RECURSIVE LEAST-SQUARES ALGORITHMS WITH GOOD NUMERICAL STABILITY FOR MULTICHANNEL ACTIVE NOISE CONTROL

F. Yu, M. Bouchard,
School of Information Technology and Engineering, University of Ottawa,
161 Louis-Pasteur, P.O. Box 450, Station A
Ottawa (Ontario), K1N 6N5, e-mail : bouchard@site.uottawa.ca

ABSTRACT

Some recursive least-squares algorithms for multichannel active noise control have recently been introduced, including computationally efficient (i.e. "fast") versions. However, these previously published algorithms suffer from numerical instability due to finite precision computations. In this paper, numerically robust recursive least-squares algorithms for multichannel active noise control systems are introduced, using QR decompositions and lattice structures. It is shown through simulations of broadband multichannel active noise control that the recursive least-squares algorithms introduced in this paper are indeed more numerically robust than the previously published algorithms, while keeping the same convergence behavior, and therefore they are more suitable for practical implementations.

1. INTRODUCTION

In a recently published paper [1], recursive least-squares (RLS) algorithms and fast-transversal-filter (FTF) algorithms were introduced for multichannel active noise control (ANC). It was reported that these algorithms can greatly improve the convergence speed of ANC systems, compared to steepest descent algorithms or their variants, as expected. However, numerical instability of the algorithms was an issue that needed to be resolved. In this paper, extensions of stable realizations of recursive least-squares algorithms such as the inverse QR-RLS algorithm [2] and the QR decomposition least-squares-lattice (QRD-LSL) algorithm [2] are introduced for the specific problem of multichannel ANC, and simulation results will be presented to validate their numerical stability. The algorithms described in this paper will be combined with an ANC structure previously introduced in a modified version of the filtered-x LMS algorithm [3]. This structure is to be called the "modified filtered-x structure" in this paper. This structure is a delay-compensating structure that eliminates the need to reduce the adaptive gain in the update equation of the adaptive filters, because of the delay found in the "error path" between the actuators and the error sensors. It is also a structure that computes estimates of the disturbance signals (i.e. primary field signals), which is a requirement for the QRD-LSL ANC algorithm to be developed in this paper. Figure 1 shows a block diagram of an ANC system with the modified filtered-x structure.

2. INVERSE QR-RLS ALGORITHM FOR MULTICHANNEL ANC SYSTEMS

For broadband multichannel ANC systems, it is important that an adaptive filtering algorithm explicitly computes the time domain coefficients of the filters. For example, looking at Fig. 1 it can be seen that the adaptive filters coefficients in the lower path (adaptation or optimization path) need to be copied to the upper path (the control path). Therefore it is not sufficient to have an algorithm that produces an estimate of a target signal for each iteration of the algorithm, it is also required to have the knowledge of the adaptive filters coefficients. The inverse QR-RLS algorithm does compute explicitly the adaptive filters coefficients [2], and to describe the inverse QR-RLS algorithm for multichannel ANC systems, the following notation is defined:

