Acoustic Echo Cancellation and Doubletalk Detection Using Estimated Loudspeaker Impulse Responses

Per Åhgren

Abstract—In this paper, we present a new approach to acoustic echo cancellation and doubletalk detection for a teleconferencing system including a loudspeaker for which an estimate of the loudspeaker impulse response is available. The approach is general in the sense that it may be applied to most existing acoustic echo cancellation and doubletalk detection algorithms. We show that the new approach reduces the computational complexity for both the echo cancellation and the doubletalk detection algorithms. Furthermore, the numerical examples show that the new approach also may increase the echo cancellation and doubletalk detection performances.

Index Terms—Acoustic echo cancellation, adaptive filtering, doubletalk detection, loudspeaker.

Fig. 1. Typical AEC setup.

I. INTRODUCTION

THE problem of acoustic echo cancellation (AEC) was introduced in [1] and is still an active field of research. Acoustic Echo Cancellers are needed for removing the acoustic echoes resulting from the acoustic coupling between the loudspeaker(s) and the microphone(s) in communication systems. In Fig. 1, a typical setup for AEC is shown. The main purpose of the setup is that the near-end speech signal $v(t)$ is to be picked up by the microphone M and propagated to the far-end room while far-end speech is to be emitted by the loudspeaker L into the near-end room. During doubletalk, which is the case when both near-end and far-end speech is present, the near-end speech in the microphone signal $y(t)$ is corrupted by the echo of the far-end speech signal $x(t)$ that is propagated in the near-end room from the loudspeaker L to the microphone M. Therefore, during doubletalk, the resulting microphone signal $y(t)$ consists of near-end speech mixed with far-end speech filtered by the near-end room impulse response $\mathbf{h}$ from the loudspeaker to the microphone

$$y(t) = \mathbf{h}^T \mathbf{x}(t) + v(t) + w(t). \quad (1)$$

In (1), $w(t)$ is noise and the input data vector $\mathbf{x}(t)$ is defined as

$$\mathbf{x}(t) = [x(t) \quad x(t-1) \quad \cdots \quad x(t-n+1)]^T \quad (2)$$

where n is the order of the room impulse response modeled as a finite impulse response (FIR) filter (in this paper we will only
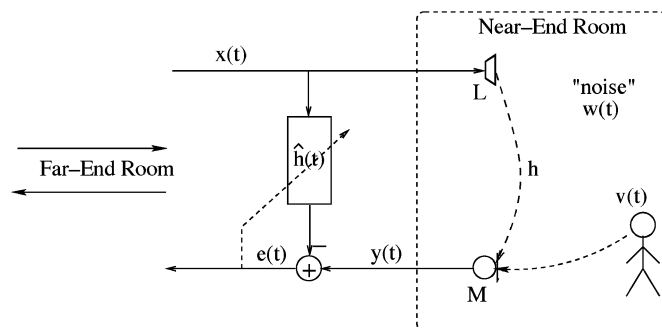
consider FIR filters which is the most common filter type for AEC)

$$\mathbf{h} = [h_0 \quad h_1 \quad \cdots \quad h_{n-1}]^T. \quad (3)$$

The room impulse response is varying with time since movements (e.g., people moving around) may occur in the room. Thus, usually in order to remove the undesired echo an adaptive filter estimate $\hat{\mathbf{h}}(t)$ of $\mathbf{h}$ is used to predict the far-end speech contribution $\mathbf{h}^T \mathbf{x}(t)$ and subtract it from the microphone signal $y(t)$. Thereby, we get the error signal

$$e(t) = y(t) - \hat{\mathbf{h}}^T(t)\mathbf{x}(t) = v(t) + \mathbf{h}^T \mathbf{x}(t) - \hat{\mathbf{h}}^T(t)\mathbf{x}(t) + w(t) \quad (4)$$

that ideally should be equal to the near-end speech signal $v(t)$. Note that in (4), for simplicity, we have assumed that $\hat{\mathbf{h}}(t)$ and $\mathbf{h}$ are of the same length. If that is not the case, then (4) has to be modified accordingly.

When no near-end speech is present the error signal $e(t)$ can be used to adapt the adaptive filter $\hat{\mathbf{h}}(t)$ using some algorithm for filter adaptation. Several different algorithms for filter adaptation in AEC have been proposed [2]. The most common one is perhaps the normalized least-mean squares (NLMS) algorithm [3] which has been shown to perform well for the AEC problem while at the same time having a rather low computational complexity.

When there is doubletalk, however, the near-end speech signal $v(t)$ disturbs the adaptation and can cause the adaptive filter to diverge. Therefore it is important to detect doubletalk in order to stop the filter adaptation when doubletalk is present. Several different algorithms have been proposed for doubletalk detection (DTD) and in this paper we choose to compare the results with the results obtained by the cross-correlation (CR) algorithm [4] and the normalized cross-correlation (NCR) algorithm [5]. We also compare the results with the results

obtained by the computationally cheap approximation of the NCR algorithm (Cheap-NCR) presented in [5].

