basic least-mean-squares (LMS) algorithm or even the fast convergence Newton-LMS, the gradient-adaptive-lattice (GAL) LMS and the discrete-cosine-transform (DCT) LMS algorithms. In this paper, multichannel RLS algorithms and multichannel fast-transversal-filter (FTF) algorithms are introduced, with the structures of some stochastic gradient descent algorithms used in ANC: the filtered-x LMS, the modified filtered-x LMS and the adjoint-LMS. The new algorithms can be used in ANC systems or for the deconvolution of sounds in TSR systems. Simulation results comparing the convergence speed, the numerical stability and the performance using noisy plant models for the different multichannel algorithms will be presented, showing the large gain of convergence speed that can be achieved by using some of the introduced algorithms.

Index Terms—Fast convergence algorithms, multichannel active noise control, transaural sound reproduction.

I. INTRODUCTION

ACTIVE noise control (ANC) systems [1]–[5] and transaural sound reproduction (TSR) systems [6]–[12] have been the subject of a lot a research in recent years. Active noise control works on the principle of destructive interference between an original “primary” disturbance sound field measured at the location of $K$ “error” sensors (typically microphones), and a “secondary” sound field that is generated by $J$ control actuators (typically loudspeakers). Transaural sound reproduction systems (i.e., binaural sound reproduction systems using loudspeakers instead of headphones) share many characteristics of ANC systems, because they also use $J$ actuators to reproduce a target sound field at $K$ sensor locations. In ANC systems (and also in active vibration control systems [13]), a common approach is to use FIR filters as adaptive controllers, in either feedforward or feedback control configurations. The use of FIR filters is also the common way of performing the adaptive deconvolution of sounds in TSR systems. Both ANC and TSR systems usually require multichannel FIR filters. Multichannel adaptive filtering algorithms like the multiple-error LMS algorithm (multichannel filtered-x LMS algorithm) [14] can be used to train those FIR filters. In the case of TSR systems, it is also possible to use Fast Fourier Transforms (FFT) with regularization techniques to compute the compensation (inverse) FIR filters [12]. The FFT approach is fast and has a low computational load. However, the use of time-domain adaptive filtering techniques in TSR systems may be preferable if

- short compensation FIR filters are required (the FFT approach produces long filters);
- learning of the compensation filters is to be performed directly from the plant, and not only from a model of the plant;
- adaptation on a sample by sample basis is required.

For the learning of FIR filters using linear adaptive filtering algorithms, it is well known that recursive-least-squares (RLS) algorithms [15] produce a faster convergence speed than stochastic gradient descent techniques, such as the basic least-mean-squares (LMS) algorithm [16] or even the fast convergence Newton-LMS [16], the gradient-adaptive-lattice LMS (GAL) [15] and the DCT-LMS algorithms [17], [18]. Some uses of monochannel recursive-least-squares algorithms for active noise control have previously been published [5], [19], [20]. In this paper, these works are extended with the introduction of multichannel RLS algorithms and low-computational multichannel fast-transversal-filter (FTF) algorithms, with the structures of some stochastic gradient descent algorithms used in ANC: the filtered-x LMS [16], the modified filtered-x LMS [22]–[25] and the adjoint-LMS [21]. The RLS-based introduced algorithms are thus called multichannel filtered-x RLS, modified filtered-x RLS and adjoint-RLS algorithms, and their low computational FTF-based versions are called the multichannel filtered-x FTF, modified filtered-x FTF and adjoint-FTF algorithms. These algorithms can be used either in ANC systems or for the deconvolution of sounds in TSR systems. Since the multichannel RLS-based algorithms can suffer from potential ill-conditioning of the correlation matrix, and because the multichannel FTF-based algorithms also have a potential for numerical instability, some heuristics to stabilize the algorithms will be discussed in the paper.

In Section II, a brief review of the multichannel filtered-x LMS, modified filtered-x LMS and adjoint-LMS algorithms is presented. Section III describes the RLS equivalents of

Manuscript received August 3, 1998; revised November 15, 1999. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Dennis R. Morgan.

The authors are with the School of Information Technology and Engineering, University of Ottawa, Ottawa, ON, KIN 6N5 Canada (e-mail: bouchard@site.uottawa.ca).

Publisher Item Identifier S 1063-6676(00)06918-2.
these algorithms: the multichannel filtered-x RLS, modified filtered-x RLS and adjoint-RLS algorithms. In Section IV, the multichannel FTF-based algorithms are presented. In Section V, the number of multiplies required for an iteration of each discussed multichannel algorithm is presented, as an indication of the computational load of the algorithms. In Section VI, simulations of a multichannel ANC system compare the convergence speed, the numerical stability and the robustness to plant-model errors of the different multichannel LMS, RLS, and FTF-based algorithms described in the paper.

II. MULTICHANNEL STOCHASTIC GRADIENT DESCENT ALGORITHMS

A. Multichannel Filtered-x LMS Algorithm

In ANC systems, the multichannel version of the filtered-x LMS algorithm [14] is the benchmark to which most adaptive filtering algorithms are compared, because it is widely used. In this algorithm, the gradient of the instantaneous sum of squared error signals is computed as a function of each coefficient in the adaptive filters. The algorithm simply uses a steepest descent approach, and thus it modifies the coefficients in the opposite direction of the gradients. Figs. 1 and 2 show two simplified block-diagrams of the filtered-x LMS algorithm, one for an ANC system and the other for a TSR system. Note that Figs. 1 and 2 show monochannel versions of the algorithm, and that some elements such as actuators and sensors are not shown, to keep the diagrams simple. In the filtered-x LMS algorithm, changes to the adaptive filters do not have an immediate impact on the error signals if there is a delay in the plant. This causes the additional requirement that a small convergence gain $\mu$ may be required in the algorithm, so that the coefficients of the adaptive filters change slowly (stationary system hypothesis of the filtered-x LMS algorithm; see [16]). To describe the multichannel filtered-x LMS algorithm and the other algorithms to be described, the following notation is defined:

- number of reference sensors in an ANC system, or number of source signals to reproduce in a TSR system;
- number of actuators in an ANC or a TSR system;
- number of error sensors in an ANC or a TSR system (in TSR systems, $K$ is typically equal to $I$);
- length of the FIR adaptive filters;
- length of (fixed) FIR filters modeling the plant between the actuators and the error sensors in an ANC or a TSR system;
- value at time $n$ of the $i$th reference signal (ANC systems), or value of the $i$th source signal (TSR systems);
- value at time $n$ of the $i$th actuator signal;
- value at time $n$ of the primary sound field at the $k$th sensor (ANC systems), or value of the $k$th target signal (TSR systems, in this case $d_k(n)$ is typically a delayed version of the source signal $x_k(n)$);
- value at time $n$ of the $i$th error sensor (in ANC systems), or the difference between $d_k(n)$ and the reproduced sound signal at the $k$th error sensor (in TSR systems);
- value at time $n$ of the $l$th coefficient in the adaptive FIR filter linking $x_i(n)$ and $y_j(n)$;
- value of the $m$th coefficient in the (fixed) FIR filter modeling the plant between $y_j(n)$ and $c_k(n)$;
- value at time $n$ of the filtered reference signal, i.e., the signal obtained by filtering the $x_i(n)$ signal with the plant model $h_{j,k}$ filter [see (2)].