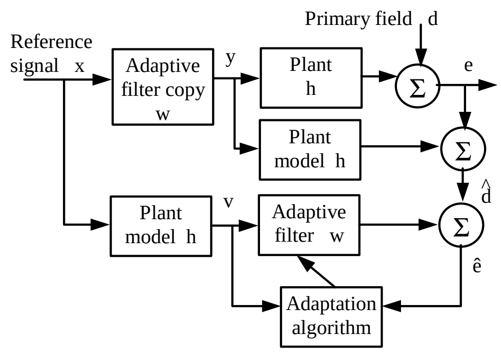

Figure 1. The "modified filtered-x" or delay-compensated structure for ANC

I	number of reference sensors in an ANC system
J	number of actuators in an ANC system
K	number of error sensors in an ANC system
L	length of the FIR adaptive filters
M	length of FIR filters modeling the plant (transfer functions between the actuators and the error sensors in an ANC system)
$x_i(n)$	value at time n of the i^{th} reference sensor signal
$y_j(n)$	value at time n of the j^{th} actuator signal
$d_k(n)$	value at time n of the primary sound field at the k^{th} error sensor
$e_k(n)$	value at time n of the k^{th} error sensor signal
$\hat{d}_k(n)$	estimate of $d_k(n)$, computed in the modified filtered-x structure of Fig. 1
$\hat{e}_k(n)$	error computed with $\hat{d}_k(n)$
$w_{i,j,l}(n)$	value at time n of the l^{th} coefficient in the adaptive FIR filter linking $x_i(n)$ and $y_j(n)$
$h_{j,k,m}$	value of the m^{th} coefficient in the FIR filter modeling the plant between $y_j(n)$ and $e_k(n)$.

$v_{i,j,k}(n)$ 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 $\boldsymbol{h}_{j,k}$ filter (see (12)).

$$\boldsymbol{w}_{i,j}(n) = \left[w_{i,j,1}(n),\ w_{i,j,2}(n),\ \cdots\ w_{i,j,L}(n)\right]^T \quad (1)$$

$$\boldsymbol{h}_{j,k} = \left[h_{j,k,1}, h_{j,k,2}, \cdots h_{j,k,M}\right]^T \quad (2)$$

$$\boldsymbol{v}_{i,j,k}(n) = \left[v_{i,j,k}(n),\ v_{i,j,k}(n-1),\ \cdots\ v_{i,j,k}(n-L+1)\right]^T \quad (3)$$

$$\boldsymbol{x}_i(n) = \left[x_i(n),\ x_i(n-1),\ \cdots\ x_i(n-L+1)\right]^T \quad (4)$$

$$\boldsymbol{x}'_i(n) = \left[x_i(n),\ x_i(n-1),\ \cdots\ x_i(n-M+1)\right]^T \quad (5)$$

$$\boldsymbol{y}_j(n) = \left[y_j(n),\ y_j(n-1),\ \cdots\ y_j(n-M+1)\right]^T \quad (6)$$

$$\boldsymbol{V}(n) = \begin{bmatrix} \begin{bmatrix} v_{1,1,1}(n) & \cdots & v_{1,1,K}(n) \\ \vdots & \ddots & \vdots \\ v_{I,J,1}(n) & \cdots & v_{I,J,K}(n) \end{bmatrix} \\ \vdots \\ \begin{bmatrix} v_{1,1,1}(n-L+1) & \cdots & v_{1,1,K}(n-L+1) \\ \vdots & \ddots & \vdots \\ v_{I,J,1}(n-L+1) & \cdots & v_{I,J,K}(n-L+1) \end{bmatrix} \end{bmatrix} \quad (7)$$

$$\boldsymbol{w}(n) = \left[\left[w_{1,1,1}(n)\ \cdots\ w_{I,J,1}(n)\right]\ \cdots\ \left[w_{1,1,L}(n)\ \cdots\ w_{I,J,L}(n)\right]\right]^T \quad (8)$$

$$\boldsymbol{e}(n) = \left[e_1(n), e_2(n), \cdots e_K(n)\right] \quad (9)$$

$$\hat{\boldsymbol{e}}(n) = \left[\hat{e}_1(n), \hat{e}_2(n), \cdots \hat{e}_K(n)\right] \quad (10).$$

With the interlaced notation of (7) and (8), the samples in $\boldsymbol{V}(n)$ are the samples of $\boldsymbol{V}(n-1)$ that have been delayed by one sample (i.e. shifted down by one block size in (7)), except for the first (upper) block of samples in $\boldsymbol{V}(n)$ which are new samples, and the last block of samples in $\boldsymbol{V}(n-1)$ which are discarded. This "block delay line" structure or "block transversal" structure is a requirement if low-computational (i.e. “fast”) versions of RLS algorithms such as the FTF algorithm in [1] and the QRD-LSL algorithm of Section 3 are to be developed for multichannel ANC systems. Using the above notation and the equations of the original inverse QR-RLS algorithm [2], the inverse QR-RLS algorithm for multichannel ANC with the modified filtered-x structure (Fig. 1) is described by (11)-(15):

$$y_j(n) = \sum_{i=1}^{I} \boldsymbol{w}_{i,j}^T(n)\boldsymbol{x}_i(n) \quad (11)$$

