Acoustic Echo Cancellation by Solving Wiener–Hopf Equations

Xi-Lin Li, xilli2@cisco.com

CISCO Building 30, 707 E Tasman Dr, Milpitas, CA 95035

Abstract

This report studies the acoustic echo cancellation (AEC) problem. Due to the nonstationarity of speech signals, the conventional stochastic gradient descent acoustic echo cancelers (AEC) typically require two essential components, double talk detector and step size controller, to assist their gradient descent. This makes their designs complicated, unreliably, and their tunings troublesome. With the ever increasing computational capacities of hardwares, least squares adaptive filters become attractive alternatives for AEC applications, and compared with stochastic gradient descent, they converge faster, and are more resistant to noise and double talk interferences, with the cost of higher computational complexities. Unfortunately, many least squares solutions, e.g., the famous recursive least squares (RLS) adaptive filter and its variations, may be numerically unstable in practice. This report presents three new acoustic echo cancellation methods by solving the Wiener–Hopf equations. One of them is a direct Wiener–Hopf equations solver using Cholesky decomposition, and the other two are iterative solutions. All these proposed solutions are numerically stable, have the same order of computational complexities as that of the RLS adaptive filter, and are demonstrated to be useful on real world recorded test sequences.

I. Introduction

Acoustic echoes exist in almost all telecommunication environments. The noticeable presence of acoustic echoes is annoying, and makes smoothing communications not possible. An acoustic echo canceler (AEC) estimates the echoes in the microphone received signal, and cancels the echoes by subtracting the microphone received signal with estimated echoes. Needless to say, AEC is an essential component in any communication device, and as a standard application of adaptive filtering theory [1], it is an interesting research topic as well. Although it is studied as early as 1960s [2], AEC still keeps to be a challenging adaptive filtering problem due to several reasons. First, the speech signal is highly nonstationary, and the stationarity assumption made in many adaptive filtering theories does not hold. Second, an AEC adaptive filter may contain thousands of tap coefficients to learn in a real time manner using limited computational resources. Lastly, the application scenarios, e.g., noise level, sampling rate, quality of loudspeaker, implementation platform, etc., vary wildly.

Stochastic gradient descent methods, e.g., the least mean square (LMS) and affine projection (AP) filters [1], are still the workhorses for AEC since they are simple yet efficient in many situations. However, as the speech signals are nonstationary, we need to freeze the adaptive filter coefficients when the echo reference speech signal is absent or when the near end speech signal dominates over the echoes. Furthermore, by using a fixed step size, a stochastic gradient descent method cannot achieve fast convergence and low steady state excess error at the same time. Thus a practical stochastic gradient descent AEC design always requires two extra components, a double talk detector and a step size controller. Ideally, the AEC adaptive filter coefficients are updated only when the echo dominates and the adaptive filter has not converged yet. However, information regarding the misadjustment of adaptive filters and the existence (absence) of far end (near end) speech signal cannot be easily obtained. Estimation of such information may be inaccurate, leading to unreliable AEC design. This is especially the case when the AEC is implemented in the subband domain where the filter coefficients are updated in a very low frequency to reduce computational load.

With the ever improving hardware computational capacities on communication devices, least squares adaptive filters, e.g., the recurse least squares (RLS) adaptive filter, become affordable for AEC design. Actually, the least squares adaptive filters may be affordable on certain low end devices when the AEC is implemented in the subband domain to reduce its computational load. Least squares adaptive filters typically converge faster and better than stochastic gradient descent filters, and are more resistant to interferences like near end noises and speech signals. Unfortunately, the RLS adaptive filter and many of its variations may not be numerically stable for AEC applications since it recursively calculates the inverse of the auto correlation matrix of the echo reference speech signal, which can be singular or close to singular, causing numerical difficulties. A round robin regularization RLS implementation is proposed in [3] to overcome this issue. Basically, it repeatedly explicitly inverses the regularized autocorrelation matrix after every fixed number of iterations, and restarts the recursive inversion of the autocorrelation matrix from this explicit inverse. It is known that explicit matrix inversion can be problematic in many situations, e.g., for matrices with large dimensions, or on machines with low numerical accuracy, or for fixed point implementation.

This report proposes to solve for the AEC adaptive filter coefficients via solving the Wiener–Hopf equations using those off–the–shelf linear system solvers. We have studied the performances of three selected solutions, Cholesky decomposition, Gauss–Seidel iteration, and a line search method. All these methods are numerically stable, have the same order of complexities as that of the RLS adaptive filter, and their performances are comparable to that of the theoretically optimal Wiener filter. These merits make them attractive alternatives to those classic stochastic gradient descent AEC designs.