In the notation for ANC systems, feedforward controllers have been implicitly assumed because of the use of reference signals $x_i(n)$. However, it is well known that feedback ANC systems using the internal model control (IMC) approach behave like feedforward controllers [26], [27], and the same adaptive filtering algorithms as in feedforward systems can be used for those feedback systems. Using the above notation, the multichannel filtered-x LMS algorithm can be described by the following equations:

$$w_{i,j}(n) = \left[ w_{i,j,1}(n), w_{i,j,2}(n), \ldots, w_{i,j,I}(n) \right]^T$$

$$h_{j,k} = \left[ h_{j,k,1}, h_{j,k,2}, \ldots, h_{j,k,M} \right]^T$$

$$v_{i,j,k}(n) = \left[ v_{i,j,k}(n), v_{i,j,k}(n-1), \ldots, v_{i,j,k}(n-L+1) \right]^T$$

$$x_i(n) = [x_i(n), x_i(n-1), \ldots, x_i(n-L+1)]^T$$

$$x'_i(n) = [x_i(n), x_i(n-1), \ldots, x_i(n-M+1)]^T$$

$$y_j(n) = [y_j(n), y_j(n-1), \ldots, y_j(n-M+1)]^T.$$
For ANC systems, the update equation of the multichannel filtered-x LMS algorithm is

$$w_{i,j}(n + 1) = w_{i,j}(n) - \mu \sum_{k=1}^{K} v_{i,j,k}(n)c_k(n)$$  \hspace{1cm} (9)

where $\mu$ is a scalar convergence gain. In the case of TSR systems, the negative sign becomes a positive sign in the equation, reflecting the different summation signs in Figs. 1 and 2. The following equation describes the behavior of the plant, if the $h_{i,j,k}$ models of the plant are exact

$$c_k(n) = d_k(n) + \sum_{j=1}^{J} h_{i,j,k}^T y_j(n).$$  \hspace{1cm} (10)

Again, (10) is valid for ANC systems. For TSR systems, the positive sign in (10) becomes a negative sign. Also, for ANC systems, the $c_k(n)$ signals are measured from error sensors, while for TSR systems the terms $\sum_{j=1}^{J} h_{i,j,k}^T y_j(n)$ are measured and the $c_k(n)$ signals are computed using (10) with the negative sign.

### B. Multichannel Modified Filtered-x LMS Algorithm

In ANC systems, the multichannel modified filtered-x LMS algorithm uses the plant models $h_{i,j,k}$ to subtract from the error signals the contribution of the actuators, so that estimates of the primary field signals $d_k(n)$ are obtained [22]–[25]. In the case of TSR systems, since the target signals $d_k(n)$ are usually known (they are typically delayed versions of the source signals $x_k(n)$), it is not required to estimate them. The modified filtered-x LMS algorithm then performs a commutation of the plant and the adaptive filters, so that the adaptive filters try to predict the estimation of the $d_k(n)$ signals (instead of the original $d_k(n)$ signals). As a result, if the $h_{i,j,k}$ FIR model filters are accurate, the behavior of the identical to the standard LMS algorithm: any change to the adaptive filter has an immediate effect on the new error signals $d_k(n)$ can be used as compared to the filtered-x LMS and the adjoint-LMS algorithms. The structure of the modified filtered-x LMS algorithm is shown in Figs. 3 and 4, with simplified monochannel block-diagrams of the algorithm for ANC and TSR systems. Defining the following variables:

- $d_k(n)$: estimate of $d_k(n)$
- $d_k^F(n)$: error computed with the commutation of the plant and the adaptive filters (as in Figs. 3 and 4);

the multichannel modified filtered-x LMS algorithm can then be described by (7), (8), and (11)–(13) for ANC systems (or (14)–(15) for TSR systems):

for ANC systems as follows:

$$d_k(n) = c_k(n) - \sum_{j=1}^{J} h_{i,j,k}^T y_j(n).$$  \hspace{1cm} (11)

$$d_k^F(n) = d_k(n) + \sum_{i=1}^{I} \sum_{j=1}^{J} w_{i,j,k}^T(n)v_{i,j,k}(n).$$  \hspace{1cm} (12)

and for TSR systems as follows:

$$w_{i,j}(n + 1) = w_{i,j}(n) + \mu \sum_{k=1}^{K} v_{i,j,k}(n)c_k^F(n)$$  \hspace{1cm} (13)

and for TSR systems as follows:

$$c_k^F(n) = d_k(n) - \sum_{j=1}^{J} h_{i,j,k}^T y_j(n)$$  \hspace{1cm} (14)

$$w_{i,j}(n + 1) = w_{i,j}(n) + \mu \sum_{k=1}^{K} v_{i,j,k}(n)c_k^F(n).$$  \hspace{1cm} (15)

Low computational exact versions of the multichannel filtered-x LMS and the multichannel modified filtered-x LMS algorithms have recently been introduced [25]. However, since the main focus of this paper is on convergence speed, the original versions of the algorithms were presented in this paper.

### C. Multichannel Adjoint-LMS Algorithm

The multichannel adjoint-LMS algorithm was recently introduced as a low-computational alternative to the multichannel filtered-x LMS algorithm [21]. Instead of filtering the reference signals or source signals $x_k(n)$ by the plant models $h_{i,j,k}$, it is the error signals $c_k(n)$ that are filtered by an adjoint (reverse) version of the plant models: $h_{i,j,k}^T$. The equations of the multichannel adjoint-LMS algorithm can be found by

- computing the gradient of the sum of squared-errors at times $n-M+1, n-M+2, \ldots n-1, n$ as a function of the adaptive filter coefficients at time $n-M+1$. Note that this means the computation of only one gradient, and not $M$ different gradients [28]:

![Fig. 3. Simplified block-diagram of an ANC system using the modified filtered-x LMS algorithm.](image1)

![Fig. 4. Simplified block-diagram of a TSR system using the modified filtered-x LMS algorithm.](image2)
Figs. 5 and 6 show simplified monochannel block-diagrams of ANC and TSR systems using the adjoint-LMS algorithm. If the following notation is defined:

\[ e_k(n) = [e_k(n), e_k(n-1), \ldots, e_k(n-M+1)]^T \]  

(17)

the multichannel adjoint-LMS algorithm can then be described by (7) and (18)–(19):

\[ e_j(n-M+1) = \sum_{k=1}^{K} [h_{jk,k}(n) e_k(n)] \]  

(18)

\[ w_{ij}(n+1) = w_{ij}(n) - \mu [e_j(n-M+1) \times e_j(n-M+1)]^T \]  

(19)

Equation (19) is valid for ANC systems. In the case of TSR systems, the negative sign becomes positive. In the multichannel adjoint-LMS algorithm, if there is a delay in the plant, then a modification to the coefficients of the adaptive filters does not have an immediate impact on the error signals \( e_k(n) \). If there is also a delay in the adjoint model (16), then the impact on the filtered-error signals \( e_j(n-M+1) \) is even more delayed. A small gain \( \mu \) may thus be required (stationary system hypothesis) if the delay of the plant is large or if the reverse adjoint model of the plant has a large delay.