$$v_{i,j,k}(n) = \boldsymbol{h}_{j,k}^T \boldsymbol{x}'_i(n) \quad (12)$$

$$\hat{d}_k(n) = e_k(n) - \sum_{j=1}^{J} \boldsymbol{h}_{j,k}^T \boldsymbol{y}_j(n) \quad (13)$$

$$e'_k(n) = d'_k(n) + \sum_{i=1}^{I}\sum_{j=1}^{J} \boldsymbol{w}_{i,j}^T(n)\boldsymbol{v}_{i,j,k}(n) \quad (14)$$

$$\begin{bmatrix} \boldsymbol{I} & \lambda^{-1/2}\boldsymbol{V}^T(n)\boldsymbol{P}^{1/2}(n-1) \\ \boldsymbol{0} & \lambda^{-1/2}\boldsymbol{P}^{1/2}(n-1) \end{bmatrix} \boldsymbol{\Theta}(n) = \begin{bmatrix} \boldsymbol{\Gamma}^{-1/2}(n) & \boldsymbol{0}^T \\ \boldsymbol{K}(n)\boldsymbol{\Gamma}^{-1/2}(n) & \boldsymbol{P}^{1/2}(n) \end{bmatrix} \quad (15)$$

$$\boldsymbol{K}(n) = \left(\boldsymbol{K}(n)\boldsymbol{\Gamma}^{-1/2}(n)\right)\left(\boldsymbol{\Gamma}^{-1/2}(n)\right)^{-1} \quad (16)$$

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) - \boldsymbol{K}(n)\hat{\boldsymbol{e}}^T(n) \quad (17)$$

with the following initial conditions:

$$\boldsymbol{w}(0) = \boldsymbol{0} \quad (18)$$

$$\boldsymbol{P}^{1/2}(0) = \delta^{-1/2}\boldsymbol{I} \quad (19).$$

Note that because of the modified filtered-x structure no μ factor or adaptive gain factor is required in (17). In (15), λ is the “forgetting factor” scalar common to all least-squares algorithms (typically $0.9 \le \lambda \le 1$), $\boldsymbol{\Gamma}(n)$ is the multichannel $K \times K$ “innovation factor” or “conversion factor” sometimes used in least-squares algorithms to convert *a priori* errors to *a posteriori* errors, $\boldsymbol{P}(n)$ is the $IJL \times IJL$ inverse of the weighted time-averaged correlation matrix i.e. $\boldsymbol{P}(n) = \left(\sum_{i=0}^{n} \lambda^{n-i}\boldsymbol{V}(i)\boldsymbol{V}^T(i)\right)^{-1}$, δ is a small positive constant that represents the initial value of the diagonal components of this time-averaged correlation matrix, $\boldsymbol{K}(n)$ is a $IJL \times K$ gain matrix with the same dimensions as $\boldsymbol{V}(n)$, $\boldsymbol{I}$ is a $K \times K$ identity matrix, and $\boldsymbol{0}$ is a $IJL \times K$ zero matrix. $\boldsymbol{\Theta}(n)$ represents the set of unitary transformations that are required to transform the left side matrix in (15) to the matrix on the right side of (15). To perform the required matrix rotations, simple Givens rotations can be used. Note that no iterative process is required for the complete transformation: only one rotation is required for each element to be zeroed [4]. The computational load of the inverse QR-RLS algorithm for multichannel ANC is evaluated in [4].

3. QRD-LSL ALGORITHM FOR MULTICHANNEL ANC SYSTEMS

The inverse QR-RLS for multichannel ANC systems introduced in the previous section has a computational load proportional to $(IJL)^2$, like the RLS algorithm for multichannel ANC systems previously published [1]. For adaptive FIR filters with a lot of coefficients (high values of L), this computational load can become too high for real-time implementations. This is the motivation for developing “fast” RLS algorithms for multichannel ANC systems, where the computational load will increase with L, and not with L^2. A low computational realization is the fast-transversal-filter (FTF) algorithm, and a FTF algorithm for multichannel ANC was introduced in [1]. However it suffers from numerical instability, even with the use of a “rescue variable” (whose value is not easy to adjust for multichannel ANC systems). The QR-decomposition least-squares-lattice (QRD-LSL) algorithm [2] is known to be a very numerically robust algorithm, and it is an algorithm that can be extended to multichannel ANC systems (as opposed to some other “fast” RLS algorithms [4]). The QRD-LSL algorithm can be developed for multichannel ANC systems from the classical description [2] and using the following additional definitions:

$\boldsymbol{E}_{b,m}(n)$ $IJ \times K$ angle-normalized backward prediction error of a multichannel backward predictor [2] of order m, at time n.

$\boldsymbol{E}_{f,m}(n)$ $IJ \times K$ angle-normalized forward prediction error of a multichannel forward predictor of order m, at time n.

$\varepsilon_m(n)$ $1 \times K$ angle-normalized error of a multichannel joint-process predictor [2] of order m, at time n

$\boldsymbol{B}_{m-1}^{1/2}(n)$ $IJ \times IJ$ square root of the weighted sum of angle-normalized backward prediction errors squares $\left(\sum_{i=0}^{n} \lambda^{n-i} \boldsymbol{E}_{b,m-1}(i) \boldsymbol{E}_{b,m-1}^{T}(i)\right)$

$\boldsymbol{F}_{m-1}^{1/2}(n)$ $IJ \times IJ$ square root of the weighted sum of angle-normalized forward prediction errors squares $\left(\sum_{i=0}^{n} \lambda^{n-i} \boldsymbol{E}_{f,m-1}(i) \boldsymbol{E}_{f,m-1}^{T}(i)\right)$

$\boldsymbol{P}_{b,m-1}(n)$ $IJ \times IJ$ auxiliary coefficients [2] of the multichannel backward predictor of order $m-1$, at time n

$\boldsymbol{P}_{f,m-1}(n)$ $IJ \times IJ$ auxiliary coefficients of the multichannel forward predictor of order $m-1$, at time n

$\boldsymbol{p}_{m-1}(n)$ $IJ \times 1$ auxiliary coefficients of the multichannel joint-process predictor of order $m-1$, at time n

$\boldsymbol{\Gamma}_{m}^{1/2}(n-1)$ $K \times K$ square root of the innovation factor from the angle-normalized error $\varepsilon_m(n-1)$

$\boldsymbol{\Theta}_{b,m\text{-}1}(n)$ matrix that represents the combination of all the unitary rotations required to compute (22)

$\boldsymbol{\Theta}_{f,m\text{-}1}(n)$ matrix that represents the combination of all the unitary rotations required to compute (21).

The QRD-LSL algorithm for multichannel ANC systems with the modified filtered-x structure is then described by (11)-(13) and (20)-(35):

For $m = 1,2,\ldots,L-1$

$$\begin{bmatrix} \lambda^{1/2}\boldsymbol{B}_{m-1}^{1/2}(n-2) & \boldsymbol{E}_{b,m-1}(n-1) \\ \lambda^{1/2}\boldsymbol{P}_{f,m-1}^{T}(n-1) & \boldsymbol{E}_{f,m-1}(n) \\ \boldsymbol{0} & \boldsymbol{\Gamma}_{m-1}^{1/2}(n-1) \end{bmatrix} \boldsymbol{\Theta}_{b,m-1}(n-1) = \begin{bmatrix} DON'T\ CARE\ 1 & \boldsymbol{0} \\ \boldsymbol{P}_{f,m-1}^{T}(n) & \boldsymbol{E}_{f,m}(n) \\ DON'T\ CARE\ 2 & \boldsymbol{\Gamma}_{m}^{1/2}(n-1) \end{bmatrix} \quad (20)$$

$$\begin{bmatrix} \lambda^{1/2}\boldsymbol{F}_{m-1}^{1/2}(n-1) & \boldsymbol{E}_{f,m-1}(n) \\ \lambda^{1/2}\boldsymbol{P}_{b,m-1}^{T}(n-1) & \boldsymbol{E}_{b,m-1}(n-1) \end{bmatrix} \boldsymbol{\Theta}_{f,m-1}(n) = \begin{bmatrix} \boldsymbol{F}_{m-1}^{1/2}(n) & \boldsymbol{0} \\ \boldsymbol{P}_{b,m-1}^{T}(n) & \boldsymbol{E}_{b,m}(n) \end{bmatrix} \quad (21)$$

