Hybrid FxRLS-FxNLMS Adaptive Algorithm for Active Noise Control in fMRI Application

Rajiv M. Reddy, *Member, IEEE*, Issa M. S. Panahi, *Senior Member, IEEE*, and Richard Briggs, *Member, IEEE*

***Abstract*—A hybrid adaptive algorithm is developed for an active noise control system that leverages the stability of the filtered-input normalized least mean squares (FxNLMS) adaptive algorithm, with the high convergence speed of the filtered-input recursive least squares (FxRLS) adaptive algorithm. This algorithm is motivated by practical issues in implementing a real-time active noise control system. It leads to fast initial convergence with low, stable steady-state error while being limited by the computational capability of hardware. It gives better convergence speed than either the FxNLMS or FxRLS algorithm individually, lower residual error, and a lower overall computational complexity than the FxRLS algorithm, when appropriate filter lengths are chosen. Experimental results are presented for the implementation of the hybrid algorithm to cancel functional magnetic resonance imaging (fMRI) acoustic noise in an fMRI test-bed.**

***Index Terms*—Acoustic noise control, active noise control (ANC), adaptive filtering, adaptive signal processing, biomedical application, functional magnetic resonance imaging (fMRI).**

I. Introduction

ACTIVE noise control (ANC) has been of interest to researchers for several years now [1]. With the availability of low-cost high-speed processors, there has been a renewed interest in ANC for a variety of applications including aircrafts [2], automobiles [3], industrial machine noise [4], as well as ANC headphones [5].

This brief looks at the application of active noise control to cancel functional magnetic resonance imaging (fMRI) acoustic noise. High acoustic noise levels from fMRI machines, besides being hazardous to the patient, interferes with medical research like the functional mapping of brain activity [6], [7] which is the motivation for this research. To evaluate the performance of active noise control for fMRI acoustic noise application, fMRI acoustic noise is collected from The University of Texas Southwestern Medical Center and an fMRI test-bed is designed to simulate the fMRI bore. Experimental results from implementing the normalized least mean squares (NLMS) and affine projection algorithm (APA) using single-tone noise signals are presented in [8] for this fMRI test-bed. Experimental results from implementing the filtered-input normalized least mean squares (FxNLMS), the filtered-input affine projection algorithm (FxAPA), and the filtered-input recursive least squares (FxRLS) adaptive algorithms for single-tone noise as well as bandlimited fMRI noise are presented in [9] using the same fMRI test-bed. These experimental results [8], [9] match theoretical results [10] showing that the FxRLS algorithm has a very high convergence rate which is desirable in this application. However, due to the computational limits of commonly used inexpensive hardware, only a short filter length can be implemented using the FxRLS algorithm on the hardware which is insufficient to model the entire system. To implement a practical system, we are then left with a tradeoff between choosing an algorithm having a high convergence speed like the FxRLS algorithm with a suboptimal filter length, and hence, a larger residual error, or using the FxNLMS algorithm with slower convergence speed but an optimal filter length, and hence, a lower residual error when it converges.

This tradeoff motivated the development of the hybrid algorithm presented in this brief. The contributions of this brief focus toward developing an algorithm with faster convergence rate for active noise control application while being limited by the computational speed of real hardware which is necessary for the fMRI application and most practical applications. This brief ignores the fact that fMRI test-bed designed is only an approximation to the real system and modifications to the test-bed may be necessary to fully account for all the practical issues in a fMRI room. The hybrid algorithm allows the user to run the computationally expensive and sometimes unstable FxRLS algorithm with a filter size that is limited only by the performance of the hardware. Once quick convergence has been obtained, based on our detection scheme, the algorithm switches to the lower computation FxNLMS algorithm, but, with a larger filter size. This algorithm has immense application in industry for a variety of applications where engineers are often faced with a tradeoff between convergence rate and computational complexity on available cost-effective hardware platforms.

In recent years, there have been other researchers who have also considered combining the properties of the NLMS and RLS algorithm, but by using different methods from our hybrid FxRLS-FxNLMS algorithm. Cascaded RLS-LMS algorithms are used for lossless audio compression where the RLS algorithm is used as a preprocessor to the LMS stage [11]–[13]. A similar cascaded approach is used in [14], [15] for an array beam forming application. However, cascaded structures are not feasible for ANC since desired signal cannot directly be

Manuscript received August 15, 2008; revised June 21, 2009. Manuscript received in final form February 01, 2010. First published March 11, 2010; current version published February 23, 2011. Recommended by Associate Editor C. Bohn. This work was supported by a subcontract from UT Southwestern Medical Center at Dallas, funded by the Department of Veterans Affairs through VA IDIQ Contract VA549-P-0027 awarded and administered by the Dallas, TX VA Medical Center. The content of this brief does not necessarily reflect the position or the policy of the Veterans Administration or the Federal government, and no official endorsement should be inferred.