The AEC algorithms as well as the DTD algorithms are to be run in real-time on a digital signal processor with limited memory and computational power. As the numerical complexities of these algorithms usually are proportional to a power of n (the length of the impulse response $\mathbf{h}$), and n usually is very large, ranging from several hundred to several thousand, it is important to minimize the computational complexity. The main purpose of this paper is to show how the knowledge of the impulse response for the loudspeaker L can be used to reduce the computational complexity of existing AEC and DTD algorithms while at the same time increasing the performance.

II. AEC AND DTD USING ESTIMATED LOUDSPEAKER IMPULSE RESPONSES

In this paper, we propose a new approach to AEC as well as DTD based on the knowledge of the impulse response of the loudspeaker L in Fig. 1. This new approach, which we will denote the loudspeaker-impulse-response (LIME) approach, may be used to modify existing AEC and DTD algorithms.

The LIME approach for the DTD problem we will denote DTD-LIME. As we will see, the DTD-LIME approach use a data model similar to the one in (1). Thus the DTD-LIME approach may probably be used for most existing DTD algorithms working with the model in (1). In this section, we show how the approach can be applied to the NCR algorithm. The reason for choosing the NCR algorithm is that it has a high numerical complexity and that it has been shown to perform well. It turns out that the DTD-LIME approach can significantly reduce the computational complexity of the NCR algorithm while still obtaining a comparable DTD performance.

The LIME approach for AEC filter adaptation algorithms we will denote the AEC-LIME approach. Similarly to DTD-LIME, AEC-LIME may probably be applied to most AEC filter adaptation algorithms as it uses a data model similar to the one in (1). As we will see, the AEC-LIME approach is best used together with the DTD-LIME approach as both have common parts. In the numerical examples we apply the approach to NLMS. It turns out that AEC-LIME yields a similar echo cancellation performance while achieving a lower computational complexity.

The LIME approach is based on the fact that the all far-end speech, and no near-end speech, is filtered by the time-invariant impulse response for the loudspeaker L in Fig. 1. This can be exploited and if we know the loudspeaker impulse response we can modify many of the existing AEC filter adaptation and DTD algorithms. These modifications are described in Sections II-A and II-B. For the LIME approach to be feasible, it is vital that we can somehow obtain the loudspeaker impulse response and this is discussed in Section II-C. In general, many DTD algorithms have problem when the acoustic path changes. This is discussed in Section II-D. In Section II-E, we present and summarize the AEC-LIME and DTD-LIME approaches for an AEC algorithm where NLMS is used as the filter adaptation algorithm, and NCR is used as DTD algorithm. Finally, in Section II-F, we discuss the computational complexities of the unified AEC-LIME and DTD-LIME approach.

A. AEC-LIME Approach

The loudspeaker impulse response $\mathbf{h}$ in (1) includes both the unknown time-varying impulse response $\mathbf{h}_E$ of the echo path in the near-end room, and the time-invariant impulse response $\mathbf{h}_L$ of the loudspeaker of which an estimate $\hat{\mathbf{h}}_L$ is assumed to be available. Assuming these impulse responses can be approximated as linear (which is a common basic assumption in AEC), we can write $\mathbf{h}$ as

$$\mathbf{h} = \mathbf{h}_L * \mathbf{h}_E \tag{5}$$

where $*$ denotes convolution, the loudspeaker impulse response $\mathbf{h}_L$ of length p is defined as

$$\mathbf{h}_L = [h_{L,0} \quad h_{L,1} \quad \cdots \quad h_{L,p-1}]^T, \tag{6}$$

and the echo path impulse response $\mathbf{h}_E$ is defined similarly. If m denotes the length of $\mathbf{h}_E$, we have from (5) that

$$n = p + m - 1. \tag{7}$$

Most AEC filter adaptation algorithms work with the data model in (1). Since we have assumed that we know an estimate $\hat{\mathbf{h}}_L$ of $\mathbf{h}_L$, we can rewrite this equation as

$$y(t) = \mathbf{h}_E^T \bar{\mathbf{x}}(t) + v(t) + w(t), \tag{8}$$

where

$$\bar{\mathbf{x}}(t) = [\bar{x}(t) \quad \bar{x}(t-1) \quad \cdots \quad \bar{x}(t-n+1)]^T \tag{9}$$

$$\bar{x}(t) = \mathbf{h}_L^T \breve{\mathbf{x}}(t) \tag{10}$$

$$\breve{\mathbf{x}}(t) = [x(t) \quad x(t-1) \quad \cdots \quad x(t-p+1)]^T. \tag{11}$$

Since (8) is almost identical to (1), the AEC filter adaptation algorithm can be applied to (8) instead of (1). As $\mathbf{h}_E$ is shorter than $\mathbf{h}$, and the computational complexities of the AEC filter adaptation algorithms usually are proportional to the order of the filter to estimate, the transition from (1)–(8) results in a reduction of the computational complexity for the AEC filter adaptation algorithm. Note, however, that this reduction is only substantial if we have a good estimate of $\mathbf{h}_L$. If the estimate $\hat{\mathbf{h}}_L$ is very poor, we still have to estimate a filter of similar length as $\mathbf{h}$ (using an input signal prefiltered by $\hat{\mathbf{h}}_L$).