For $m = 1,2,\ldots,L$

$$\begin{bmatrix} \lambda^{1/2}\boldsymbol{B}_{m-1}^{1/2}(n-1) & \boldsymbol{E}_{b,m-1}(n) \\ \lambda^{1/2}\boldsymbol{p}_{m-1}^{T}(n-1) & \varepsilon_{m-1}(n) \end{bmatrix} \boldsymbol{\Theta}_{b,m-1}(n) = \begin{bmatrix} \boldsymbol{B}_{m-1}^{1/2}(n) & \boldsymbol{0} \\ \boldsymbol{p}_{m-1}^{T}(n) & \varepsilon_{m}(n) \end{bmatrix} \quad (22)$$

with

$$\boldsymbol{E}_{f,0}(n) = \boldsymbol{E}_{b,0}(n) = \begin{bmatrix} v_{1,1,1}(n) & \cdots & v_{1,1,K}(n) \\ \vdots & \ddots & \vdots \\ v_{I,J,1}(n) & \cdots & v_{I,J,K}(n) \end{bmatrix} \quad (23)$$

$$\varepsilon_0(n) = \left[\hat{d}_1(n), \hat{d}_2(n), \cdots \hat{d}_K(n)\right] \quad (24)$$

$$\boldsymbol{\Gamma}_{0}^{1/2}(n-1) = \boldsymbol{I} \quad (25)$$

and the following initial conditions:

for $m = 1,2,\ldots,L$

$$\boldsymbol{p}_{m\text{-}1}(0) = \boldsymbol{0} \quad (26)$$

$$\boldsymbol{F}_{m-1}^{1/2}(0) = \boldsymbol{B}_{m-1}^{1/2}(0) = \boldsymbol{B}_{m-1}^{1/2}(-1) = \delta^{1/2}\boldsymbol{I} \quad (27)$$

for $m = 1,2,\ldots,L-1$

$$\boldsymbol{P}_{f,m\text{-}1}(0) = \boldsymbol{P}_{b,m\text{-}1}(0) = \boldsymbol{0} \quad (28).$$

The unitary rotations in (20)-(22) can be computed in a similar manner to the rotation in (15), using Givens rotations [4]. Equation (22) provides the multichannel auxiliary joint-process coefficients $\boldsymbol{p}_{m\text{-}1}(n)$, but these coefficients are not the time-domain adaptive filters coefficients $w_{i,j,l}(n)$ that are required for the upper path of an ANC structure as shown in Fig. 1. A time-varying inverse transformation from the $\boldsymbol{p}_{m\text{-}1}(n)$ coefficients to the $w_{i,j,l}(n)$ coefficients is thus required. Such an inverse transformation exists for the classical QRD-LSL algorithm [2], and it can be extended to the multichannel ANC case. However, the inverse transform will have a computational load proportional to $(IJL)^2$, and the reason for developing the QRD-LSL for multichannel ANC systems was to avoid such a $(IJL)^2$ dependency. But it is possible to update the multichannel joint-process auxiliary coefficients $\boldsymbol{p}_{m\text{-}1}(n)$ on a sample by sample basis, as in any recursive least-squares algorithm, and to update the time-domain coefficients $w_{i,j,l}(n)$ at a reduced rate. As long as the period between updates is less than the time constant caused by the forgetting factor λ, this will not greatly affect the convergence performance or the tracking performance of the algorithm. The computational load of the resulting QRD-LSL algorithm for multichannel ANC is evaluated in [4].