R. M. Reddy was with The University of Texas, Dallas, TX 75080 USA. He is now with the R&D Department, LumenVox, San Diego, CA 92123 USA (e-mail: rajivmreddy@gmail.com).

I. M. S. Panahi is with the Department of Electrical Engineering, The University of Texas, Dallas, TX 75080 USA (e-mail: issa.panahi@utdallas.edu).

R. Briggs is with the Department of Radiology, The University of Texas, Southwestern Medical Center, Dallas, TX 75390 USA (e-mail: richard.briggs@utsouthwestern.edu).

Digital Object Identifier 10.1109/TCST.2010.2042599

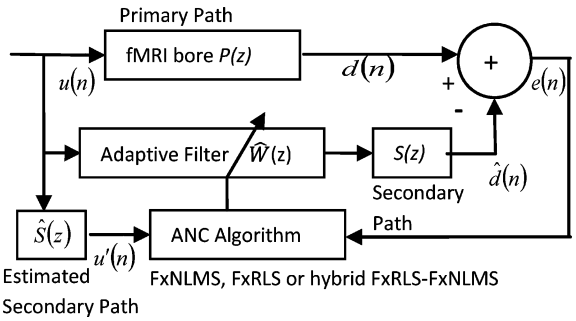

Fig. 1. Filtered input ANC algorithm setup.

accessed in ANC, and hence, intermediate error signals cannot be introduced between the cascaded stages for this control application. A parallel RLS-LMS adaptation for estimating fast fading channels is presented in [16]. However, the improved convergence speed is obtained with a higher computational cost which does not address the computational limitations faced in real-time implementations on cost-effective digital signal processing (DSP) platforms.

The advantages of the proposed hybrid algorithm in real-time implementation are as follows.

1) Faster convergence than FxNLMS by itself.
2) Faster tracking capability to changes in the input signals, primary paths or secondary paths.
3) Lower overall computational complexity than FxRLS, when appropriate filter lengths are chosen.
4) Better than or equal Sound Pressure Level (SPL) cancellation as compared to FxNLMS or FxRLS by themselves for the same number of iterations.
5) Ensures stability of the FxRLS algorithm which has the tendency to become unstable and diverge, even after showing some convergence.

II. Active Noise Control

For ANC application, we need to use filtered-input adaptive algorithms, i.e., the input to the adaptive algorithm is filtered by the secondary path estimate $\hat{S}(z)$ to take care of the effects of the secondary path $S(z)$ on the adaptive filter $\widehat{W}(z)$ [1]. Fig. 1 shows the block diagram of a filtered-input ANC algorithm. $P(z)$ represents the system transfer function for the noise signal propagation from the source to the canceling point in the acoustic cavity. $S(z)$ represents the system transfer function for the sensor (microphone), loudspeaker and electrical and acoustical transmission. We solve for $\widehat{W}(z)$, such that

$$P(z) - S(z)\cdot\widehat{W}(z) \to 0 \tag{1}$$

$$\Rightarrow \widehat{W}(z) \to \frac{P(z)}{S(z)} \tag{2}$$

where $P(z)$ is called the primary path and $S(z)$ is called the secondary path. The estimate of the desired signal $\hat{d}(n)$ to be cancelled is obtained by filtering the measured reference signal, $\mathbf{u}(n)$, by an adaptive linear causal FIR filter whose coefficients are represented by the column vector $\hat{\mathbf{w}}(n)$. Thus

$$\hat{d}(n) = s(n) * \left([\widehat{\mathbf{w}}(n)]^{\boldsymbol{T}}_{(\mathbf{L+M})\times\mathbf{1}}[\mathbf{u}(n)]_{(\mathbf{L+M})\times\mathbf{1}}\right) \tag{3}$$

where $[\,]^{\boldsymbol{T}}$ is the vector transpose, $s(n)$ is the secondary path impulse response, $L+M$ is the total filter length, where

$$[\widehat{\mathbf{w}}(n)]_{(\mathbf{L+M})\times\mathbf{1}} = [w(0)\ldots w(L)\ldots w(L+M)]^{\boldsymbol{T}}. \tag{4}$$

III. Hybrid FxRLS-FxNLMS Algorithm

To estimate $[\widehat{\mathbf{w}}(n)]_{(\mathbf{L+M})\times\mathbf{1}}$, we use the proposed hybrid FxRLS-FxNLMS algorithm which involves switching between 2 algorithms, namely FxRLS and FxNLMS.