B. DTD-LIME Approach

Most DTD algorithms work with the data model in (1) and rely of the fact that $y(t)$ is a filtered version of $x(t)$ (filtered by the impulse response $\mathbf{h}$) and that $v(t)$ is not. In the DTD-LIME approach we modify the data model in (1) and end up with the following model:

$$y(t) = \mathbf{h}_L^T \tilde{\mathbf{x}}(t) + v(t) + w(t) \tag{12}$$

where

$$\tilde{\mathbf{x}}(t) = [\tilde{x}(t) \quad \tilde{x}(t-1) \quad \cdots \quad \tilde{x}(t-n+1)]^T \tag{13}$$

$$\tilde{x}(t) = \hat{\mathbf{h}}_E^T \breve{\mathbf{x}}(t) \tag{14}$$

$$\breve{\mathbf{x}}(t) = [x(t) \quad x(t-1) \quad \cdots \quad x(t-m+1)]^T \tag{15}$$

and $\hat{\mathbf{h}}_E$ is the estimate of $\mathbf{h}_E$ obtained from the model in (8) with AEC-LIME. The computational complexity of many DTD

algorithms are generally proportional to the length of the filter in the AEC data model. Thus, by applying the DTD algorithms to the model in (12) instead of the model in (1), we will lower the computational complexity of the algorithms significantly since the filter $\mathbf{h}_L$ in (12) generally is much shorter than the filter $\mathbf{h}$ in (1).

C. Estimation of the Loudspeaker Impulse Response

The impulse response of a loudspeaker may be obtained in different ways. The best, and perhaps most direct way, is to compute it from measurements taken in an anechoic chamber. There are, however, also methods for computing the impulse response from measurements taken in an ordinary echoic room [6].

If the loudspeaker impulse responses were time-varying the LIME-approach would not be feasible. Fortunately, it seems that the loudspeaker impulse responses are relative time-invariant, at least for more sophisticated loudspeakers. However, no scientific results have been published about this, instead this property has simply been assumed by the industry and the assumption seem to be correct. Indeed, this time-invariance is a property used by music products such as the *Dirac Research Corrector* that can compensate for the acoustic properties of loudspeakers [7].

It should also be noted that what we mean by the loudspeaker impulse response is the part of the impulse response that corresponds to the electronics in the loudspeaker and the amplifier. It is clear that the loudspeaker impulse response is highly dependent on what direction to the loudspeaker it is measured for. What we are interested in is, however, the part that is directional independent (the case is the same for the *Corrector* product mentioned above).

D. Sensitivity to Changes in the Acoustic Path

A case that many DTD algorithms have problems with is when the acoustic path between the loudspeaker and the microphone changes. Often these changes are detected as doubletalk. Unfortunately, the DTD-LIME approach is definitely sensitive to this. This is easily seen from the model in (12) where the input signal $\tilde{\mathbf{x}}(t)$ is dependent on $\mathbf{h}_E$. If $\mathbf{h}_E$ changes a lot, this model will not be valid anymore. For small changes in $\mathbf{h}_E$ the model should still be applicable but large changes in $\mathbf{h}_E$ will be detected as doubletalk. Another DTD algorithm the same problem appears for is Cheap-NCR which is dependent on an estimate of $\mathbf{h}$ that has to be valid. For DTD-LIME, as well as for Cheap-NCR, there are several practical solutions to this problem. One way is to detect the changes in the acoustic paths separately. However, the simplest way is probably to use a snapshot of the adaptive filter estimate computed just before the doubletalk was detected to cancel the echo, and continue to adapt the filter during the doubletalk and use the most recent adaptive filter estimate in the DTD. A study of these solutions is, however, not included in this paper.

E. AEC-LIME and DTD-LIME Approaches Applied to NLMS and NCR

The AEC-LIME and DTD-LIME approaches are summarized in the steps below where the DTD-LIME approach is

applied to the NCR algorithm and the AEC-LIME approach is applied when NLMS is the adaptive algorithm. We will assume that we have previously computed an estimate $\hat{\mathbf{h}}_L$ of the loudspeaker impulse response $\mathbf{h}_L$.

i) Compute

$$\bar{x}(t) = \hat{\mathbf{h}}_L^T \breve{\mathbf{x}}(t). \quad (16)$$

ii) If doubletalk is not present, compute (adaptively and recursively in time) an estimate $\hat{\mathbf{h}}_E$ of $\mathbf{h}_E$ from $y(t)$ and $\bar{x}(t)$ using NLMS and use $\hat{\mathbf{h}}_E^T \bar{\mathbf{x}}(t)$ as the estimate of the echo signal used to cancel the echo. (This is the AEC part.)