The transformation from the coefficients $\boldsymbol{p}_{m\text{-}1}(n)$ to the $w_{i,j,l}(n)$ coefficients is computed by (29)-(35), where $\boldsymbol{K}_{f,m}(n)$ and $\boldsymbol{K}_{b,m}(n)$ have dimensions $IJ \times IJ$, and $\boldsymbol{k}_m(n)$ has dimensions $IJ \times 1$. The transposed operation in (29)-(31) is only required if $\boldsymbol{B}_{m-1}^{1/2}(n)$ and $\boldsymbol{F}_{m-1}^{1/2}(n)$ are upper triangular (they could be lower triangular depending on the sequence of Givens rotations that were used in (20)-(22) [4]). $\boldsymbol{A}_0(n)$ and $\boldsymbol{C}_0(n)$ are $IJ \times IJ$ identity matrices, and all the components of $\boldsymbol{A}_m(n)$, $\boldsymbol{C}_m(n)$ or $\boldsymbol{L}(n)$ shown in (32)-(34) also have dimensions $IJ \times IJ$.

for $m = 1,2,\ldots,L-1$:

$$\boldsymbol{K}_{f,m}(n) = -\left(\boldsymbol{B}_{m-1}^{1/2\,T}(n-1)\right)^{-1} \boldsymbol{P}_{f,m-1}(n) \quad (29)$$

$$\boldsymbol{K}_{b,m}(n) = -\left(\boldsymbol{F}_{m-1}^{1/2\;T}(n)\right)^{-1} \boldsymbol{P}_{b,m-1}(n) \quad (30)$$

for $m = 1,2,\ldots,L$:

$$\boldsymbol{k}_{m-1}(n) = \left(\boldsymbol{B}_{m-1}^{1/2\;T}(n)\right)^{-1} \boldsymbol{p}_{m-1}(n) \quad (31)$$

for $m = 1,2,\ldots,L-1$:

$$\underbrace{\begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{A}_{m,1}(n) \\ \vdots \\ \boldsymbol{A}_{m,m}(n) \end{bmatrix}}_{\boldsymbol{A}_m(n)} = \underbrace{\begin{bmatrix} \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{A}_{m-1,1}(n) \\ \vdots \\ \boldsymbol{A}_{m-1,m-1}(n) \end{bmatrix} \\ \boldsymbol{0} \end{bmatrix}}_{\begin{bmatrix} \boldsymbol{A}_{m-1}(n) \\ \boldsymbol{0} \end{bmatrix}} + \underbrace{\begin{bmatrix} \boldsymbol{0} \\ \begin{bmatrix} \boldsymbol{C}_{m-1,m-1}(n-1) \\ \vdots \\ \boldsymbol{C}_{m-1,1}(n-1) \\ \boldsymbol{I} \end{bmatrix} \end{bmatrix}}_{\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{C}_{m-1}(n-1) \end{bmatrix}} \boldsymbol{K}_{f,m}(n) \quad (32)$$

$$\underbrace{\begin{bmatrix} \boldsymbol{C}_{m,m}(n) \\ \vdots \\ \boldsymbol{C}_{m,1}(n) \\ \boldsymbol{I} \end{bmatrix}}_{\boldsymbol{C}_m(n)} = + \underbrace{\begin{bmatrix} \boldsymbol{0} \\ \begin{bmatrix} \boldsymbol{C}_{m-1,m-1}(n) \\ \vdots \\ \boldsymbol{C}_{m-1,1}(n) \\ \boldsymbol{I} \end{bmatrix} \end{bmatrix}}_{\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{C}_{m-1}(n) \end{bmatrix}} + \underbrace{\begin{bmatrix} \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{A}_{m-1,1}(n) \\ \vdots \\ \boldsymbol{A}_{m-1,m-1}(n) \end{bmatrix} \\ \boldsymbol{0} \end{bmatrix}}_{\begin{bmatrix} \boldsymbol{A}_{m-1}(n) \\ \boldsymbol{0} \end{bmatrix}} \boldsymbol{K}_{b,m}(n) \quad (33)$$

$$\boldsymbol{L}(n) = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{C}_{1,1}^T(n) & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{C}_{2,2}^T(n) & \boldsymbol{C}_{2,1}^T(n) & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{C}_{L-1,L-1}^T(n) & \boldsymbol{C}_{L-1,L-2}^T(n) & \cdots & \boldsymbol{C}_{L-1,1}^T(n) & \boldsymbol{I} \end{bmatrix} \quad (34).$$

$$\boldsymbol{w}(n) = -\boldsymbol{L}^T(n) \begin{bmatrix} \boldsymbol{k}_0(n) \\ \boldsymbol{k}_1(n) \\ \vdots \\ \boldsymbol{k}_{L-1}(n) \end{bmatrix} \quad (35).$$