II. AEC AND ADAPTIVE FILTERS

A. The AEC Problems

Fig. 1 shows the diagram of a typical AEC system, where:

1) $u(t)$ is the far end speech signal with $t \geq 1$ being the discrete time index. It is sent to the AEC adaptive filter to be used as the echo reference signal, and also to the loudspeaker to be played back.
2) $\boldsymbol{h} = [h_0, h_1, h_2, \ldots]^T$ is a vector modeling the impulse response of the loudspeaker–enclosure–microphone environment.
3) $v(t)$ is the near end signal, typically a mixture of speech signals and noises.
4) $x(t) = v(t) + \sum_{i \geq 0} h_i u(t-i)$ is the microphone signal containing both the near end signal $v(t)$, and acoustic echoes $h_i u(t-i)$. We do not consider nonlinear echo model here.
5) $\boldsymbol{w} = [w_0, w_1, \ldots w_{\ell-1}]^T$ is a finite impulse response (FIR) adaptive filter with ℓ tap coefficients to learn. Ideally, $\boldsymbol{w}$ converges to $\boldsymbol{h}$ such that the adaptive filter can exactly reproduce the echoes.
6) $y(t) = \sum_{i=0}^{\ell-1} w_i u(t-i) = \boldsymbol{w}^T \boldsymbol{u}(t)$ is the estimated echo at time t, where

$$\boldsymbol{u}(t) = [u(t), u(t-1), \ldots, u(t-\ell+1)]^T$$

is the tap input vector at time t.
7) $e(t) = x(t) - y(t)$ is the echo removed signal.

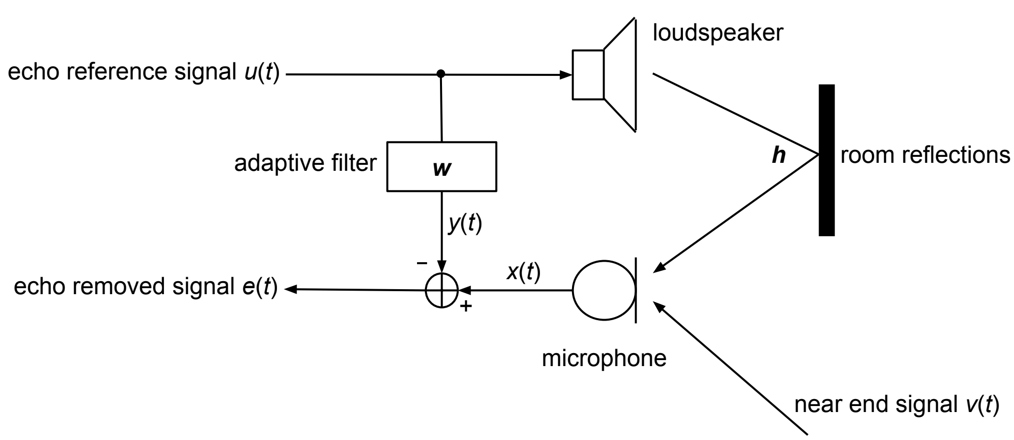

Fig. 1. Diagram of an AEC system.

Since $u(t)$ and $v(t)$ are generated by independent sources, they should be independent as well. Then by minimizing the energy of $e(t)$, we are able to remove any component in $x(t)$ that is linearly correlated to $u(t)$. Mathematically, we can put the AEC problem as

$$\min_{\boldsymbol{w}}: \quad E[e^2(t)], \tag{1}$$

where E denotes expectation. Note that it is possible to use cost functions modeling the super–Gaussian distribution of speech signals, e.g., $E[|e(t)|]$ by assuming a Laplace distribution for the echo removed signal. These cost functions typically lead to double talk resistant stochastic gradient descent AEC designs, but introduce new challenges like the difficulty of performance analysis and step size selection. We stick to the mean squared error due to its simplicity and widely applicability.

