MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Speech Recognizer Based Maximum Likelihood Beamforming

Bhiksha Raj Michael Seltzer* Manuel Reyes†

TR-2003-087 July 2003

Abstract

In this paper we present a speech-recognizer-based maximum-likelihood beamforming technique, that can be used both for signal enhancement and speaker separation. The presented techniques uses an HMM-based speech recognizer as a statistical model for the target signal to be enhanced or separated. The parameters of a filter-and-sum array processor are estimated to maximize the likelihood of the output as measured using the speech recognizer. The filter-and-sum operation may be performed either in the time domain or the frequency domain. When used for speaker separation, the beamforming must be performed individually for each of the speakers. Since the competing signal is also in-domain speech in this case, the statistical model used for the beamforming is now a factorial HMM formed from the HMM for the target, and that for the competing speakers(s).

*Microsoft Corporation
†Columbia University

Publication History:

1. First printing, TR-2003-087, July 2003

Speech Recognizer based Maximum Likelihood Beamforming

Bhiksha Raj[1], Michael Seltzer[2], and Manuel Reyes[3]
1. Mitsubishi Electric Research Labs, Cambridge, MA 02139
2. Microsoft Corporation, Redmond, WA, 98052
3. Columbia University, New York, NY, 10027

Abstract

In this paper we present a speech-recognizer-based maximum-likelihood beamforming technique, that can be used both for signal enhancement and speaker separation. The presented technique uses an HMM-based speech recognizer as a statistical model for the target signal to be enhanced or separated. The parameters of a filter-and-sum array processor are estimated to maximize the likelihood of the output as measured using the speech recognizer. The filter-and-sum operation may be performed either in the time domain or the frequency domain. When used for speaker separation, the beamforming must be performed individually for each of the speakers. Since the competing signal is also in-domain speech in this case, the statistical model used for the beamforming is now a factorial HMM formed from the HMM for the target, and that for the competing speaker(s).

1. Introduction

The signal to noise ratio (SNR) of recorded speech signals can be considerably enhanced by recording them through multiple microphones simultaneously, and combining the recordings properly. The manner in which the multiple recordings must be combined in order to obtain the best results has been the subject of much research over the years.

The simplest array processing method is delay-and-sum beamforming [1]. Signals from any source must travel different distances to the different microphones, the recordings of which are consequently delayed with respect to each other. Delay-and-sum beamforming consists of aligning the recordings with respect to each other, in order to cancel out these relative delays, and averaging them. Any interfering noise signals from sources that are not exactly coincident with the speech source remain misaligned and are attenuated by the averaging. It can be shown that if the noise signals corrupting each microphone channel are uncorrelated to each other and the target speech signal, delay-and-sum beamforming results in a 3 dB increase in the SNR of the output signal for every doubling of the number of microphones in the array [1].

The term “beamforming” derives from the fact that such processing can be shown to selectively enhance signals from a narrow beam of locations around the desired source. The narrower the beam, the better the ability of the array to pick up the desired source. The beam width and directivity of the delay-and-sum beamformer can be improved by increasing the number of microphones in the array, and by appropriate geometric arrangement of the microphones.

Far more effective than delay-and-sum beamforming is filter-and-sum beamforming. In this method, the signal recorded by each microphone channel is filtered by an associated filter, before the various channels are averaged. The spatial characteristics of the beamformer can be controlled by modifying the parameters of the microphone filters.

The design of beamformers usually involves the estimation of filter parameters such that the desired signal is maximally enhanced. Unfortunately, the desired signal cannot be known beforehand, and the actual design process optimizes alternative criteria that are expected to relate to the enhancement achieved on the desired signal. Sidelobe cancellation techniques design the array filters to attenuate signal energy from locations where the signal is known not to be [2]. Noise suppression methods design the array to suppress a known or estimated noise signal [3]. Least squares methods attempt to maximize the SNR of the processed signal using estimates of the power spectrum of the desired speech signal (e.g. [4]).