A. Detection

A detection scheme is necessary to decide when to switch algorithms. We use different filter lengths for the FxNLMS and FxRLS algorithms with the FxNLMS filter length being closer to the optimal filter length. Hence, we assume that the FxRLS algorithm will leave a large residual error at the canceling point but converge faster, while the FxNLMS algorithm will converge slowly but leave a lower residual error. As a result, we use the FxRLS algorithm first until the algorithm converges to the lowest possible residual error. Once it stops converging, we switch over to the FxNLMS algorithm with an optimal filter length, thus, continuing the convergence to the optimal filter coefficients at a slower rate. Hence, the decision to choose the algorithm is based upon the rate of change of the sound pressure level (SPL) measured at the canceling point expressed as

$$\left|\frac{d}{dn}\mathrm{SPL}(n)\right| \underset{\mathrm{SPL}}{\overset{\mathrm{RLS}}{\gtrless}} \tau_0 \tag{5}$$

where τ_0 is the SPL rate of change detection threshold. By setting the threshold on the absolute value of the SPL rate of change, it switches the algorithm back to the FxRLS case for quick convergence, when the primary or secondary path changes suddenly. We find τ_0 empirically. The $\mathrm{SPL}(n)$ is computed as follows:

$$\mathrm{SPL}(n) = 20\log\left(\frac{p_{\mathrm{rms}}}{p_{\mathrm{ref}} = 2E-5\ \mathrm{pa}}\right) \tag{6}$$

where the reference pressure p_{ref} equals 2×10^{-5} Pa (Pascal) for sound in air and root-mean-square pressure, p_{rms}, at the microphone is calculated by (7)

$$p_{\mathrm{rms}} = \sqrt{\frac{1}{N}\left(\sum_{n=0}^{N}\left(\frac{\mathrm{e(n)}}{\mathrm{Mic_{gen}} = 47\ \frac{\mathrm{mV}}{\mathrm{pa}}}\right)^2\right)} \tag{7}$$

where the microphone sensitivity, $\mathrm{Mic_{gen}}$, for the microphones we used is 47 mV/Pa. The length N is set such that the error power is based on 500 ms of data which is a standard for the *slow* setting in most commercial SPL meters. This error power

$\sum_{n=0}^{N}(\mathrm{e(n)})^2$ may also be approximated with an appropriate low pass estimator to reduce the complexity of the detector.

B. Filtered-Input Recursive Least Squares (FxRLS) Algorithm

As mentioned earlier, we use different filter lengths in the FxNLMS and FxRLS algorithms due to the different computational complexity of the FxNLMS and FxRLS algorithm. For the FxRLS algorithm, only the first L coefficients are updated

$$[\widehat{\mathbf{w}}(n)]_{(\mathbf{L+M})\times\mathbf{1}} = \left[([\widehat{\mathbf{w}}(n)]_{\mathbf{L}\times\mathbf{1}})^{\boldsymbol{T}} \underbrace{0 \quad \dots \quad 0}_{M}\right]^{\boldsymbol{T}} \tag{8}$$

$$[\widehat{\mathbf{w}}(n)]_{\mathbf{L}\times\mathbf{1}} = [\widehat{w}(0)\widehat{w}(1)\dots\widehat{w}(L)]^{\boldsymbol{T}}. \tag{9}$$

The rest of the equations for the FxRLS algorithm are shown in (10)–(14)

$$\mathbf{P}(0) = \delta^{-1}[\mathbf{I}]_{\mathbf{L}\times\mathbf{L}} \tag{10}$$

where $\mathbf{P}(0)$ initializes the $L \times L$ signal correlation matrix inverse $\mathbf{P}(n)$, $\mathbf{I}$ is the identity matrix, and δ is a small constant for high SNR or a large constant for low SNR

$$\boldsymbol{\pi}(n) = \mathbf{P}(n-1)\mathbf{u}'(n) \tag{11}$$

where the column vector $\mathbf{u}'(n)$ equals the reference signal $\mathbf{u}(n)$, filtered by the secondary path estimate $\hat{S}(z)$. $\mathbf{u}(n)$ represents L present and past samples of the input noise signal at time n

$$\mathbf{k}(n) = \frac{\boldsymbol{\pi}(n)}{\lambda + \mathbf{u}'^{\boldsymbol{T}}(n)\boldsymbol{\pi}(n)} \tag{12}$$

$$[\widehat{\mathbf{w}}(n)]_{\mathbf{L}\times\mathbf{1}} = [\widehat{\mathbf{w}}(n-1)]_{\mathbf{L}\times\mathbf{1}} + \mathbf{k}(n)e^*(n) \tag{13}$$

$$\mathbf{P}(n) = \lambda^{-1} + \mathbf{P}(n-1) - \lambda^{-1}\mathbf{k}(n)\mathbf{u}'^{\boldsymbol{T}}(n)\mathbf{P}(n-1) \tag{14}$$

where $\boldsymbol{\pi}(n)$ and $\mathbf{k}(n)$ are dummy variables; and λ is a forgetting factor [10].