B. Adaptive Filters

1) Stochastic Gradient Descent: The well known least mean squares (LMS) adaptive filter updates the filter coefficients using instantaneous gradient descent as

$$\boldsymbol{w}(t+1) = \boldsymbol{w}(t) + \mu e(t)\boldsymbol{u}(t), \tag{2}$$

where $\mu > 0$ is the update step size, and $\boldsymbol{w}$ is time varying, suggested by its index t. Normalized LMS (NLMS) adaptive filter,

$$\boldsymbol{w}(t+1) = \boldsymbol{w}(t) + \mu_0 \frac{e(t)\boldsymbol{u}(t)}{\boldsymbol{u}^T(t)\boldsymbol{u}(t) + \delta_{\rm lms}}, \tag{3}$$

is a more popular choice as the step size here is normalized, i.e., $0 < \mu_0 \ll 1$. Here, $\delta_{\rm lms} > 0$ is used to suppress gradient noise when the tap input vector $\boldsymbol{u}(t)$ is too small, but a large $\delta_{\rm lms}$ will slow down the convergence of NLMS. Derived from the same principle of minimal disturbance, affine projection (AP) adaptive filters are more elaborated versions of the NLMS adaptive filters. Since AP adaptive filters involve matrix inversion, lower order AP adaptive filters are more popular for AEC. However, adjacent speech samples are highly correlated, and thus the performance gain of a lower order AP adaptive filter over the NLMS adaptive filter may be limited. Although numerically stable when used with a small enough step size, these stochastic gradient descent based adaptive filters may converge slow, and are sensitive to the perturbations from near end speech signals and noises. In practice, they are used along with double talk detector and step size controller to avoid divergence.

2) Least Squares Adaptive Filters: By letting the derivative of the cost function in (1) with respect to $\boldsymbol{w}$ be zero, we obtain the closed–form solution for the optimal $\boldsymbol{w}$ as the solution of the following Wiener–Hopf equation

$$\boldsymbol{R}\boldsymbol{w} = \boldsymbol{c}, \tag{4}$$

where $\boldsymbol{R} = E[\boldsymbol{u}(t)\boldsymbol{u}^T(t)]$ is the autocorrelation matrix of the input tap vector, and $\boldsymbol{c} = E[x(t)\boldsymbol{u}(t)]$ is the cross correlation vector between $x(t)$ and $\boldsymbol{u}(t)$. In practice, the expectation is replaced with exponentially decaying weighted average, leading to adaptive solution

$$\boldsymbol{R}(t)\boldsymbol{w}(t) = \boldsymbol{c}(t), \tag{5}$$

with

$$\begin{aligned}\boldsymbol{R}(t) &= \lambda\boldsymbol{R}(t-1) + \boldsymbol{u}(t)\boldsymbol{u}^T(t),\\ \boldsymbol{c}(t) &= \lambda\boldsymbol{c}(t-1) + \boldsymbol{x}(t)u(t),\end{aligned}$$

where $0 \ll \lambda \leq 1$ is a forgetting factor, and $\boldsymbol{R}(0)$ and $\boldsymbol{c}(0)$ are the initial guesses of $\boldsymbol{R}$ and $\boldsymbol{c}$, respectively. Noting that the AEC system may not be static, e.g., a slowly time varying loudspeaker–enclosure–microphone acoustic environment, we always choose $0 \ll \lambda < 1$, and $\lambda = 1$ is not considered. The famous RLS adaptive filter recursively updates the inverse of $\boldsymbol{R}(t)$ using the Woodbury matrix identity [5] as

$$\boldsymbol{R}^{-1}(t) = \lambda^{-1}\boldsymbol{R}^{-1}(t-1) - \frac{\lambda^{-2}\boldsymbol{R}^{-1}(t-1)\boldsymbol{u}(t)\boldsymbol{u}^T(t)\boldsymbol{R}^{-1}(t-1)}{1+\lambda^{-1}\boldsymbol{u}^T(t)\boldsymbol{R}^{-1}(t-1)\boldsymbol{u}(t)}. \tag{6}$$

Although being simple and elegant, this process may encounter numerical difficulties whenever $\boldsymbol{R}(t)$ is close to singular, a situation not rare in AEC. There are several variations of the RLS adaptive filter, notably the square–root and the order–recursive adaptive filters, however, none of them can be guaranteed to be numerically stable.

In this report, we consider several off–the–shelf linear system solvers for solving the Wiener–Hopf equations, and study their performance on real–world AEC problems. Particularly, we are interested in numerically stable solutions having the same order of computational complexities as that of the RLS adaptive filters.