iii) Compute

$$\tilde{x}(t) = \hat{\mathbf{h}}_E^T \breve{\mathbf{x}}(t). \quad (17)$$

iv) Directly applying the NCR algorithm developed in [5] on $y(t)$ and $\tilde{\mathbf{x}}(t)$ we get the following decision variable

$$\xi(t) = \frac{1}{\sigma_y(t)} \sqrt{\mathbf{r}_{\tilde{\mathbf{x}}y}^T(t) \mathbf{R}_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}}^{-1}(t) \mathbf{r}_{\tilde{\mathbf{x}}y}(t)}. \quad (18)$$

For this decision variable, we have that doubletalk is detected at time sample t if $\xi(t) \geq T$, and not detected if $\xi(t) < T$, where T is constant threshold that should be chosen to minimize the probability of false alarm (P_f), as well as the probability of missed detection (P_m) (defined in Section III-A).

In (18), $\mathbf{R}_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}}$ and $\mathbf{r}_{\tilde{\mathbf{x}}y}$ are defined as

$$\mathbf{R}_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}}(t) = E\left\{\tilde{\mathbf{x}}(t)\tilde{\mathbf{x}}^T(t)\right\}, \quad \mathbf{r}_{\tilde{\mathbf{x}}y}(t) = E\left\{\tilde{\mathbf{x}}(t)y(t)\right\} \quad (19)$$

where E denotes the expectation operator. The standard deviation $\sigma_y(t)$ is defined as $\sigma_y(t) = \sqrt{Ey^2(t)}$. In practice, estimates $\hat{\mathbf{R}}_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}}$, $\hat{\mathbf{r}}_{\tilde{\mathbf{x}}y}$ and $\hat{\sigma}_y(t)$ are used in (18) instead of $\mathbf{R}_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}}$, $\mathbf{r}_{\tilde{\mathbf{x}}y}$ and $\sigma_y(t)$. In this paper, we choose to compute these over a sliding time window of length q

$$\hat{\mathbf{R}}_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}}(t) = \frac{1}{q} \sum_{k=0}^{q-1} \tilde{\mathbf{x}}(t-k)\tilde{\mathbf{x}}^T(t-k) \quad (20)$$

$$\hat{\sigma}_y(t) = \sqrt{\frac{1}{q} \sum_{k=0}^{q-1} y^2(t-k)} \quad (21)$$

$$\hat{\mathbf{r}}_{\tilde{\mathbf{x}}y}(t) = \frac{1}{q} \sum_{k=0}^{q-1} \tilde{\mathbf{x}}(t-k)y(t-k). \quad (22)$$

It is important to note that applying the DTD-LIME algorithm for a loudspeaker impulse response of length 1 ($p = 1$) is not a good idea, since for $p = 1$ ideally (for correctly estimated $\hat{\mathbf{h}}_E$ and without any noise) we have $y(t) = h_{L,0}\tilde{x}(t)$ and thus $\hat{\mathbf{r}}_{\tilde{\mathbf{x}}y}^T(t)\hat{\mathbf{R}}_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}}^{-1}(t)\hat{\mathbf{r}}_{\tilde{\mathbf{x}}y}(t) = \hat{\sigma}_y^2(t)$ and we get $\xi(t) \equiv 1$ in (18). Furthermore, for small values of p, the DTD-LIME approach can probably be expected to perform poorly since the extra information added by using the known impulse response is minor [e.g., when applied to the NCR algorithm fewer correlation lags are used for computing $\xi(t)$ in (18)].

TABLE I
SUMMARY OF THE NUMBER OF MULTIPLICATIONS PER SAMPLE FOR AN AEC
SETUP WITH NLMS AS FILTER ADAPTATION ALGORITHM AND NCR AS DTD
ALGORITHM WITH AND WITHOUT THE LIME APPROACH

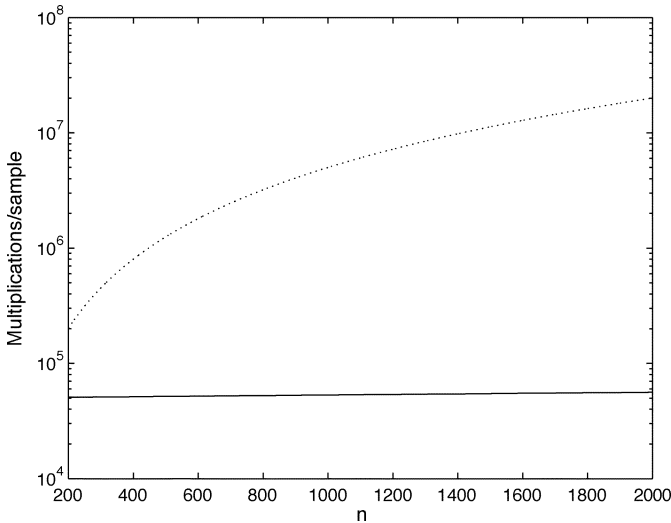| Step	NCR variant with LIME	NCR variant without LIME
i	p	0
ii	$2m+1$	$2n+1$
iii	m	0
iv	$5p^2+4p$	$5n^2+4n$
Total	$5p^2+5p+3m+1$	$5n^2+6n+1$

Fig. 2.   Required multiplications per sample as a function of the filter length n for the AEC setup with NLMS as adaptive algorithm and NCR as DTD algorithm with (solid) and without (dotted) the LIME approach.