III. MULTICHANNEL RECURSIVE-LEAST-SQUARES ALGORITHMS

The classical equations for the monochannel RLS algorithm [15] can be adapted in a straightforward way to produce multichannel algorithms for ANC and TSR systems, as described in this section.

A. Multichannel Filtered-x RLS and Modified Filtered-x RLS Algorithms

To update the coefficients of the adaptive filters, the multichannel filtered-x RLS algorithm uses an approach similar to the multichannel filtered-x LMS algorithm in the update equation (9), but a gain matrix is used to multiply the error signals [as shown by (25)], instead of directly using the filtered-reference signals. The multichannel filtered-x RLS algorithm combines a decorrelation of the filtered-reference signals (reducing the effect of the eigenvalue spread in the filtered-reference signals correlation matrix) and a minimization of a weighted sum of the past squared errors. The procedure to modify the classical RLS algorithm to produce the multichannel filtered-x RLS is the following.

- Each input sample in the input vector of the classical RLS algorithm is replaced by a matrix, to produce the multichannel filtered-reference matrix \( V(n) \).

\[ V(n) = \begin{bmatrix} v_{1,1}(n) & \cdots & v_{1,K}(n) \\ \vdots & \ddots & \vdots \\ v_{L,1}(n) & \cdots & v_{L,K}(n) \\ v_{1,1}(n-L+1) & \cdots & v_{1,K}(n-L+1) \\ \vdots & \ddots & \vdots \\ v_{L,1}(n-L+1) & \cdots & v_{L,K}(n-L+1) \end{bmatrix} \]  

(20)

The number of rows in \( V(n) \) reflects the total number of coefficients in all the adaptive filters (degrees of freedom in the system), while the number of columns reflects the number of error signals to be minimized by the adaptive filters.

- Each coefficient in the weight vector (adaptive filter) in the classical RLS algorithm is replaced by a \( L \times K \) matrix, to produce the multichannel weight vector \( w(n) \) (see (21), shown at the bottom of the page).

\[ w(n) = \begin{bmatrix} w_{1,1}(n) & \cdots & w_{1,L}(n) \\ \vdots & \ddots & \vdots \\ w_{K,1}(n) & \cdots & w_{K,L}(n) \end{bmatrix} \]  

(21)

The interlaced notation of (20) and (21) is not the only possibility to represent the multichannel signals and the adaptive filters. However, with this notation, \( V(n) \) can be computed by...
shifting down $\mathbf{V}(n-1)$, discarding the oldest samples, and adding the new samples, which is a requirement if low-computational versions of the RLS algorithms (as in Section IV) are to be developed. The multichannel filtered-x RLS algorithm is defined by (7), (8), and (23)–(25)

$$
\mathbf{K}(n) = \lambda^{-1} \mathbf{P}(n) \mathbf{V}(n)(\mathbf{I} + \lambda^{-1} \mathbf{V}^T(n) \mathbf{P}(n) \mathbf{V}(n))^{-1} \tag{23}
$$

$$
\mathbf{P}(n+1) = \lambda^{-1} \mathbf{P}(n) - \lambda^{-1} \mathbf{K}(n) \mathbf{V}^T(n) \mathbf{P}(n) \tag{24}
$$

$$
\mathbf{w}(n+1) = \mathbf{w}(n) - \mu \mathbf{K}(n) \mathbf{e}^T(n). \tag{25}
$$

Equation (25) is valid for ANC systems. In the case of TSR systems, the negative sign becomes positive. A gain scalar $\mu$ was introduced in (25), because modifications to the coefficients of $\mathbf{w}(n)$ must be reduced if there is a large delay in the plant, as previously discussed for the filtered-x LMS algorithm (stationary system hypothesis). This can provide stability, at the price of a reduced convergence speed. Note that with the use of the gain scalar $\mu$, the multichannel filtered-x RLS algorithm is not an exact recursive-least-squares algorithm in a strict sense, because the exact solution of the least-squares equations is not computed at every iteration of the algorithm. This comment is also valid for some other algorithms to be introduced in this section and in Section IV.

The multichannel modified filtered-x RLS algorithm is very similar to the multichannel filtered-x RLS algorithm. However, the modified filtered-x RLS algorithm performs the commutation of the plant models and the adaptive filters [(11) and (12) for ANC systems, or (14) for TSR systems], and it computes the estimated error signals $e'_j(n)$, just like the multichannel modified filtered-x LMS algorithm. If the models of the plant (filters) are accurate, then the modified filtered-x RLS algorithm behaves like a multichannel version of the standard RLS algorithm, and a good convergence speed can be expected. The multichannel modified filtered-x RLS algorithm is thus defined by (7), (8), (11), and (12) [or (14)], (23), (24), and (27), using the following definition for the error vector:

$$
\mathbf{e}'(n) = [e'_1(n), e'_2(n), \ldots, e'_K(n)] \tag{26}
$$

$$
\mathbf{w}(n+1) = \mathbf{w}(n) - \mu \mathbf{K}(n) \mathbf{e}'^T(n). \tag{27}
$$

Equation (27) is valid for ANC systems. In the case of TSR systems, the negative sign becomes positive. If the $\mathbf{h}_{1,k}$ models of the plant are accurate, the use of a gain scalar in (27) is not required, because there is no stationary system hypothesis in the modified filtered-x algorithms. A value of $\mu = 1,0$ is thus an optimal value in this algorithm (for the convergence speed), and in that case the exact solution of the least-squares equations is computed at every iteration of the algorithm.

### B. Multichannel Adjoint-RLS Algorithm

The multichannel adjoint-RLS algorithm uses the same filtered-error signals $e'_j(n-M+1)$ that are used by the multichannel adjoint-LMS algorithm, but a decorrelated gain vector is used in the coefficients update equation [as shown by (33)], instead of the $x_i(n)$ reference signals. This approach reduces the effect of the eigenvalue spread in the reference signals correlation matrix, and it minimizes a weighted sum of the past squared filtered-error signals. If the $x_i(n)$ signals were uncorrelated white noise signals, the eigenvalue spread of the correlation matrix would already be reduced, and the convergence speed gain over the multichannel adjoint-LMS algorithm would then be small. Also, since the statistics of the minimized filtered-error $e'_j(n-M+1)$ signals usually differ from the statistics of the error signals $e_k(n)$, the convergence speed of the multichannel adjoint-RLS algorithm will typically be sub-optimal (slower) compared to the filtered-x RLS and the modified filtered-x RLS algorithm. However, as described in [21] and, later in this paper, the adjoint approach requires less computations, so its use may still be justified in some systems. For the multichannel adjoint-RLS, the procedure to adapt the classical RLS algorithm is the following.