It is a common practice to implement AEC in the subband domain to reduce its computational complexities, with the cost of acceptable processing latency and signal distortions. For AEC purpose, discrete Fourier transform (DFT) modulated filter banks are the most widely used. With slight abuse of notations, we use the same letters to represent the same signals in the time and transformed domains, but with different indices to distinguish them, e.g., $u(n,k)$ denotes the transformed domain signal of $u(t)$ at the nth block and the kth subband. Slightly different from the time domain implementation, signals in the transformed domain may be complex valued. Let the estimated echo of the kth subband be

$$y(n,k) = \boldsymbol{w}^H(n,k)\boldsymbol{u}(n,k), \tag{7}$$

where $\boldsymbol{u}(n,k) = [u(n,k), u(n-1,k), \ldots, u(n-L+1,k)]^T$ is the input tap vector, superscript H denotes Hermitian transpose, and L is the filter length. For the kth subband, its least–squares solutions are given by equations

$$\begin{aligned}\boldsymbol{R}(n,k) &= \lambda\boldsymbol{R}(n-1,k) + \boldsymbol{u}(n,k)\boldsymbol{u}^H(n,k),\\ \boldsymbol{c}(n,k) &= \lambda\boldsymbol{c}(n-1,k) + x^*(n,k)\boldsymbol{u}(n,k),\\ \boldsymbol{R}(n,k)\boldsymbol{w}(n,k) &= \boldsymbol{c}(n,k),\end{aligned}$$

where superscript $*$ denotes conjugate. We present our results based on the subband implementation.

III. Practical AEC Solutions

A. Regularized Wiener–Hopf Equations and Wiener Solution

The regularized Wiener–Hopf equations are given by linear system

$$[\boldsymbol{R}(n,k) + \delta_{\mathrm{ls}}\boldsymbol{I}]\,\boldsymbol{w}(n,k) = \boldsymbol{c}(n,k), \tag{8}$$

where $\boldsymbol{I}$ is a conformable identity matrix, and δ_{ls} is a small positive constant to ensure that the above linear system is well defined. Off–the–shelf algorithms like the Gaussian elimination can be used to solve for $\boldsymbol{w}(n,k)$ directly to obtain the Wiener solution optimal in the least squares sense. However, complexity of such a brutal force solution is $\mathcal{O}(L^3)$, one order higher than that of the RLS solution. For this reason, such a theoretically optimal Wiener solution is impractical for moderate or large L, still, it is good to be used as a benchmark for measuring the performance of other more practical solutions.

B. Cholesky Decomposition

Cholesky decomposition is a standard method for solving linear system with positive definite matrix. For the AEC problem, Cholesky factor of the autocorrelation matrix is saved, updated, and used to solve the linear system. To ensure numerical stability, the diagonal elements of the Cholesky factor are forced to be no smaller than a positive constant.

Let the Cholesky decomposition of $\boldsymbol{R}(n,k)$ be

$$\boldsymbol{\Gamma}^H(n,k)\boldsymbol{\Gamma}(n,k) = \boldsymbol{R}(n,k), \tag{9}$$

where $\boldsymbol{\Gamma}(n,k)$ is an upper triangular matrix with nonnegative diagonal elements. The adaptive filter coefficients are given by

$$\boldsymbol{w}(n,k) = \left\{\lfloor\boldsymbol{\Gamma}(n,k)\rfloor_{\delta_{\mathrm{ls}}^{0.5}}\right\}^{-1}\left\{\lfloor\boldsymbol{\Gamma}(n,k)\rfloor_{\delta_{\mathrm{ls}}^{0.5}}\right\}^{-H}\boldsymbol{c}(n,k), \tag{10}$$

where $\lfloor \mathbf{\Gamma}(n,k) \rfloor_{\delta_{\text{ls}}^{0.5}}$ are the same as $\mathbf{\Gamma}(n,k)$ except that its diagonal elements are forced to $\delta_{\text{ls}}^{0.5}$ whenever the original corresponding ones are smaller than $\delta_{\text{ls}}^{0.5}$. Here, δ_{ls} is used to force the Cholesky factor to be nonsingular in a way different from the diagonal loading method shown in (8). Since $\lfloor \mathbf{\Gamma}(n,k) \rfloor_{\delta_{\text{ls}}^{0.5}}$ is triangular, in the actual implementation, inversions in the above equation are replaced with backward and forward substitutions to reduce the computational complexity to $\mathcal{O}(L^2)$. The Cholesky factor can be updated by

$$\mathbf{\Gamma}(n,k) = \text{cholupdate}\left[\sqrt{\lambda}\lfloor \mathbf{\Gamma}(n-1,k) \rfloor_{\delta_{\text{ls}}^{0.5}},\ \boldsymbol{u}(n,k)\right], \tag{11}$$