F. Numerical Complexity Comparison

In this Section, we compare the computational complexities for an AEC setup where NLMS is used as the adaptive algorithm and NCR is used as DTD algorithm for the cases when the LIME approach is used and when it is not. Note that $\hat{\mathbf{R}}_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}}^{-1}(t)$ and $\hat{\mathbf{r}}_{\tilde{\mathbf{x}}y}(t)$ can be computed recursively in time using the matrix inversion lemma [8], requiring $4p^2+2p$ multiplications and $2p$ multiplications per time sample, respectively (this is for the case when $\hat{\mathbf{R}}_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}}^{-1}(t)$ and $\hat{\mathbf{r}}_{\tilde{\mathbf{x}}y}(t)$ are computed recursively over a sliding time window). The number of multiplications required for computing the NCR decision variable for each time sample are then easily found to be (not counting the square root) $5p^2+4p$ when the LIME approach is used. The total numbers of multiplications required per time sample for an AEC setup using NLMS as adaptive algorithm and NCR as DTD algorithm are presented in Table I for the cases when the LIME-approach is used and when it is not used. It is clear that the computational complexity of the AEC setup is much higher without the LIME approach than with the LIME-approach. To further illustrate the gain in computational performance, in Fig. 2 we show the number of multiplications per sample for the algorithms in Table I as a function of n. In the figure, we use a somewhat typical value

of $p=100$. Again, it is clear that the LIME-approach offers a significant reduction in the computational complexity.

Note that we have chosen not to compare with the numerical complexities when the LIME approach is applied to the other two DTD algorithms (CR and Cheap-NCR) used in the numerical examples. The reason for this is that for these the gain in computational complexity is minor since the numerical complexity of the CR algorithm is only proportional to n, and the Cheap-NCR algorithm can be easily be shown to require just a few multiplications to be computed and just a few values to be stored when implemented in a sliding window manner [9].

III. NUMERICAL EXAMPLES

To evaluate the performance of the DTD algorithms with the LIME-approach, we have used an evaluation scheme similar to the one that was proposed in [4]. This scheme is described in Section III-B. In Section III-A, some basic definitions are given and in Section III-C the results of the numerical simulations are presented.

A. Definitions

The probability of missed detection (P_m), and the probability of false alarm (P_f) are defined as

$$P_m = \frac{N_{Dm}}{N_D}, \quad P_f = \frac{N_{Df}}{N_{ND}} \qquad (23)$$

where N_{Dm} is the number of samples where doubletalk was not detected but was present, N_D is the total number of samples where doubletalk was present, N_{Df} is the number of samples where doubletalk was detected but where no doubletalk was present, and N_{ND} is the total number of samples where doubletalk was not present.

The near-end to far-end speech ratio (NFR), and the signal-to-noise ratio (SNR) are defined as

$$\mathrm{NFR} = 10\log\left(\frac{E\left[y(t)-v(t)\right]^2}{Ev^2(t)}\right) \qquad (24)$$

$$\mathrm{SNR} = 10\log\left(\frac{E\left[y(t)-w(t)\right]^2}{Ew^2(t)}\right) \qquad (25)$$

where $y(t)$, $w(t)$ and $v(t)$ are defined in (1).

The echo return loss enhancement (ERLE) is a measure of the echo cancellation performance, defined as

$$\mathrm{ERLE}(t) = -10\log\left(\frac{E\left[e^2(t)\right]}{E\left[y^2(t)\right]}\right) \qquad (26)$$

$$\approx -10\log\left(\frac{\sum_{k=0}^{K-1} e^2(t-k)}{\sum_{k=0}^{K-1} y^2(t-k)}\right), \qquad (27)$$

where K is the length of a window over which the ERLE is computed. Note that the ERLE measure is only applicable when there is no near-end speech present.

The misalignment is a measure of how well the adaptive filter $\hat{\mathbf{h}}(t)$ in an AEC setup approximates the true filter $\mathbf{h}$

$$\text{Misalignment}\left(\hat{\mathbf{h}}(t)\right) = 10\log\left(\frac{\left\|\hat{\mathbf{h}}(t) - \mathbf{h}\right\|^2}{\|\mathbf{h}\|^2}\right). \quad (28)$$

Note that if the lengths of $\hat{\mathbf{h}}(t)$ and $\mathbf{h}$ differ, the shorter one is padded with zeros when computing the misalignment

B. DTD Algorithm Evaluation Scheme

i)   Generate 2 s of data according to the model in (1) without any doubletalk present $(v(t) \equiv 0)$.

ii)   Apply the detector to the data and choose a threshold T that gives a P_f of 0.1.

iii)   Create nine different data sets, each in which one of three different 1/2-s speech samples are added in three different positions into the original data set from step i).

iv)   Apply the detector to all the nine data sets and compute the average probability of missed detection (P_m).