4. SIMULATION RESULTS

Simulations of a broadband multichannel ANC system with $I = 1$, $J = 2$ and $K = 2$ were performed, using the same experimentally measured acoustic transfer functions as in [1]. The simulations were performed using a C program implementation, in single precision floating point representation (32 bits). The code from [5] was used for matrix inversions with LU decompositions. Adaptive filters with $L = 100$ coefficients were used. The convergence gain of least-squares algorithms over steepest descent algorithms for multichannel ANC systems has already been documented [1], so results about numerical stability of the recursive-least-squares algorithms will mostly be discussed here. Table 1 compares the numerical stability of the different recursive least-squares algorithms. If the algorithm is stable, then the steady state performance of the algorithm is shown, and if the algorithm is unstable, the approximate number of iterations before the algorithm diverges is shown. From Table 1, it is clear that the RLS and FTF algorithms developed for multichannel ANC systems are numerically unstable, as previously reported in [1]. The inverse QR-RLS and QRD-LSL algorithms were stable over millions of iterations.

Multichannel ANC algorithm	$\lambda = 1.0$, $\delta = 1.0$	$\lambda = 0.999$, $\delta = 1.0$
RLS	unstable, 500 iterations	unstable, 200 iterations
FTF (no rescue variable)	unstable, 50000 iterations	unstable, 27000 iterations
inverse QR-RLS	stable, 25 dB average attenuation	stable, 23 dB average attenuation
QRD-LSL	stable, 25 dB average attenuation	stable, 22 dB average attenuation

Table 1. Simulation results of multichannel ANC.

Although the QRD-LSL requires less computations than the inverse QR-RLS algorithm (if the time domain coefficients are not updated on a sample by sample basis), the inverse QR-RLS is simpler to implement and preliminary simulation results with low precision numerical representations (12 or 16 bits in fixed point) or with ill-conditioned systems (with more actuators than error sensors for example) have shown that it is more stable than the QRD-LSL. Therefore if one can afford the computational complexity of the inverse QR-RLS algorithm, then it is a good recursive-least-squares algorithm to use for multichannel ANC.

5. CONCLUSION

This paper addressed the development of recursive least-squares algorithm with good numerical stability for multichannel ANC systems. Two numerically stable multichannel ANC recursive-least-squares algorithms were introduced: the inverse QR-RLS and QRD-LSL algorithms. The QRD-LSL algorithm requires less computations than the inverse QR-RLS algorithm, if the time domain coefficients are not computed on a sample by sample basis. Simulation results have shown that the two algorithms are stable in a 32 bits floating point number representation environment.

6. REFERENCES

[1] M. Bouchard and S. Quednau, "Multichannel recursive least-squares algorithms and fast-transversal-filter algorithms for active noise control and sound reproduction systems," *IEEE Trans. Speech Audio Processing*, vol. 8, n.5, Sept. 2000

[2] S. Haykin, *Adaptive filter theory*, 3rd edition, Englewood Cliffs (NJ), Prentice Hall, 989 p., 1996

[3] I.S. Kim, H.S. Na, K.J. Kim and Y. Park, “Constraint filtered-x and filtered-u least-mean-square algorithms for the active control of noise in ducts,” *J. Acoust. Soc. Am.*, vol. 95, pp. 3379-3389, June 1994

[4] M. Bouchard, “Recursive least-squares algorithms with good numerical stability and constrained least-squares algorithms for multichannel active noise control or transaural sound reproduction systems” submitted for publication in *IEEE Trans. Speech Audio Processing*

[5] W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, *Numerical Recipes in C : The Art of Scientific Computing*, 2nd edition, Cambridge Univ. Press, 994 p., 1993