where cholupdate denotes the rank–1 update of Cholesky factorization [4], i..e,

$$\mathbf{\Gamma}^H(n,k)\mathbf{\Gamma}(n,k) = \lambda \lfloor \mathbf{\Gamma}^H(n-1,k) \rfloor_{\delta_{\text{ls}}^{0.5}} \lfloor \mathbf{\Gamma}(n-1,k) \rfloor_{\delta_{\text{ls}}^{0.5}} + \boldsymbol{u}^H(n,k)\boldsymbol{u}(n,k),$$

an operation well defined and with complexity $\mathcal{O}(L^2)$ as well. To summarize, the Cholesky decomposition solution involves four steps: rank–1 update, positive definite enforcement, forward substitution and backward substitution. This solution has complexity $\mathcal{O}(L^2)$, the same order as that of the RLS method.

It is worthy to note that the Cholesky decomposition method is closely analogous to the QR-RLS adaptive filters. In fact, the proposed Cholesky method can be regarded as a square–root adaptive filter since it saves and updates the square root of the autocorrelation matrix. Similar to (6), it is possible to calculate the Cholesky decomposition of the inverse of the autocorrelation matrix recursively. However, there is no simple way to guarantee its numerical stability. Also, the involved rank–1 downdate of Cholesky decomposition will have immediate numerical difficulties when the inverse of the autocorrelation matrix becomes indefinite due to roundoff errors.

C. Gauss–Seidel Iteration

Gauss–Seidel iteration is another standard method for solving linear systems [6]. Unlike the Cholesky decomposition method, which directly solves the Wiener–Hopf equations, Gauss–Seidel iteration is an iterative method, and the iterations converge when the matrix is positive definite. Let us decompose $\boldsymbol{R}(n,k)$ into two parts as

$$\boldsymbol{R}(n,k) = \boldsymbol{P}(n,k) + \boldsymbol{Q}(n,k), \tag{12}$$

where $\boldsymbol{P}(n,k)$ is a lower triangular matrix, and $\boldsymbol{Q}(n,k)$ is a strictly upper triangular matrix. Then the Gauss–Seidel iteration is given by

$$\boldsymbol{w}(n,k) = \left[\boldsymbol{P}(n,k) + \delta_{\text{ls}}\boldsymbol{I}\right]^{-1}\left[\boldsymbol{c}(n,k) - \boldsymbol{Q}(n,k)\boldsymbol{w}(n-1,k)\right]. \tag{13}$$

Again, since $\boldsymbol{P}(n,k) + \delta_{\text{ls}}\boldsymbol{I}$ is a triangular matrix, matrix inversion in the above equation should be implemented as forward substitution to reduce computational load. Note that as the Gauss–Seidel iteration is an iterative method, $\boldsymbol{w}(n,k)$ is not an exact solution of (8). However, since the echo paths change slowly, $\boldsymbol{w}(n,k)$ will be close enough to the Wiener solution given enough iterations. Like the RLS adaptive filter, the Gauss–Seidel Wiener–Hopf equations solver has complexity $\mathcal{O}(L^2)$.

The Gauss–Seidel iteration belongs to the family of stationary iterative methods for solving linear systems. We choose it because it is simple, and converges for linear system with positive definite matrix. The Jacobi iteration is another simple method. It converges for diagonally dominant system of linear equations, but not necessarily always converges for linear system with positive definite matrix. More advanced methods from this family will be too complicated for our AEC problem.

D. Line Search

Line search is yet another general optimization method. Let us define fitting error I

$$\boldsymbol{\varepsilon}_1(n,k) = [\boldsymbol{R}(n,k) + \delta_{\rm ls}\boldsymbol{I}]\,\boldsymbol{w}(n-1,k) - \boldsymbol{c}(n,k). \tag{14}$$

It naturally defines a search direction for optimizing $\boldsymbol{w}(n,k)$. Hence the new tap coefficients are give by update

$$\boldsymbol{w}(n,k) = \boldsymbol{w}(n-1,k) - \alpha(n,k)\boldsymbol{\varepsilon}_1(n,k), \tag{15}$$

where $\alpha(n,k)$ is a step size to be determined. By substituting (14) into (15), we have

$$\boldsymbol{w}(n,k) = \{\boldsymbol{I} - \alpha(n,k)\,[\boldsymbol{R}(n,k) + \delta_{\rm ls}\boldsymbol{I}]\}\,\boldsymbol{w}(n-1,k) - \boldsymbol{c}(n,k), \tag{16}$$

which suggests that a good choice of $\alpha(n,k)$ should satisfies

$$0 < \alpha(n,k) < \frac{2}{\max\operatorname{eig}\boldsymbol{R}(n,k) + \delta_{\rm ls}}, \tag{17}$$

where $\max\operatorname{eig}\boldsymbol{R}(n,k)$ denotes the maximum eigenvalue of $\boldsymbol{R}(n,k)$.