- Each input sample in the input vector of the classical RLS algorithm is replaced by a $I \times 1$ vector to produce the multichannel reference vector $\mathbf{x}(n)$ (see (28), shown at the bottom of the page).
- Each coefficient in the weight vector (adaptive filter) of the classical RLS algorithm is replaced by a $J \times 1$ matrix to produce the multichannel weight matrix $\mathbf{W}(n)$.

$$
\mathbf{W}(n) = \begin{bmatrix}
  w_{1,1,1}(n) & \cdots & w_{1,J,1}(n) \\
  \vdots & \ddots & \vdots \\
  w_{I,1,1}(n) & \cdots & w_{I,J,1}(n)
\end{bmatrix}, \tag{29}
$$

- Scalar error signal $e(n)$ is replaced by a multichannel $J \times 1$ vector to produce the multichannel filtered-error vector $\mathbf{e}'(n-M+1)$

$$
\mathbf{e}'(n-M+1) = [e'_1(n-M+1), e'_2(n-M+1), \ldots, e'_J(n-M+1)]^T. \tag{30}
$$

The adjoint-RLS algorithm is thus defined by (7), (18), and (31)–(33)

$$
\mathbf{k}(n) = \lambda^{-1} \mathbf{P}(n) \mathbf{x}(n-M+1)(1 + \lambda^{-1} \mathbf{x}^T(n-M+1) \mathbf{P}(n) \mathbf{x}(n-M+1))^{-1} \tag{31}
$$

$$
\mathbf{P}(n+1) = \lambda^{-1} \mathbf{P}(n) - \lambda^{-1} \mathbf{k}(n) \mathbf{x}^T(n-M+1) \mathbf{P}(n) \tag{32}
$$

$$
\mathbf{x}(n) = \begin{bmatrix}
x_1(n) & \cdots & x_I(n) \\
\vdots & \ddots & \vdots \\
x_1(n-L+1) & \cdots & x_I(n-L+1)
\end{bmatrix}^T \tag{28}
$$
\( \mathbf{W}(n+1) = \mathbf{W}(n) - \mu \mathbf{k}(n) \mathbf{x}^T(n-M+1). \)  

Equation (33) is valid for ANC systems. In the case of TSR systems, the negative sign becomes positive.

### C. Stability of RLS-Based Algorithms

Since the RLS-based algorithms compute recursively the value of the inverse of a time-averaged correlation matrix, instability can occur if the correlation matrix is ill-conditioned. In multichannel ANC and TSR systems, the ill-conditioning may be caused by some coupling between different “paths” in the plant, or between the different reference signals, or simply by the finite numerical resolution. To try to avoid these potential problems, an approach is to add a small amount of noise to the filtered-reference signals \( \mathbf{v}_{j,k}(n) \) (in (23) and (24) for the filtered-x RLS or the modified filtered-x RLS algorithms) or to the \( x_p(n) \) reference signals (in (31) and (32) for the adjoint-RLS algorithm). This causes a bias to the solution found by the RLS-based algorithms and slows down the convergence, but it improves the numerical stability of the algorithms.

### IV. Multichannel Fast-Transversal-Filter Algorithms

The computational load of the algorithms introduced in the previous section is proportional to \( P^2, J^2, K \) and \( L^2 \) for the multichannel filtered-x RLS and modified filtered-x RLS algorithms, and proportional to \( P^2 \) and \( L^2 \) for the multichannel adjoint-RLS algorithm (see Section V for a description of the computational load). This computational load may be too high for some applications, so lower computational (fast) equivalents of these algorithms will now be presented. Two types of algorithms were considered: fast-transversal-filter (FTF) and least-square-lattices (LSL) algorithms [15]. FTF algorithms are sometimes referred to as computationally reduced versions of the original fast Kalman or fast RLS algorithms [29]–[31]. The LSL algorithms (in particular the versions using error feedback) are known to be a very numerically robust algorithms, while FTF algorithms are known to have typically bad numerical properties [15]. However, for multichannel ANC and TSR applications, FTF algorithms may be more useful than LSL algorithms because of the following drawbacks of the LSL.

- LSL algorithms require the knowledge of the desired (target) signals \( d_k(n) \), which may be unknown in the case of ANC systems (with the filtered-x and adjoint algorithms).
- The \( \mathbf{v}_{j,k}(n) \) coefficients are not computed by the LSL algorithms. Instead, coefficients in a time-varying orthogonal transformed basis are computed. Using directly the structure of the LSL algorithms (i.e., the forward and backward lattice predictors, and the joint-process estimator) for the computation of the actuator signals \( y_j(n) \) would not be possible in a multichannel filtered-x LSL or a modified filtered-x LSL algorithm. This is because the joint-process estimation of those LSL-based algorithms works on transformed signals that have the dimensions of the filtered-reference signals matrix \( \mathbf{V}(n) \), which are not the same as the dimensions of the reference signals vector \( \mathbf{x}(n) \).
- An inverse transformation from the coefficients in the time-varying orthogonal transformed basis to the \( \mathbf{v}_{j,k}(n) \) coefficients would thus be required. This transformation exists [15], but it is costly to compute the transformation matrix and to perform the transformation.
- Computational load of LSL algorithms is higher than for FTF algorithms [15].

Because of the above drawbacks of LSL-based algorithms, an FTF algorithm will be used in this section to develop low-computational equivalents of the multichannel RLS-based algorithms of the previous section. Classical FTF algorithms typically become unstable after a finite number of iterations, but it is possible to predict this instability by looking at the value of a “rescue variable” [15], which can be computed in these algorithms. Whenever the value of the rescue variable is outside of a specific range (typically between 0.0 and 1.0 + \( \varepsilon \), where \( \varepsilon \) is a small value), then the FTF algorithm must stop the update of the \( \mathbf{v}_{j,k}(n) \) coefficients, reinitialize its variables (but not the \( \mathbf{v}_{j,k}(n) \) coefficients), and then restart the update of the \( \mathbf{v}_{j,k}(n) \) coefficients. Instead of using a rescue variable, another approach to stabilize the FTF algorithm is to introduce error feedback in the algorithm [32].