C. Simulations

The model in (1) is used to generate the data. The impulse response $\mathbf{h}$ in (1) is obtained in an ordinary office room using an AEC setup with a loudspeaker with known (computed in an anechoic chamber) impulse response $\mathbf{h}_L$. For the simulations, a sampling frequency of 8 kHz is used in order to keep the computational complexity of the simulations for NCR (without the LIME approach) reasonably low (the low sampling frequency allowed using shorter impulse responses, thereby lowering the computational complexity).

In the first numerical example, the doubletalk detection performance of the CR, NCR and Cheap-NCR algorithms with, and without, the LIME approach are tested using the evaluation scheme presented in Section III-B. As far-end speech signal a 2-s speech sample is used and three 1/2-s speech samples are used for the near-end speech signals. The total room impulse response (including the loudspeaker impulse response) has a length of 250 filter taps (the reason for performing the simulation with so short impulse responses is mainly that the NCR algorithm without the LIME approach is too computationally complex to allow much longer filters) and the loudspeaker impulse response is truncated to a length of 75. The length q of the sliding data window used to compute $\hat{\mathbf{R}}_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}}$, $\hat{\mathbf{r}}_{\tilde{\mathbf{x}}y}$ and $\hat{\sigma}_y(t)$ is set to $q = 500$. The estimate $\hat{\mathbf{h}}_E$ of the echo paths used in the detector is estimated from 2 s of data generated using the model in (1) without any doubletalk. The detectors are evaluated for different NFR and SNR and the results are displayed in Figs. 2–5, where P_m is plotted as a function of the NFR. It is clear from the figures that the NCR, Cheap NCR and CR algorithms with the LIME approach outperforms their counterparts without the LIME approach when the SNR is reasonably high (above 10 dB). The reason that the LIME approach works poorly for low SNR is probably that the estimate of $\mathbf{h}_E$ computed in the LIME approach is too poor for the model in (12) to be sufficiently accurate. It may seem strange that in general the
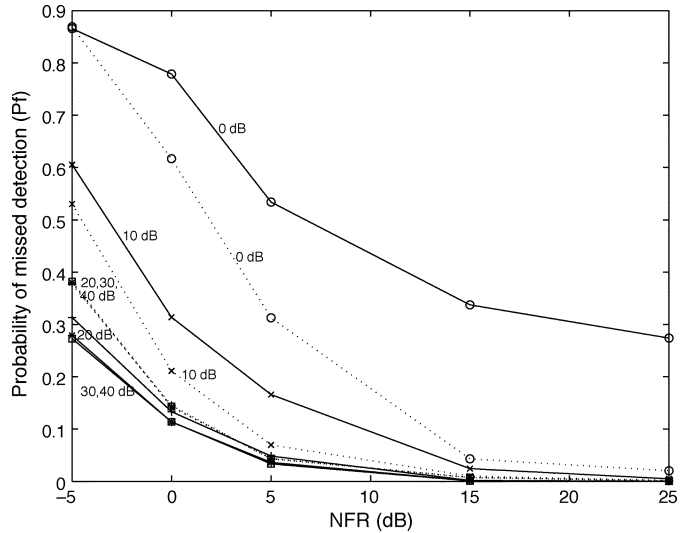

Fig. 3. Probability of miss for the NCR algorithm with the LIME approach (solid) and for the NCR algorithm without the LIME approach (dotted) as a function of the NFR for different values of the SNR (marked as numbers in the plot).
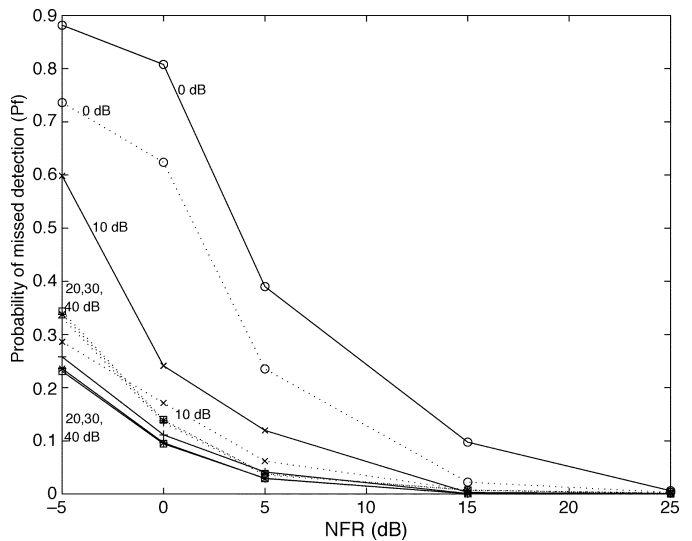