Eq. (17) provides a simple way for the selection of $\alpha(n,k)$, but generally not efficient. Let us consider two criteria for the selection of an optimal $\alpha(n,k)$.

1) Minimum fitting error II: One way is to define fitting error 2

$$\boldsymbol{\varepsilon}_2(n,k) = [\boldsymbol{R}(n,k) + \delta_{\rm ls}\boldsymbol{I}]\,\boldsymbol{w}(n,k) - \boldsymbol{c}(n,k), \tag{18}$$

and select $\alpha(n,k)$ by minimizing $\boldsymbol{\varepsilon}_2^H(n,k)\boldsymbol{\varepsilon}_2(n,k)$. Straightforward calculations show that the optimal step size in this sense is

$$\alpha_1(n,k) = \frac{\boldsymbol{\varepsilon}_1^H(n,k)\,[\boldsymbol{R}(n,k) + \delta_{\rm ls}\boldsymbol{I}]\,\boldsymbol{\varepsilon}_1(n,k)}{\boldsymbol{\varepsilon}_1^H(n,k)\,[\boldsymbol{R}(n,k) + \delta_{\rm ls}\boldsymbol{I}]^2\,\boldsymbol{\varepsilon}_1(n,k)}. \tag{19}$$

2) Weighted minimum fitting error II: Since $\boldsymbol{R}(n,k)+\delta_{\text{ls}}\boldsymbol{I}$ is positive definite, we can introduce another criterion by weighting fitting error II as

$$\boldsymbol{\varepsilon}_2^H(n,k)\left[\boldsymbol{R}(n,k)+\delta_{\text{ls}}\boldsymbol{I}\right]^{-1}\boldsymbol{\varepsilon}_2(n,k).$$

The optimal step size following this criterion is

$$\alpha_2(n,k)=\frac{\boldsymbol{\varepsilon}_1^H(n,k)\boldsymbol{\varepsilon}_1(n,k)}{\boldsymbol{\varepsilon}_1^H(n,k)\left[\boldsymbol{R}(n,k)+\delta_{\text{ls}}\boldsymbol{I}\right]\boldsymbol{\varepsilon}_1(n,k)}. \tag{20}$$

The weighted fitting error II criterion only differs from the least squares criterion by certain terms independent of $\boldsymbol{w}(n,k)$. In practice, the two optimal step sizes given in (19) and (20) lead to virtually identical solutions.

Like the Gauss–Seidel method, line search is an iterative method, and $\boldsymbol{w}(n,k)$ is not an exact solution of (8). Still, with enough iterations, the line search solution should be close enough to the Wiener solution when the echo paths change slowly. Complexity of this method is $\mathcal{O}(L^2)$ as well.

IV. Experimental Results

We compare the performance of different AEC solutions on a real world captured test sequence from a laptop computer. Fig. 2 shows the waveforms at different stages. A frequency domain AEC with 256 bins is implemented. The adaptive AEC filter coefficients cover an echo tail length of 100 ms. For NLMS, we choose step size 0.1 and $\delta_{\text{lms}}=2^{-6}$; for the least squares solutions, we choose $\lambda=0.99$ and $\delta_{\text{ls}}=2^{-21}$.

Fig. 3 summarizes the echo return loss enhancement (ERLE) when no DTD is used. ERLE convergence curve of the Cholesky solution is virtually the same as that of the optimal Wiener solution. Gauss–Seidel iteration and line search are two iterative solutions, and compared with the Wiener solution, they show slower convergence rate at the initial stage, but soon catch up with the performance of Wiener solution. Between these two iterative solutions, line search significantly outperforms Gauss–Seidel iteration. NLMS performs the worst. It converges slowly and poorly for the echo only part, and without a DTD, it diverges during the double talk stage.

Fig. 4 summarizes the echo return loss enhancement (ERLE) when DTDs are used. The DTD design is simple. For each AEC method, we keep two running examples simultaneously on the same test sequence. Filter coefficients in the first example are always updated, and it is used to detect echo dominating parts of the sequence. Filter coefficients in the second example are updated only when the ERLE of the first filter is no smaller than 6 dB. From Fig. 4, we find that except the Gauss–Seidel iteration, all the other least squares methods almost converge equally fast and well. NLMS benefits the most from DTD, and becomes stabilized during the double talk period. Still, the performance gap between NLMS and least squares methods are significant.