Computationally reduced FTF algorithms use a variable called the “conversion factor,” that can be used to convert \( \text{a priori} \) prediction errors to \( \text{a posteriori} \) prediction errors. Although this definition for the conversion factor is also valid in multichannel algorithms, some of the recurrent equations used to compute the innovation factor do not generalize to the multichannel case (the dimensions of the matrix terms in these equations do not match). Therefore, it was required to use the original fast-Kalman/fast-RLS algorithm from [29], [30] as the FTF algorithm to be generalized to the multichannel case. The ratio of computations of the original monochannel fast-Kalman/fast-RLS algorithm compared to a computationally reduced FTF version is approximately 10 : 7.

#### A. Multichannel Filtered-x FTF and Multichannel Modified Filtered-x FTF Algorithms

The procedure to adapt the fast-RLS algorithm from [29] to multichannel filtered-x FTF and multichannel modified filtered-x FTF algorithms will now be described. The dimension changes required on the input signal, the weight vector (adaptive filter) and the error signal of the FTF algorithm to produce the \( \mathbf{V}(n), \mathbf{w}(n), \) and \( \mathbf{e}(n) \) vectors and matrices are exactly the same as in the previous section for the multichannel filtered-x RLS algorithm. Some additional variables have to be transformed.

- \( \text{a priori} \) and \( \text{a posteriori} \) forward prediction errors at time \( n \) become the \( IJ \times K \) matrices \( \mathbf{T}(n) \) and \( \mathbf{T}^T(n) \).
- Backward prediction errors at time \( n \) becomes a \( IJ \times K \) matrix \( \mathbf{T}(n) \).
- Forward predictor filter at time \( n \) becomes a \( IJ(L-1) \times IJ \) matrix \( \mathbf{F}(n) \).
- Backward predictor filter at time \( n \) becomes a \( IJ(L-1) \times IJ \) matrix \( \mathbf{B}(n) \).
- Minimum value of the sum of weighted forward \( \text{a posteriori} \) prediction errors at time \( n \) becomes a \( IJ \times IJ \) matrix \( \mathbf{Q}(n) \).
• Gain matrix of prediction order \( L \) at time \( n \) becomes a \( IJL \times K \) matrix \( \mathbf{K}_L(n) \).
• Gain matrix of prediction order \( L - 1 \) at time \( n \) becomes a \( IJ(L - 1) \times K \) matrix \( \mathbf{K}_{L-1}(n) \).
• Vector containing the last \( L - 1 \) input samples becomes the \( IJ(L - 1) \times K \) matrix \( \mathbf{V}_{L-1}(n) \) of (34), shown at the bottom of the page, containing the \( L - 1 \) last samples of all the \( v_{i,j,k}(n) \) filtered-reference signals.

The filtered-x FTF algorithm can be described by (7), (8), and (35)–(43):

\[
\Psi(n) = \begin{bmatrix} v_{1,1,1}(n) & \cdots & v_{1,1,K}(n) \\
\vdots & \ddots & \vdots \\
v_{L,J,1}(n) & \cdots & v_{L,J,K}(n) \end{bmatrix} + \mathbf{F}^T(n-1)\mathbf{V}_{L-1}(n-1) \\
\Phi(n) = \begin{bmatrix} v_{L+1,1}(n-L+1) & \cdots & v_{L+1,K}(n-L+1) \\
\vdots & \ddots & \vdots \\
v_{L,J,1}(n-L+1) & \cdots & v_{L,J,K}(n-L+1) \end{bmatrix} + \mathbf{B}^T(n-1)\mathbf{V}_{L-1}(n) \\
\Omega(n) = \lambda\Omega(n-1) + \Psi(n)\Phi^T(n) \\
\mathbf{K}_L(n) = \begin{bmatrix} \mathbf{C}_{L-1}(n) \\
\mathbf{C}_1(n) \end{bmatrix} = \begin{bmatrix} 0 \\
\mathbf{F}(n) \end{bmatrix} \\
\mathbf{K}_{L-1}(n) = (\mathbf{C}_{L-1}(n) - \mathbf{B}(n-1)\mathbf{C}_1(n)) \times (\mathbf{I} - \mathbf{F}^T(n)\mathbf{C}_1(n))^{-1} \\
\mathbf{B}(n) = \mathbf{B}(n-1) - \mathbf{K}_{L-1}(n)\Phi^T(n) \\
\mathbf{w}(n+1) = \mathbf{w}(n) - \mu\mathbf{K}_L(n)\mathbf{e}(n)^T. 
\]

Equation (44) is valid for ANC systems. In the case of TSR systems, the negative sign becomes positive. As in the modified filtered-x RLS, the gain scalar in (44) may be set to 1.0 if the \( \mathbf{h}_{i,j,k} \) models of the plant are accurate (no stationary system hypothesis, and exact computation of the solution of the least-squares equations).

**B. Multichannel Adjoint-FTF Algorithm**

The procedure to adapt the fast-RLS algorithm from [29] to the multichannel adjoint-FTF will now be described. The dimension changes required on the input signal, the weight vector (adaptive filter) and the error signal of the FTF algorithm to produce the \( \hat{\mathbf{x}}(n) \), \( \mathbf{W}(n) \) and \( \mathbf{e}^T(n-M+1) \) vectors and matrices are exactly the same as for the multichannel adjoint-RLS. The following variables also have to be transformed.

- \textit{A priori} and \textit{a posteriori} forward prediction errors at time \( n \) become \( I \times 1 \) vectors \( \hat{\psi}(n) \) and \( \psi^T(n) \).
- Backward prediction errors at time \( n \) becomes an \( I \times 1 \) vector \( \phi(n) \).
- Forward predictor filter at time \( n \) becomes a \( I(L-1) \times I \) matrix \( \mathbf{F}(n) \).
- Backward predictor filter at time \( n \) becomes a \( I(L-1) \times I \) matrix \( \mathbf{B}(n) \).
- Minimum value of the sum of weighted forward \textit{a posteriori} prediction errors at time \( n \) becomes a \( I \times I \) matrix \( \mathbf{Q}(n) \).
- Gain vector of prediction order \( L \) at time \( n \) becomes a \( IL \times 1 \) vector \( \mathbf{k}_L(n) \).
- Gain vector of prediction order \( L - 1 \) at time \( n \) becomes the \( I(L-1) \times 1 \) vector \( \mathbf{k}_{L-1}(n) \).
- Vector containing the last \( L - 1 \) input samples of the adaptive filter becomes the \( I(L-1) \times 1 \) vector \( \mathbf{x}_{L-1}(n) \) of (45),

\[
\mathbf{V}_{L-1}(n) = \begin{bmatrix} v_{1,1,1}(n) & \cdots & v_{1,1,K}(n) \\
\vdots & \ddots & \vdots \\
v_{L,J,1}(n) & \cdots & v_{L,J,K}(n) \\
v_{L+1,1}(n-L+2) & \cdots & v_{L+1,K}(n-L+2) \\
\vdots & \ddots & \vdots \\
v_{L,J,1}(n-L+2) & \cdots & v_{L,J,K}(n-L+2) \end{bmatrix} 
\]
containing the \(L-1\) last samples of all the \(x_i(n)\) reference signals (see (45), shown at the bottom of the page).