All of these methods are based on estimates or knowledge of some aspects of either the desired speech signal or the noise. In this paper we hypothesize that a beamforming algorithm might be aided by detailed knowledge of the structure of the signal that we wish to capture with the array. We have at our disposal large corpora of speech, running into several hundred hours of of recorded and carefully transcribed data. Often we even have a reasonably good idea of the kind of things that might be said, such as when where microphone arrays are employed to record speech for a recognizer that performs a command and control task. We hypothesize that a detailed statistical model that captures all the apposite information in these corpora could be used to guide a beamforming algorithm very effectively.

We choose an HMM-based state-of-the art speech recognizer as our statistical representation of choice. HMM-based recognizers model the distribution of sound units, typically phonemes, as HMMs. They incorporate phonotactic constraints about the manner in which sounds can follow one another through phonetic dictionaries that list valid phoneme sequences that map onto words. Additional constraints are incorporated through the use of *context-dependent* models, that restrict the distribution of any sound unit based on the identity of the adjacent units. Finally, recognizers also incorporate detailed information about the expected language, through grammars or N-gram language models that assign probabilities to different word sequences. The statistical parameters of all of the HMMs are learnt from large corpora of recorded speech. Linguistic constraints are typically learnt from large corpora of text.

The HMM-based speech recognition system thus provides stringent statistical constraints on the characteristics of a proper speech signal. It is therefore reasonable to assume that if the beamformer were optimized such that the output of the signal best conforms to the detailed constrains encoded by the

recognizer, the beamformer would be better able to pick out a speech signal from and exclude other non-speech sounds. Additionally, since it may be possible to actually selectively enhance speech signals that conform to the *language* expected by the recognizer, thereby leading to the possibility of selectively enhancing speech signals relating to a particular topic from a medley of speech signals.

The beamforming algorithm presented in this paper is based on the above hypothesis. It utilizes the speech recognizer to provide statistical constraints on the output of the beamformer. Specifically, it attempts to optimize beamformer parameters such that the likelihood of the output of the array, as measured by a speech recognizer, is maximized.

Beamforming can not only be used to enhance a speech signal in a mix of speech and non-speech signals, but also to separate out the signals from multiple users who are speaking simultaneously. This may be achieved by beamforming separately for each of the speakers and extricating their signal from the medley. The beamforming algorithm presented in this paper is easily extended to this problem. The additional twist here is that when extracting the signals for any speaker, we must consider the fact the signals from the other speakers (that we wish to suppress) may also satisfy the statistical constraints of the recognizer. Thus the algorithm must be modified to include compounded statistical models that simultaneously model both the speech from the desired speaker, and that from the competing speaker(s).

In the rest of this paper we describe the beamforming algorithm, both for signal enhancement and speaker separation. We note that the statistical constraints in an HMM-based speech recognition system can be separated into acoustic constraints, captured by the HMMs for the context-dependent phonetic units, and linguistic constraints, captured by the language model used by the recognizer. For the case of signal enhancement we present a complete algorithm that utilizes statistical acoustic and linguistic constraints. For the speaker separation case we present a simpler version of the algorithm that requires deterministic language constraints, and only the acoustic constraints are statistical. For the latter case we emphasize that we do not aim to present a state-of-the-art speech separation system; rather we only intend to demonstrate the feasibility of utilizing detailed statistical models of speech for speaker separation.

In Section 2 we outline the basic filter-and-sum strategy used by the algorithm for beam forming. Section 3 presents a brief description of versions of the beamforming algorithm that use only acoustic, and both acoustic and linguistic constraints from the recognizer. In Section 4 we present experimental results for the beamformer. For the purposes of experimental evaluation, we use speech recognition performance, rather than subjective human perception, as a metric. In Section 5 we present a factorial HMM based modification of the beamforming algorithm for separating signals from multiple simultaneous speakers. In Section 6 we present experimental evaluation of the latter algorithm. Finally, in Section 7 we present our conclusions and discuss avenues of future work.

2. Filter-and-sum array-processing

We employ traditional filter-and-sum processing to combine the signals captured by the array. In an optional first step the speech source is localized and the relative channel delays caused by path length differences to the source are resolved so that all waveforms captured by the individual microphones are aligned with respect to each other. Several algorithms have been proposed in the literature to do this [5], and any of them can be applied here. In our work we have employed simple cross-correlation to determine the delays among the multiple channels. However, the algorithm has been experimentally verified to work equally well then the signals are not aligned beforehand. In this case our algorithm automatically estimates the appropriate delays for the filters, at the cost of additional computation.