C. Filtered-Input Normalized Least Mean Squares (FxNLMS) Algorithm

When we switch to the FxNLMS algorithm from the FxRLS algorithm, the adaptive filter is zero padded to $L + M$ taps. Hence, $[\widehat{\mathbf{w}}(n)]_{(\mathbf{L+M})\times\mathbf{1}}$ from (8) is used in the FxNLMS adaptive filter

$$[\widehat{\mathbf{w}}(n+1)]_{(\mathbf{L+M})\times\mathbf{1}} = [\widehat{\mathbf{w}}(n)]_{(\mathbf{L+M})\times\mathbf{1}} + \frac{\mu\mathbf{u}'(n)e^*(n)}{||u'(n)||^2 + \boldsymbol{\varepsilon}} \tag{15}$$

where μ is the convergence step-size and ε is a small positive value that prevents a zero division.

If there is a sudden change in the primary or secondary path the algorithm switches back to the FxRLS algorithm.

IV. Computational Complexity

We now compute a criterion to find the maximum length of the optimal filter, $L + M$, given that we know the length of the suboptimal filter, L, used for the FxRLS algorithm. For each algorithm we have

$$\text{Complexity of FxNLMS} = \begin{cases} 3(L+M)+1 \text{ additions} \\ 3(L+M)+1 \text{ multiplications} \\ 1 \text{ additions} \end{cases} \tag{16}$$

$$\text{Complexity of FxRLS} = \begin{cases} L^2+5L+1 \text{ additions} \\ L^2+3L \text{ multiplications} \\ 1 \text{ additions} \end{cases}. \tag{17}$$

The maximum complexity of the FxNLMS algorithm has to be equal to or less than that of the FxRLS algorithm. Thus

$$\Rightarrow \begin{bmatrix} 3(L+M)+1 \text{ additions} \\ 3(L+M)+1 \text{ multiplies} \\ 1 \text{ additions} \end{bmatrix} \leq \begin{bmatrix} L^2+5L+1 \text{ additions} \\ L^2+3L \text{ multiplies} \\ 1 \text{ additions} \end{bmatrix}.$$

Assuming that there are hardware multipliers and adders on the DSP, it implies a single cycle for each computation. Thus, we must have

$$\Rightarrow M < \frac{(2L^2+2L-1)}{6}. \tag{18}$$

As long as (18) is met, the overall computational complexity of applying this hybrid algorithm will be less than that of applying the FxRLS algorithm. Due to the quadratic nature of (18), this condition is easily met. The additional freed up computations may be used to run other low priority applications on the DSP.

V. Experimental Setup

In this section, we describe the experimental setup used for the MATLAB simulations and real-time experiments to compare FxNLMS, FxRLS and hybrid FxRLS-FxNLMS methods.

A. Noise Data Set

To test the performance of the hybrid algorithm, we use real fMRI noise collected from a Siemens Trio 3 Tesla whole-body human MRI system at the The University of Texas Southwestern Medical Center, as well as bandlimited white noise. The fMRI data was initially sampled at 64 kHz when collected. We use the 70 Hz-2 kHz bandwidth of this signal (by down-sampling the data to 4 kHz) to test the performance of our algorithm in active noise control. The fMRI acoustic noise and its frequency response are shown in Fig. 2(f) and (c).

B. Hardware Setup

The hardware used includes a floating-point DSP development platform (DSK) from Texas Instruments, TI 6713 DSK, a simulated fMRI bore test-bed with a manikin placed inside it, two 52si Belkin speakers, two 4942 BK microphones, and a CT4200 Crown amplifier. A NI 6733 digital-to-analog converter (DAC) and a NI 4472 analog-to-digital converter (ADC)