As in the multichannel adjoint-RLS algorithm, the multichannel adjoint-FTF algorithm uses the filtered-error signals \(c_i(n-M+1)\) introduced with the adjoint-LMS algorithm. It also uses a decorrelated gain vector \(k_{L-1}(n)\) for the update of the coefficients in the adaptive filters. The adjoint-FTF algorithm is defined by (7), (18) and (46)–(54)

\[
\psi(n) = \begin{bmatrix} x_1(n - M + 1) \\ \vdots \\ x_j(n - M + 1) \end{bmatrix} + F^T(n-1)x_{L-1}(n - M) \tag{46}
\]

\[
\varphi(n) = \begin{bmatrix} x_1(n - L - M + 2) \\ \vdots \\ x_j(n - L - M + 2) \end{bmatrix} + B^T(n-1)x_{L-1}(n - M + 1) \tag{47}
\]

\[
F(n) = F(n-1) - k_{L-1}(n-1)\psi^T(n) \tag{48}
\]

\[
\psi'(n) = \begin{bmatrix} x_1(n - M + 1) \\ \vdots \\ x_j(n - M + 1) \end{bmatrix} + F^T(n)x_{L-1}(n - M) \tag{49}
\]

\[
\Omega(n) = \lambda \Omega(n-1) + \psi'(n)\psi'^T(n) \tag{50}
\]

\[
k_L(n) = \begin{bmatrix} c_{L-1}(n) \\ c_2(n) \end{bmatrix} + \begin{bmatrix} I \\ F(n) \end{bmatrix} \Omega^{-1}(n)\psi'(n) \tag{51}
\]

\[
k_{L-1}(n) = (c_{L-1}(n) - B(n-1)c_2(n)) \times (1 - \varphi^T(n)c_2(n))^{-1} \tag{52}
\]

\[
B(n) = B(n-1) - k_{L-1}(n)\varphi^T(n) \tag{53}
\]

\[
W(n+1) = W(n) - \mu k_L(n)\varphi^T(n-M+1) \tag{54}
\]

Equation (54) is valid for ANC systems. In the case of TSR systems, the negative sign becomes positive. Note that there is a \(I \times I\) matrix inversion in the multichannel adjoint-FTF algorithm, and in some systems it may be required to perform some special processing (for example pseudo-inversing) if the matrix is ill-conditioned.

C. Rescue Variables

As mentioned earlier, it is possible to use a “rescue variable” that will indicate when an FTF-based algorithm is about to diverge, if the rescue variable has a value outside of a given range. In the case of the multichannel filtered-x FTF and modified filtered-x FTF algorithms, the rescue variable is the scalar term \(1 - \varphi^T(n)c_2(n)\) in (52), and again it can be used to monitor the condition of the algorithm.

V. COMPUTATIONAL COMPLEXITY OF THE ALGORITHMS

The number of multiplies required for an iteration of each multichannel algorithm presented in this paper was evaluated for an ANC system, and the results are shown in Table I. From this table, it can be seen that RLS-based algorithm will have a much higher computational load for large values of \(L\), and also that adjoint structures will have a significantly lower computational load for large values of \(I, J\) and \(K\). In the simulations that will be described in the next section, an ANC system with \(I = 2, J = 3, K = 2, L = 300\) and \(M = 128\) was used. The resulting computational complexity for the different algorithms was

- 6938 multiplies/iteration for the multichannel filtered-x LMS;
- 11 306 multiplies/iteration for the multichannel modified filtered-x LMS;
- 4371 multiplies/iteration for the multichannel adjoint-LMS;
- 16 224 942 multiplies/iteration for the multichannel filtered-x RLS (plus a \(K \times K\) matrix inversion);
- 16 229 310 multiplies/iteration for the multichannel modified filtered-x RLS (plus a \(K \times K\) matrix inversion);
- 1 085 573 multiplies/iteration for the multichannel adjoint-RLS;
- 165 014 multiplies/iteration for the multichannel filtered-x FTF (plus a \(I \times I\) and a \(K \times K\) matrix inversion);
- 169 382 multiplies/iteration for the multichannel modified filtered-x FTF (plus a \(I \times I\) and a \(K \times K\) matrix inversion);
- 1 335 573 multiplies/iteration for the multichannel adjoint-FTF (plus a \(I \times I\) matrix inversion).

These numbers clearly show the high computational load of RLS-based algorithms, and the low computational load of adjoint structures. The complexity of FTF-based algorithms is somewhere between the complexity of LMS-based algorithms and the complexity of RLS-based algorithms.

VI. SIMULATIONS OF THE MULTICHANNEL ALGORITHMS

A. Description of the Simulations

The multichannel filtered-x LMS, modified filtered-x LMS, adjoint-LMS and the RLS and FTF versions of these algorithms (introduced in this paper) were all tested in simulations of an ANC system. The simulations were performed using MatLab™. Some simulations with TSR systems were also performed, with results always similar to the results of ANC systems, therefore

\[
x_{L-1}(n) = \begin{bmatrix} x_1(n) & \cdots & x_j(n) \end{bmatrix} \cdots \begin{bmatrix} x_1(n-L+2) & \cdots & x_j(n-L+2) \end{bmatrix}^T \tag{45}
\]
TABLE I
COMPUTATIONAL COMPLEXITY (NUMBER OF MULTIPLIES) OF THE MULTICHANNEL ALGORITHMS

<table>
<thead>
<tr>
<th>Multichannel algorithm</th>
<th>Number of multiplies (excluding matrix inversions)</th>
<th>Matrix inversions</th>
</tr>
</thead>
<tbody>
<tr>
<td>filtered-x LMS</td>
<td>$L(IJ + JK) + M(JK) + K$</td>
<td>none</td>
</tr>
<tr>
<td>modified filtered-x LMS</td>
<td>$L(IJ + 2JK) + M(IJ + JK) + K$</td>
<td>none</td>
</tr>
<tr>
<td>adjoint-LMS</td>
<td>$L(2IJ) + M(JK) + J$</td>
<td>none</td>
</tr>
<tr>
<td>filtered-x RLS</td>
<td>$L^2(2IJ^2K + I^2J^2) + L(2IJK^2 + 2IJ + IJ)$</td>
<td>one $K \times K$</td>
</tr>
<tr>
<td>modified filtered-x RLS</td>
<td>$L^2(2IJ^2K + I^2J^2) + L(2IJK^2 + 3IJ + IJ)$</td>
<td>one $K \times K$</td>
</tr>
<tr>
<td>adjoint-RLS</td>
<td>$L^2(3I^2) + L(2IJ + 2I) + M(JK) + J + 2$</td>
<td>one scalar inversion</td>
</tr>
<tr>
<td>filtered-x FTF</td>
<td>$L(IJ + JK^2 + IJK) + (L - 1)(7I^2J^2K) + M(IJK)$</td>
<td>one $IJ \times IJ$</td>
</tr>
<tr>
<td>modified filtered-x FTF</td>
<td>$L(IJ + JK^2 + 2IJK) + (L - 1)(7I^2J^2K) + M(IJK + JK)$</td>
<td>one $IJ \times IJ$</td>
</tr>
<tr>
<td>adjoint-FTF</td>
<td>$L(2IJ + I) + (L - 1)(7I^2) + M(JK) + 3I^2 + J$</td>
<td>one $I \times I$</td>
</tr>
</tbody>
</table>