Fig. 4. Probability of miss for the Cheap-NCR algorithm with the LIME approach (solid) and for the Cheap-NCR algorithm without the LIME approach (dotted) as a function of the NFR for different values of the SNR (marked as numbers in the plot).

results of the Cheap-NCR algorithm are better than those obtained by the NCR algorithm. One should, however, be careful when comparing the results of the NCR and Cheap-NCR algorithms since it is hard to make a fair comparison. For instance, the Cheap-NCR algorithm requires that an estimate of $\mathbf{h}$ is computed before it is used. Thus, in a sense, it uses more data than the NCR algorithm that uses a data window of length $q = 500$ and can thereby achieve better results than the NCR algorithm even though it is just an approximation of NCR.

In the second numerical example, we study the performance of the AEC-LIME approach applied to an AEC-setup where NLMS is the adaptive algorithm. As far-end speech signal a 10-s speech sample is used, and the total impulse responses are 550 long. The loudspeaker impulse response (that is common to all total impulse responses used in the simulation) is of length 100.
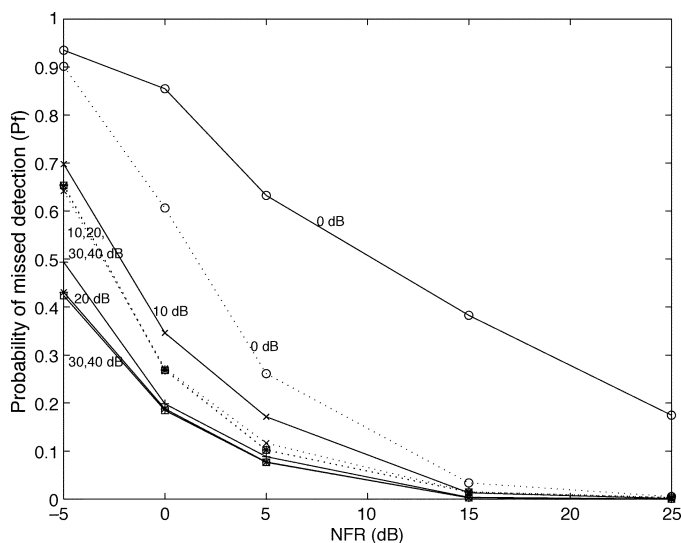
Fig. 5. Probability of miss for the CR algorithm with the LIME approach (solid) and for the CR algorithm without the LIME approach (dotted) as a function of the NFR for different values of the SNR (marked as numbers in the plot).
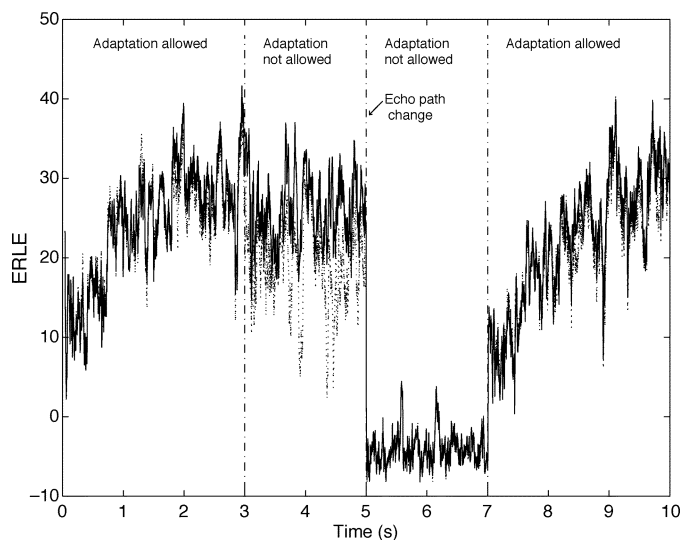

Fig. 6. Echo cancellation performance in terms of ERLE as a function of time for NLMS with AEC-LIME (solid) and NLMS without AEC-LIME (dotted).

The length of the filters estimated by NLMS with and without the LIME-approach are set to 450 and 500, respectively. The SNR is set to 35 dB. In order to simulate a reasonably realistic AEC-setup, we introduced changes in $\mathbf{h}_E$. During the first 5 s, $\mathbf{h}_E$ is kept constant. After 5 s, $\mathbf{h}_E$ is changed abruptly (corresponding to somebody suddenly blocking or moving the loudspeaker or microphone) and then again kept constant for the rest of the simulation. Furthermore, filter adaptation is not allowed from 3 to 7 s, corresponding to a doubletalk situation. Note, however, that we did not add any near-end speech as the ERLE measure is only valid when there is no near-end speech present. This does, however, not modify the interpretation of the simulation results. The simulation results are shown in Fig. 6, where the ERLE is plotted as a function of time.