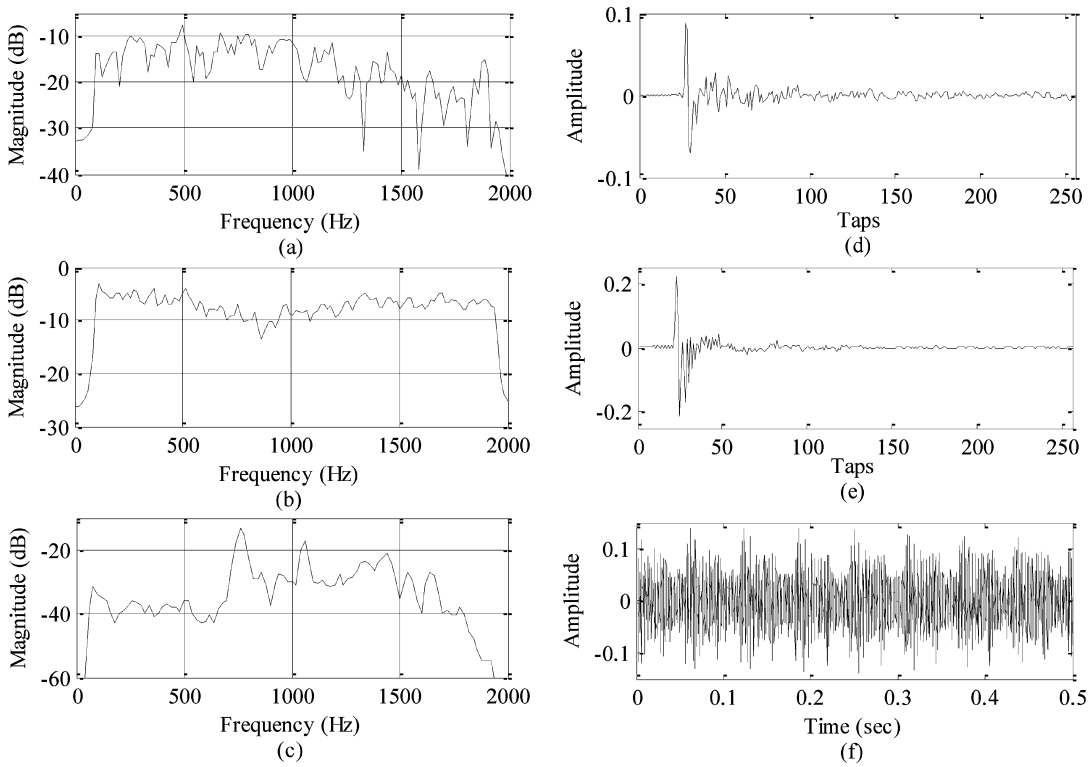

Fig. 2. (a) Primary path frequency response. (b) Secondary path frequency response. (c) fMRI noise frequency response. (d) Primary path impulse response. (e) Secondary path impulse response. (f) fMRI noise in time domain.

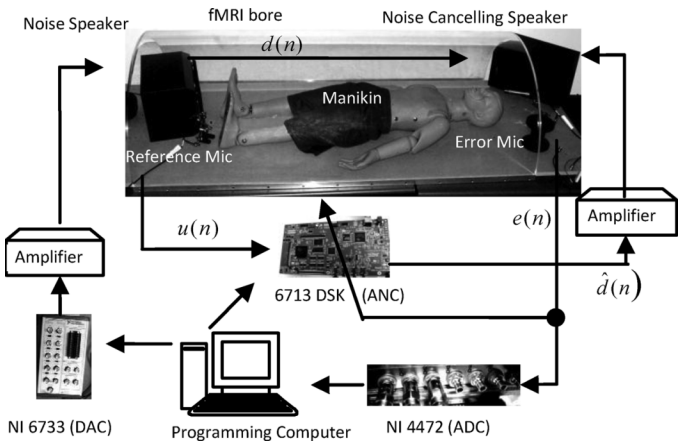

Fig. 3. Block diagram showing the hardware setup of the test-bed used in the real-time experiments.

are used to automate the noise generation system, system identification, and the data collection system. The hardware connections are as shown in Fig. 3.

The bore and the manikin together form the *primary path*, $P(z)$, while the *reference microphone*, DSK, *amplifier*, and *loudspeaker* form the *secondary path*, $S(z)$, in the ANC model shown in Fig. 1. The *reference microphone* is placed behind the *noise loud-speaker* and is covered with foam to reduce the effect of the *noise cancelling loudspeaker* feedback path on the reference signal. The dimensions of the room are 4.65 m (L) $\times$ 2.90 m (W) $\times$ 2.74 m (H).

C. Primary and Secondary Paths

The MATLAB simulations are carried out using estimates of realistic primary and secondary paths as shown in Fig. 2, modeled from our mock fMRI test-bed as described above. Estimating the models of primary and secondary paths are simple system identification problems, and hence, we will not go into the details of identifying them. The models were obtained as linear causal stable single-input single-output transfer functions $P(z)$ and $S(z)$, respectively. Such approximation models serve our objectives aimed at comparing the performance of the algorithms under the same conditions. The impulse response of the primary path and secondary paths are 256 taps at a 4 kHz sampling rate.

VI. Simulations and Real-Time Results

In this section, we compare the performance of the FxNLMS, FxRLS and hybrid FxRLS-FxNLMS algorithms in MATLAB simulation as well as in a real-time implementation.

A. MATLAB *Simulations*

The τ_0, L, $L+M$, and λ values were varied within the following sets. $\tau_0 = \{0.5, 1, 2, 4\}$ dB/s, $L = \{32, 46, 64, 128\}$, $L+M = \{64, 128, 256\}$, $\lambda = \{0.995, 0.999, 0.9995\}$. We varied μ within the range [0.001,0.5] to find the optimal value that gives the best convergence. Over 3000 MATLAB simulations were conducted with various combinations of these parameters to verify the validity and repeatability of the proposed method.