Fig. 7. Example of an impulse response and frequency response used for the transfer functions between the noise source and the primary field signals.

results of TSR systems will not be presented in this section. The simulated system had the dimensions $I = 2$, $J = 3$ and $K = 2$. This is to reflect the well known principle that an additional actuator can greatly help to find a causal solution for an acoustic control system or an inverse acoustic system (MINT algorithm, [33] and [34]). However, this also produces an underdetermined system where the multichannel correlation matrix of the filtered reference signals will be highly ill-conditioned, and this will greatly affect the stability of the least-squares algorithms. The simulations were based on the systems illustrated in Figs. 1, 3, and 5, with the following additions:

- transfer functions were added to generate the reference signals and the primary field signals (disturbance signals) from a “white noise source”;
- for the RLS and FTF algorithms, the update of the coefficients was of course not performed simply with the gradient of the instantaneous squared error, but instead the recursive-least-squares equations of the previous sections were used.

All the transfer functions used in the simulations were acoustic transfer functions experimentally measured in a duct. A total of ten measured transfer functions were used: two between a noise source and the reference signals, six between the actuators and the error sensors (the plant transfer functions), and two between a noise source and the primary field signals. The transfer functions between the noise source and the error sensors had a length of 256 samples with a delay of approximately 35 samples, so that a causal solution would exist for the adaptive FIR filters. Fig. 7 shows the impulse response and frequency response of one of those transfer functions. The other transfer functions that were used (for the plant between the actuators and the error sensors, and for the transfer functions between the noise source and the reference signals) had a length of 128 samples, and a small delay. Fig. 8 shows the impulse response and frequency response of one of those other transfer functions. The length of the adaptive FIR filters was $L = 300$, and Fig. 9 shows a typical impulse response and frequency response of an adaptive FIR filter after convergence. The $\mathbf{h}_{ijk}$ filters modeling the plant transfer functions had the same length as the plant transfer functions: $M = 128$. The length of the adjoint filters $\mathbf{h}_{ijk}^j$ [see (16)] was thus also $M = 128$. The forgetting factor coefficient in the recursive-least-squares algorithms was set to $\lambda = 0.99$. 
For the initialization of the $\mathbf{P}(n)$ matrix (inverse of the time-averaged correlation matrix) in the RLS algorithms, the identity matrix multiplied by a value of $10^6$ was used. For the FTF algorithms, the initial value of $\mathbf{Q}(n)$ was an identity matrix multiplied by $10^6$. The choice of a high value to initialize $\mathbf{Q}(n)$ means a slower initial convergence, but it greatly improved the numerical stability of the FTF algorithms. The value of the gain $\mu$ that produced the fastest convergence speed was found by trial and error for each algorithm. The initial values of the adaptive FIR filters and the predictor filters (in the FTF algorithms) were all set to zero.

As mentioned in Section III, the RLS-based algorithms can become unstable if the correlation matrix of the system is ill-conditioned, and the addition of a random noise signal component in these algorithms can help to solve this problem. This causes a bias on the optimal solution and a loss of convergence speed, but the stability of the RLS-based algorithms can be improved. However, this was not always sufficient to stabilize the RLS-based algorithms in the simulations, as will be discussed later in this section. In the case of the FTF-based algorithms, if no protection using a “rescue variable” was programmed in the algorithms, numerical instability always eventually occurred. A rescue variable was thus used to re-initialize the algorithms when the value of the rescue variable was outside of the range [0, 1,001], and this was found sufficient to always stabilize the FTF-based algorithms.

B. Convergence Speed and Numerical Stability

Fig. 10 shows the performance of the multichannel LMS-based and FTF-based algorithms described in the paper, for the simulated ANC system. The performance is defined by the averaged power of the error signals divided by the averaged power of the primary field signals (disturbance or target signals). All the algorithms were run five times during 50,000 iterations, and the average results are plotted in Fig. 10. The modified filtered-x FTF algorithm clearly produced the best performance (using a step size of 1.0 and a decorrelated filtered reference gain vector), followed by the filtered-x FTF (also using a decorrelated filtered reference gain vector, but with a smaller step size because of the plant delay), then by the modified filtered-x LMS (using a large step size). The three remaining algorithms showed a poor performance: the filtered-x LMS algorithm (using a smaller step size because of the plant delay), the adjoint-FTF (suboptimal, and using a very small step size because of the combined delay of the plant and the adjoint filters $\mathbf{y}_{j,k}$) and the adjoint-LMS (also using a very small step size because of the combined delay of the plant and the adjoint filters). The adjoint-based algorithms produced the worst performance, and one important factor was the large delay of the adjoint plant models. However, for plants that can be modeled with fewer coefficients (for example plant models that are valid at a few frequencies only), for harmonic
control), the performance of the adjoint algorithms could become similar to the performance of the filtered-x algorithms, while having a lower computational load.

It is clear that a large convergence speed gain was achieved with the use of the modified filtered-x FTF or the filtered-x FTF algorithm, over the other algorithms in Fig. 10. However, the performance of the modified filtered-x FTF or the filtered-x FTF algorithm would be even better if it were not for the numerical instability that required frequent re-initialization of the algorithms. Still, as can be observed from Fig. 10, those re-initializations did not cause undesirable transients in the performance of the algorithms. In principle, the multichannel RLS-based algorithms introduced in this paper should provide an even greater convergence speed than the FTF-based algorithms (at the price of a much increased computational load), because they do not require dynamic re-initializations using a rescue variable. This greater convergence speed was indeed observed in the simulations, but it was not possible to both fully stabilize the RLS-based algorithms and keep the fast-convergence behavior. This is because adding noise to some input signals in the RLS-based algorithms (as described before) improves the stability but decreases the convergence speed. Fig. 11 shows the performance of the multichannel RLS-based algorithms when a noise signal with a power of 10% of the input signals was added to the input signals. Fig. 11 also shows the performance of the FTF-based algorithms, and only 5000 iterations are shown, before any divergence of the RLS-based algorithms would occur. From Fig. 11, it can be seen that the modified filtered-x RLS algorithm outperforms the re-initialized modified filtered-x FTF algorithm, even if a 10% input signal noise level was used in the RLS-based algorithm. The filtered-x RLS with a 10% input signal noise level also outperformed the re-initialized filtered-x FTF algorithm, while the adjoint-RLS and re-initialized adjoint-FTF algorithms produced approximately the same poor performance, again mainly because of the adjacent plant model delay.