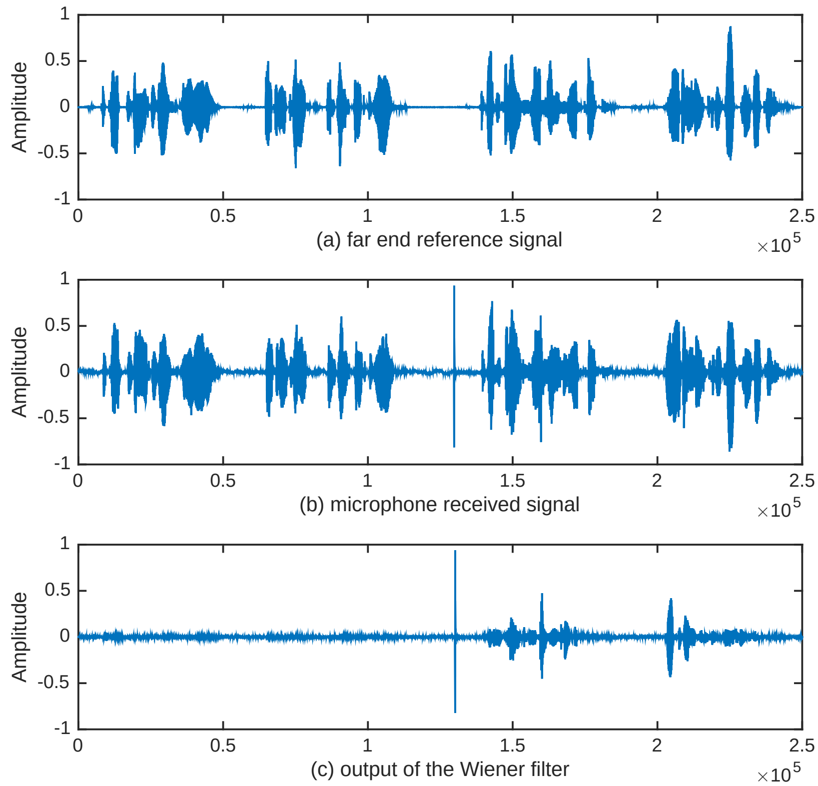

Fig. 2. Real world captured speech signals for AEC testing. The sample rate is 16000 Hz. The first half of the recording is echo only part; the second half of the recording has double talks. Fig. (a) shows the far end reference speech signal; Fig. (b) plots the microphone received signal; Fig. (c) shows the echo removed signal after the optimal Wiener filtering.

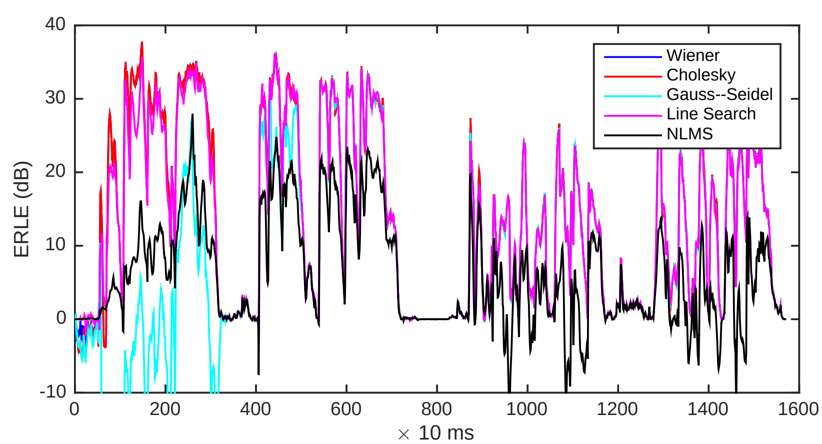

Fig. 3. ERLE convergence curves when no double talk detector is used.