Fig. 4 shows the convergence plots in SPL for FxNLMS, FxRLS, and hybrid FxRLS-FxNLMS algorithms tested on

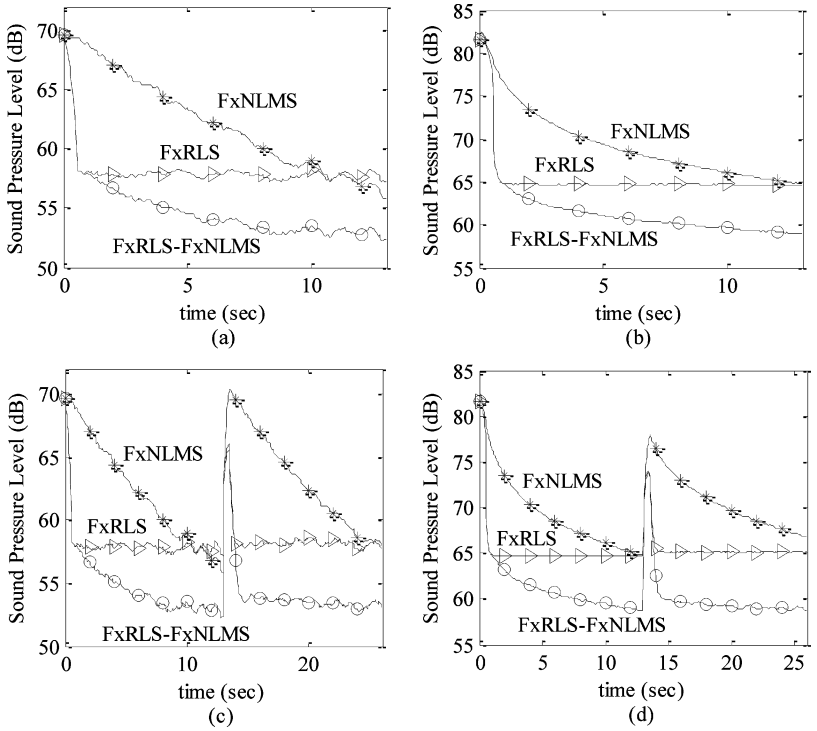

Fig. 4. SPL convergence plot for FxNLMS, FxRLS, and hybrid FxRLS-FxNLMS algorithms when $\tau_0 = 4$ dB/s, $L + M = 128$ taps, $L = 46$ taps, $\mu = 0.005$, and $\lambda = 0.999$. Data used: (a) white noise; (b) fMRI noise; (c) white noise; (d) fMRI noise. Primary path is changed in the middle of the data set for (c) and (d).

TABLE I
SIMULATION AND REAL-TIME RESULTS

Description	Data Set	FxRLS NAL	FxNLMS NAL	Hybrid FxRLS-FxNLMS NAL
Simulation/Fig. 4a	white	12.4 dB	13.8 dB	**17.3 dB**
Simulation/Fig. 4b	fMRI	16.9 dB	16.8 dB	**22.7 dB**
Simulation/Fig. 4c	white	11.6 dB	11.9 dB	**16.6 dB**
Simulation/Fig. 4d	fMRI	16.5 dB	14.8 dB	**22.8 dB**
Real-Time	white	10.1 dB	5.3 dB	**15.2 dB**
Real-Time	fMRI	15.4 dB	9.3 dB	**21.5 dB**

Data Set – white noise data or fMRI data is used in each experiment

NAL – Net Attenuation Level/ amount of noise cancellation

MATLAB where $\tau_0 = 4$ dB/s, $L + M = 128$ taps, $L = 46$ taps, $\mu = 0.005$ and $\lambda = 0.999$. The net attenuation levels (NAL), i.e., the total amount of noise cancellation in decibels, is shown in Table I for the simulations as well as the real-time experiments. The results presented here for the simulations use the parameters that match those that we found to be practically implementable in the real-time setup. We chose FxRLS filter length, L, to be 46 taps since it is the maximum filter length that we could implement using the FxRLS algorithm with the computational limits of the TI 6713 DSK. We chose the FxNLMS filter length $L + M$ to be 128 taps since it is the more optimal filter length for the primary and secondary paths used. Also, the τ_0, μ, and λ values used in the results presented here were chosen based upon the best practical values implementable on the real-time system that were the most stable and showed the best convergence.

In Fig. 4(a) and (b), we compare the convergence performance and NAL of the FxNLMS, FxRLS, and hybrid algorithms in white noise and fMRI noise to find that our hybrid algorithm performs better than the FxRLS or FxNLMS algorithm for either noise data set.