As we can see NLMS without the LIME approach performs similarly to NLMS with the LIME approach when there is no
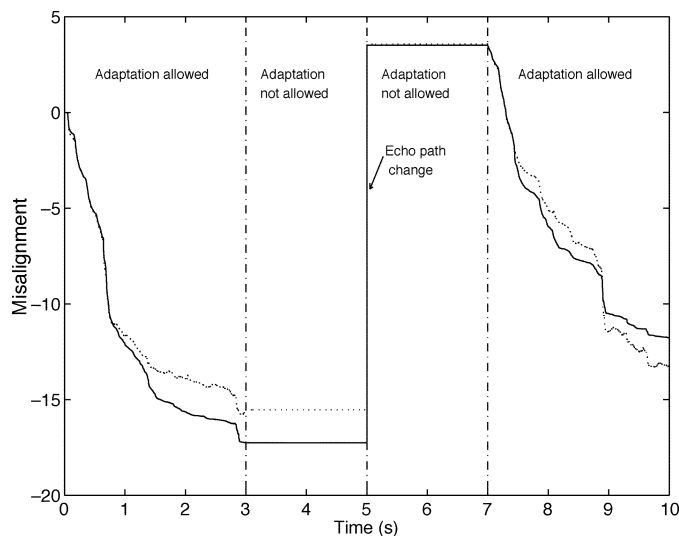

Fig. 7. Echo cancellation performance in terms of misalignment as a function of time for NLMS with AEC-LIME (solid) and NLMS without AEC-LIME (dotted).

doubletalk. However, when there is doubletalk (and filter adaptation is not allowed), NLMS with the LIME approach performs better than NLMS without the LIME approach. After the change in $\mathbf{h}_E$ at 5 s, both algorithms performs poorly, but that is to be expected as the previous estimates for $\mathbf{h}_E$ are inaccurate after the change. In Fig. 7, the results for the same simulation are displayed in terms of misalignment. Again we see that NLMS with the LIME approach performs similarly to NLMS without the LIME approach. It is clear that using the LIME-approach for AEC it is possible to reduce the length of the adaptive filter, and still get a comparable, or even better, AEC performance.

IV. CONCLUSION

We have proposed a new approach to doubletalk detection and acoustic echo cancellation that can be used for most doubletalk and echo cancellation algorithms. When applied to some doubletalk detection algorithms it may offer a lower computational complexity. Furthermore, the numerical examples show that when applied to the NCR, CR, and Cheap-NCR doubletalk detection algorithms it may also improve the doubletalk detection performance for reasonably high SNR.

When applied to echo cancellation algorithms, the approach offers a minor improvement in computational complexity. However, as the simulations show, it may improve the echo cancellation performance.

REFERENCES

[1] M. M. Sondhi, "An adaptive echo canceler," *Bell Syst. Tech. J.*, vol. XLVI, no. 3, pp. 497–510, 1967.
[2] C. Breining, P. Dreiseitel, E. Hänsler, A. Mader, B. Nitsch, H. Puder, T. Schertler, G. Schmidt, and J. Tilp, "Acoustic echo control–an application of very-high-order adaptive filters," *IEEE Signal Process. Mag.*, vol. 16, no. 4, pp. 42–69, Jul. 1999.
[3] S. Haykin, *Adaptive Filter Theory*, 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 1996.
[4] J. H. Cho, D. R. Morgan, and J. Benesty, "An objective technique for evaluating doubletalk detectors in acoustic echo cancelers," *IEEE Trans. Speech Audio Process.*, vol. 7, no. 6, pp. 718–724, Nov. 1999.

[5] J. Benesty, D. R. Morgan, and J. H. Cho, "A new class of doubletalk detectors based on cross-correlation," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 2, pp. 168–172, Mar. 2000.

[6] P. Åhgren and P. Stoica, "A simple method for estimating the impulse responses of loudspeakers," *IEEE Trans. Consumer Electron.*, vol. 49, no. 4, pp. 889–893, Nov. 2003.

[7] (2003, Jul.) Corrector. [Online] Available: http://www.diracresearch.se

[8] P. Stoica and R. Moses, *Introduction to Spectral Analysis*. Upper Saddle River, NJ: Prentice-Hall, 1997.

[9] P. Åhgren, A New Doubletalk Detection Algorithm With a Very Low Computational Complexity, preprint, 2004.

Per Åhgren received the Ph.D. degree in electrical engineering (with a specialization in signal processing) in April 2004 from the Department of Systems and Control, Uppsala University, Uppsala, Sweden.

Since August 2004, he has held a Postdoctorate position at the Linnaeus Centre for Bioinformatics, Uppsala University. His research interests include signal processing for acoustic echo cancellation, doubletalk detection, sterophonic acoustic echo cancellation, adaptive filtering, array processing, system indentification, QTL analysis, and bioinformatics in general.