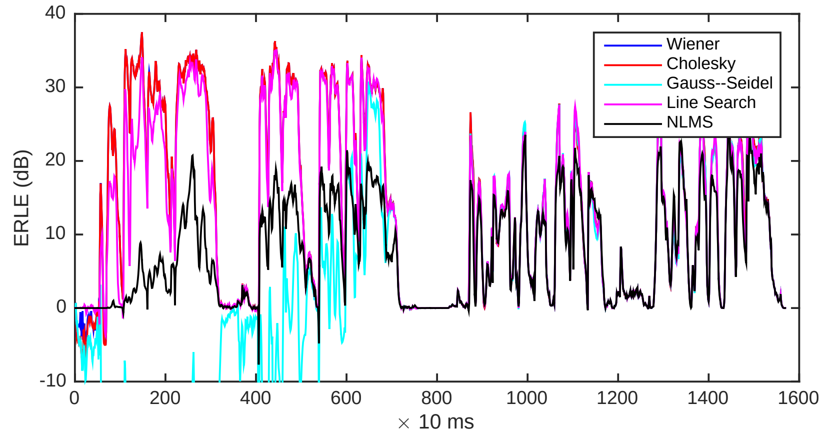

Fig. 4. ERLE convergence curves when double talk detectors are used.

V. Discussions and Conclusions

In this report, we have studied the performance of three AEC (acoustic echo cancellation) solutions, Cholesky decomposition, Gauss–Seidel iteration and line search. All of them have the same order of complexity as that of the recursive least squares (RLS) method, achieve equally good performance as the Wiener solution at the steady state, and are guaranteed to be numerically stable. Unlike those stochastic gradient descent AEC solutions, these least squares solutions converge fast, and are resistant to double talk, making accurate double talk detector (DTD) and fine tuned step size controller less indispensable.

Cholesky decomposition is a direct method for solving the Wiener–Hopf equations, and always achieves the best performance in the least squares sense. However, the rank–1 update of Cholesky decomposition and followed backward and forward substitutions for solving the linear system involves L square root and more division operations, where L is the dimension of the Wiener–Hopf equations. Thus the Cholesky decomposition method might not be suitable for platforms where square root and division operations are computationally expensive, e.g., a fixed point digital signal processor (DSP). The other two methods, Gauss–Seidel iteration and line search are iterative methods. They may show suboptimal performance at the initial converging stage, but their steady state performances can be equally good as that of the Wiener solution. Compared with the Cholesky decomposition method, one significant advantage of the two iterative methods is that they are less sensitive to rounding errors. This make them the ideal choice when the working precision is low. Among these three algorithms, our experiences suggest that line search should be the first considered choice due to its excellent tradeoff between performance and simplicity of implementation. Cholesky decomposition is the most complicated and well performed method, however, it only demonstrates marginal performance gain over line search, and often, the performance difference is too small to be subjectively noticeable. Gauss–Seidel iteration seems have little advantage over the other two methods, and we do not propose it. This is not astonishing since the Gauss–Seidel iteration is not optimal in any sense.

The proposed AEC solutions can be generalized or augmented in several ways to enhance their utilities. One generalization is the application to stereo AEC. By redefining the input tap vector $\boldsymbol{u}(n,k)$ to include both the left and the right echo reference signals, the same AEC solutions work for stereo AEC. Similarly, when $\boldsymbol{u}(n,k)$ includes nonlinearly distorted replica of the original echo reference signal, the proposed solutions can be used for nonlinear AEC. When a single clipping nonlinearity is used, typically we are able to achieve an ERLE gain of $3 \sim 6$ dB. In many application scenarios, the group delay between the echo reference signal and echoes received by the microphone varies in a wide range, e.g., up to 0.5 second on many Android devices. It is not realistic to estimate an adaptive filter with taps covering such a long delay uncertainty range. A convenient solution is to augment the proposed AEC solutions with a delay estimation module, and to assign adaptive filter taps only around the estimated group delay. This delay estimation can be obtained by searching for the location of the largest coefficient in the cross correlation vector, $\boldsymbol{c}(n,k)$.

REFERENCES

[1] S. Haykin, *Adaptive Filter Theory*, 4th Edition, Prentice Hall, 2002.

[2] M. M. Sondhi, “An adaptive echo canceler,” *Bell System Technical Journal*, vol. 46, no. 3, pp. 497–511, Mar. 1967.

[3] J. W. Stokes and J. C. Platt, “Robust RLS with round robin regularization including application to stereo acoustic echo cancellation,” in *ICASSP*, pp. 736–739, Toulouse, France, May 2006.

[4] G. W. Stewart, *Basic Decompositions*, Philadelphia: Soc. for Industrial and Applied Mathematics, 1998.

[5] https://en.wikipedia.org/wiki/Woodbury_matrix_identity

[6] https://en.wikipedia.org/wiki/Gauss%E2%80%93Seidel_method