In Fig. 4(c) and (d), we compare the tracking capability of the FxNLMS, FxRLS, and hybrid FxRLS-FxNLMS algorithms in white noise and fMRI noise. We test the tracking capability by changing the primary path transfer function half way through the data set by distorting it with white noise. We find that our hybrid FxRLS-FxNLMS algorithm is able to track changes in the system at least as good as the FxRLS algorithm and much better than the FxNLMS algorithm for either noise data set. Hence, we see considerably better performance for the hybrid FxRLS-FxNLMS algorithm as compared to FxRLS and FxNLMS algorithms in all the cases shown in Fig. 4. We also examined the

effect of changing the primary path at different points during convergence by distorting the primary path with white noise to find that the hybrid algorithm showed the same tracking capability as FxRLS after a sudden change in the primary path.

1) Stability: The FxNLMS algorithm diverges when ì is too high like $\mu = 0.5$ and becomes stable when μ is decreased. The value of μ is bounded by the error in the secondary path estimate [1], [17] as shown in (19)

$$\mu < \left(\frac{2\cos\left(\angle S(w) - \angle \hat{S}(w)\right)}{c|S(w)|^2 p(w)} \right) \tag{19}$$

where c is a constant. Since we do not have access to $S(z)$ in practice, we cannot calculate (19) easily and precisely. Hence, we use $\mu = 0.005$ which we found empirically to be the highest convergence rate to produce stable convergence using the FxNLMS algorithm in the real-time experiments.

The FxRLS algorithm tends to diverge due to instability of the estimated correlation matrix inverse $P(n)$ once the optimal coefficients are found with the suboptimal filter length. We found this problem especially prominent when we used single tones. If λ is smaller, the algorithm diverges faster. It shows greater stability when λ is increased but it still eventually diverges for single tones in our case.

The hybrid FxRLS-FxNLMS algorithm solves the instability problem of the FxRLS algorithm as long as the selected threshold τ_0 is sufficiently large to switch over to the FxNLMS algorithm before the FxRLS algorithm becomes unstable. This allows us to choose a smaller λ for much faster convergence and tracking of signal changes. If τ_0 is too small, like 0.5 dB/s, the algorithm waits until the FxRLS is almost unstable before it switches over to the FxNLMS algorithm. If τ_0 is too large, it does not affect stability, but, it does effect the convergence speed, as described in the next section.

2) Convergence Speed: The FxNLMS algorithm converges much slower as compared to FxRLS but continues to converge to a point with lower residual error than the FxRLS because the filter length for the FxNLMS algorithm is more optimal. If μ is decreased the convergence speed of the FxNLMS algorithm decreases.

The FxRLS algorithm exhibits faster convergence than the FxNLMS algorithm. When λ is decreased, convergence speed and tracking capability increases, but, stability decreases.

The hybrid FxRLS-FxNLMS algorithm converges with the FxRLS algorithm as long as the SPL decreases at a rate greater than τ_0. When the convergence stagnates with the FxRLS algorithm, it switches to the FxNLMS algorithm which gradually continues to converge to the optimal solution. Hence, we see much better performance with the hybrid algorithm. If τ_0 is too small, the algorithm waits until the FxRLS algorithm has almost fully converged before switching over to the FxNLMS algorithm. Since, the rate of convergence exponentially decreases as the adaptive filter gets closer to its optimal solution, if τ_0 has a lower threshold, the detector waits too long before switching to FxNLMS, thus, decreasing the higher convergence speed advantage of the hybrid algorithm. If τ_0 is too large, the algorithm switches prematurely to the slower FxNLMS algorithm before it has taken full advantage of the convergence speed of the faster FxRLS algorithm, and hence, leading to lower overall convergence speed. Consequently, selecting an appropriate τ_0 for the hybrid algorithm is important. However, while choosing an optimal τ_0 may show the best convergence rate, choosing a less optimal τ_0 does not necessarily discount the advantages of the hybrid algorithm. We still get better performance than the FxNLMS and FxRLS algorithms individually, and improve stability of the FxRLS algorithm.

B. Real-Time Experimental Results

The FxNLMS, FxRLS, and the hybrid FxRLS-FxNLMS algorithms are implemented on the TI 6713 DSK. The same bandlimited 70 Hz–2 kHz fMRI noise and white noise are tested on this system with the sampling rate set to 4 kHz. All the parameters are similar to those used in the simulations. We are limited by the processing power of the 6713DSK and other hardware platforms, and hence, we take the FxRLS filter length, $L = 46$ taps. Since 128 taps is the more optimal length of the adaptive filter with the primary and secondary paths we used, the length of the FxNLMS algorithm $L + M$ is set to 128 taps. We set $\mu = 0.005$, $\tau_0 = 4$ dB/s, and $\lambda = 0.999$. The results are presented in Table I. These results match our simulations and we see that the hybrid algorithm performs far better than the FxNLMS or FxRLS algorithm individually.