To fully stabilize the RLS-based algorithms, it was required to increase the input signal noise level in the algorithms, and this caused a convergence speed that was similar or slower than for the FTF-based algorithms. Therefore, the use of the computationally expensive RLS-based algorithms became unjustified in the simulated $I = 2$, $J = 3$, and $K = 2$ underdetermined ANC system. It should be noted that there were some simulated systems and dimensions for which only a very low level of noise on the input signals (or sometimes no noise at all) was required to stabilize the RLS-based algorithms ($I = 2$, $J = 2$, and $K = 1$ system for example). In some further work, it would be interesting to use a numerical stable QR decomposition version or a constrained version of the RLS-based algorithms, so that it becomes unnecessary to add noise to the input signals in the algorithms.

C. Robustness to Plant Model Errors

The adaptive filtering algorithms described in this paper use models of the plant (or adjacent models or the plant), i.e., models of the transfer functions between the actuators and the error sensors. The simulation results that were presented so far in this section were performed with models that were exactly the same as the plant. However, in practice, this is never the case. In order to evaluate the effect of the precision of the plant models on the performance of the algorithms, simulations of the same multichannel ANC system with noisy plant models were performed for the modified filtered-x FTF, filtered-x FTF, modified filtered-x LMS and filtered-x LMS algorithms. These simulations with noisy models show the performance of the algorithms for a particular system and a particular noise component, and further theoretical work and simulations would be required to fully characterize the behavior of the algorithms with noisy models. Nevertheless, the simulations can provide some indications that the algorithms can perform very well using models of limited precision. Simulation results using the exact models were compared with simulations results obtained by using models with a signal to noise ratio (SNR) of 20 dB, 10 dB, and 5 dB. For all algorithms, a SNR ratio of 20 dB produced a performance almost identical to the performance obtained by using the exact models (less than 1 dB of difference in the learning curves).

The performance of the modified filtered-x FTF algorithm using exact and noisy plant models is shown in Fig. 12 (for 10 dB and 5 dB SNR). It can be observed that a SNR of 10 dB on the plant models is sufficient to obtain a convergence speed that is close to the convergence speed with exact models. However, a SNR of only 5 dB produces a learning curve that shows signs of divergence, even with a reduced step size. Fig. 13 shows the same curves for the filtered-x FTF algorithm. In this case, it was required to use a smaller step size to avoid some transients when noisy models were used, and this explains why the 10 dB SNR curve in Fig. 13 shows a significant difference with
the exact model convergence curve, but the performance is still acceptable. As with the previous algorithm, a SNR of only 5 dB on the plant models produced a learning curve with a lower convergence speed and with signs of instability.

The performance of the modified filtered-x LMS algorithm with exact and noisy models is shown in Fig. 14 (for 10 dB and 5 dB SNR). Plant models with a 10 dB SNR reduced the performance of the algorithm, but the algorithm still converged with an acceptable speed. A 5 dB SNR further reduced the convergence speed, and a smaller step size was required to keep the algorithm stable. Finally, Fig. 15 shows the same curves for the filtered-x LMS algorithm. As can be seen from this picture, using 10 dB SNR plant models or 5 SNR plant models did not cause a large decrease of performance for this algorithm. In summary, for a given SNR of the plant models and assuming that the algorithms converge (SNR sufficiently high), the rank of the different algorithms for the convergence speed remained the same as in simulations where exact models were used. Also, a 20 dB SNR on the models proved to be sufficient to obtain a performance almost identical to the performance with exact models, and good performance could be achieved in many cases with models having only a 10 dB SNR.

VII. CONCLUSION

In this paper, multichannel recursive-least-squares algorithms and multichannel fast-transversal-filter algorithms were introduced, for applications in active noise control and transaural sound reproduction. Simulations of a multichannel ANC system showed that some algorithms introduced in this paper can greatly increase the convergence speed of adaptive FIR filters, as compared to common ANC multichannel algorithms such as the filtered-x LMS, the modified filtered-x LMS, and the adjoint-LMS algorithms. The multichannel FTF-based algorithms were successfully numerically stabilized, however for some systems it was not possible to both fully stabilize and keep the fast convergence speed for the multichannel RLS-based algorithms. Some further work on the RLS-based algorithms could thus include numerically stable QR decompositions, realizations, or constained realizations of the algorithms. Simulations with noisy plant models were also performed in the paper, showing that the algorithms with high convergence speed keep their fast convergence properties for plant models having a SNR of 10 dB or more. Some further work on the issue of robustness of the algorithms to plant model errors would be of interest.

REFERENCES


[27] S. J. Elliott, T. J. Sutton, B. Rafaely, and M. Johnson, “Design of feedforward controllers for the Audio Research Group under the supervision of Dr. S. P. Lipshitz. Following his graduation in 1995, he was a Signal Processing Research Assistant with the Group of Acoustics and Vibration, University of Sherbrooke. From 1995 to 1997, he also was with SoftDb Active Noise Control Systems, Inc., which he co-founded. In January 1998, he joined the University of Ottawa, Ottawa, ON, Canada, where he is currently an Assistant Professor with the School of Information Technology and Engineering. His current research interests are signal processing, adaptive filtering, and neural networks applied to speech, audio, and acoustics. Dr. Bouchard is a member of the Ordre des Inégénieurs du Québec, the Audio Engineering Society, and the Acoustical Society of America.

[28] M. Bouchard (S’90–M’98) received the B.Ing., M.Sc., and Ph.D. degrees in electrical engineering from the University of Sherbrooke, Sherbrooke, PQ, Canada, in 1993, 1995, and 1997, respectively.


Stephan Quednau received the B.S. degree in electrical engineering from the University of Waterloo, Waterloo, ON, Canada, in 1995. He is currently pursuing the M.S. degree in electrical engineering at the School of Information Technology and Engineering, University of Ottawa, Ottawa, ON.

While at the University of Waterloo, he investigated idle-channel tones in multibit dithered noise shapers for the Audio Research Group under the supervision of Dr. S. P. Lipshitz. Following his graduation in 1995, he was a Communications Hardware and Software Engineer with the Communications Research Center, Ottawa. His area of interests include adaptive linear and nonlinear signal processing, nonuniform sampling and signal compression with applications to audio signal processing and acoustic noise cancellation.

Mr. Quednau is a member of the Audio Engineering Society.
学霸图书馆
www.xuebalib.com

本文献由“学霸图书馆-文献云下载”收集自网络，仅供学习交流使用。

学霸图书馆（www.xuebalib.com）是一个“整合众多图书馆数据库资源，
提供一站式文献检索和下载服务”的24小时在线不限IP图书馆。
图书馆致力于便利、促进学习与科研，提供最强文献下载服务。

图书馆导航：
图书馆首页 文献云下载 图书馆入口 外文数据库大全 疑难文献辅助工具