VII. Discussion and Conclusion

We have shown an algorithm that has immense application in real-time active noise control implementations. This algorithm was conceived as a solution to the computational limitations posed by hardware in implementing the higher complexity FxRLS algorithm. This led us to consider our hybrid algorithm that implements the largest filter length we could process on the DSP using FxRLS, and after convergence, switching to the slower FxNLMS, with the more optimal filter length. Our results show that the hybrid FxRLS-FxNLMS algorithm is more stable, has much better convergence speed, and shows faster tracking capability of changes in noise signal than the FxNLMS or FxRLS algorithms individually. Our real-time implementation and experiments validate these claims.

References

[1] S. M. Kuo and D. R. Morgan, "Active noise control: A tutorial review," *Proc. IEEE*, vol. 87, no. 6, pp. 943–973, Jun. 1999.

[2] R. Cabell, D. Palumbo, and J. Vipperman, "A principal component feedforward algorithm for active noise control: Flight test results," *IEEE Trans. Control Syst. Technol.*, vol. 9, no. 1, pp. 76–83, Jan. 2001.

[3] M. H. Costin and D. R. Elzinga, "Active reduction of low-frequency tire impact noise using digital feedback control," *IEEE Control Syst. Mag.*, vol. 9, no. 5, pp. 3–6, Aug. 1989.

[4] S. M. Kuo, X. Kong, and W. S. Gan, "Applications of adaptive feedback active noise control system," *IEEE Trans. Control Syst. Technol.*, vol. 11, no. 2, pp. 216–220, Mar. 2003.

[5] S. M. Kuo, S. Mitra, and W. S. Gan, "Active noise control system for headphone applications," *IEEE Trans. Control Syst. Technol.*, vol. 14, no. 2, pp. 331–335, Mar. 2006.

[6] N. J. Shah, L. Jäncke, M.-L. Grosse-Ruyken, and H. W. Müller-Gärtner, "Influence of acoustic masking noise in fMRI of the auditory cortex during phonetic discrimination," *J. Magn. Resonance Imag.*, vol. 9, pp. 19–25, Feb. 1999.

[7] W. B. Edmister, T. M. Talavage, P. J. Ledden, and R. M. Weisskoff, "Improved auditory cortex imaging using clustered volume acquisition," *Human Brain Mapping*, vol. 7, pp. 89–97, 1999.

[8] R. M. Reddy, I. M. S. Panahi, and R. Briggs, "Analysis and realization of an active noise control system on DSP hardware using an fMRI bore model," in *Proc. 29th IEEE Annual Eng. Med. Biol. Soc. Conf.*, Lyon, Aug. 23–26, 2007, pp. 3417–3420.

[9] R. M. Reddy, I. M. S. Panahi, R. Briggs, and E. Perez, "Performance comparison of FXRLS, FXAPA and FXLMS active noise cancellation algorithms on an fMRI bore test-bed," in *Proc. Eng. Med. Biol. Workshop*, Dallas, Nov. 11–12, 2007, pp. 130–133.

[10] S. Haykin, *Adaptive Filter Theory*, 4th ed. Upper Saddle River, NJ: Prentice-Hall, 2002.

[11] R. Yu and C. C. Ko, "Lossless compression of digital audio using cascaded RLS-LMS prediction," *IEEE Trans. Speech Audio Process.*, vol. 11, no. 6, pp. 532–537, Nov. 2003.

[12] D. Y. Huang, "Performance analysis of an RLS-LMS algorithm for lossless audio compression," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Vancouver, Canada, May 17–21, 2004.

[13] H. B. Huang, S. Rahardja, X. Lin, R. Yu, and P. Franti, "Cascaded RLS-LMS predictor in MPEG-4 lossless audio coding," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Toulouse, France, May 14–21, 2006, vol. 5, pp. 181–184.

[14] J. A. Srar and K.-S. Chung, "Adaptive array beam forming using a combined RLS-LMS algorithm," in *Proc. Asia-Pacific Conf. Commun. (APCC)*, Oct. 14–16, 2008, pp. 1–5.

[15] J. A. Srar and K.-S. Chung, "Performance of RLMS algorithm in adaptive array beam forming," in *Proc. IEEE Singapore Int. Conf. Commun. Syst. (ICCS)*, Nov. 19–21, 2008, pp. 493–498.

[16] S. Oikawa, Y. Tsuda, and T. Shimamura, "A parallel estimator with LMS and RLS adaptation for fast fading channels," in *Proc. Int. Symp. Intell. Signal Process. Commun. (ISPACS)*, Dec. 12–15, 2006, pp. 845–848.

[17] D. Zhou and V. DeBrunner, "A new active noise control algorithm that does not require secondary path identification," *IEEE Trans. Signal Process.*, vol. 55, no. 5, pp. 1719–1729, May 2007.