Once the signals are time aligned, each of the signals is passed through an FIR filter whose parameters are determined by the calibration scheme described in the following section. The filtered signals are then added to obtain the final signal. This procedure can be mathematically represented as follows:

$$y[n] = \sum_{i=1}^{N} h_i[n] \otimes x_i[n - \tau_i] \quad (1)$$

where $x_i[n]$ represents the n^{th} sample of the signal recorded by the i^{th} microphone, τ_i represents the delay introduced into the i^{th} channel to time align it with the other channels, $h_i[k]$ represents the k^{th} coefficient of the FIR filter applied to the signal captured by the i^{th} microphone, $\otimes$ represents the convolution operation, and $y[n]$ represents the n^{th} sample of the final output signal. N is the total number of microphones in the array.

3. Beamformer Design

Figure 1 shows the overall design of the filter optimization procedure. The goal of the algorithm is to choose the filter parameters $h_i[k]$ that maximize the likelihood of $y[n]$, the output of the array, as measured by the recognizer. We distinguish between two versions of the algorithm: a) a calibration algorithm, that utilizes only statistical acoustic constraints from the recognizer and b) an unsupervised algorithm that utilizes both statistical and linguistic constraints.

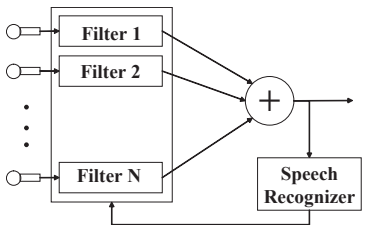

Figure 1. The design of the beamformer. Filter parameters are set to maximize recognizer likelihood.

3.1 Filter Calibration

For the calibration algorithm we assume that the *correct* transcription, *i.e.* the sequence of words in the utterance, is known. Thus the only statistical constraints applied are acoustic. In practice, we utilize only a single *calibration* utterance from the user, for which the transcription is known, to optimize the filters. Future utterances by that speaker are processed with the estimated filters. The implicit assumption in this procedure is that the user does not move too much once their calibration utterance has been recorded. This is not an unrealistic assumption is several situations, such as in automobiles, or users

speaking to their desktop computers.

Since the transcription of the calibration utterance is known, an HMM that is specific to that transcription can now be constructed by concatenating the HMMs for the phonemes that make up the words in the sentence, in appropriate order. We derive the phoneme HMMs from the speech recognizer itself. Filter optimization is then performed using the HMM for the known transcription.

HMM-based speech recognition systems do not operate directly on the speech signal itself. Rather, they operate on a frame-based parameterization of the speech signal. We therefore pose the optimization problem in the context of these frame-based parameterizations. In this paper we assume that each frame of speech is parameterized as a vector of Mel-frequency cepstral coefficients (MFCC); however, the approach taken is equally applicable to any other type of feature vector. Let $\boldsymbol{h}$ represent a vector composed of all filter parameters for all microphones. Let $\boldsymbol{y}_j(\boldsymbol{h})$ represent the signal $y[n]$ in the j^{th} frame of the calibration utterance, expressed as a function of $\boldsymbol{h}$. The MFCC vector the j^{th} frame, $\boldsymbol{z}_j(\boldsymbol{h})$, is computed as

$$\boldsymbol{z}_j(\boldsymbol{h}) = DCT(\log(\boldsymbol{M}\left|DFT(\boldsymbol{y}_j(\boldsymbol{h}))\right|^2)) \quad (2)$$

where $\boldsymbol{M}$ represents the matrix of weighting coefficients of the Mel filters. The entire utterance is parameterized into the sequence of vectors $\boldsymbol{z}_1(\boldsymbol{h}), \boldsymbol{z}_2(\boldsymbol{h}), \ldots, \boldsymbol{z}_T(\boldsymbol{h})$, which we represent as $\boldsymbol{Z}(\boldsymbol{h})$.

The likelihood of any utterance must be computed over all possible state sequences through the HMM for the utterance. In order to simplify the computation, we observe that in an HMM-based system, the likelihood of any data sequence is largely represented by the likelihood of the most likely state sequence through the HMMs. The log-likelihood of $\boldsymbol{Z}(\boldsymbol{h})$ can therefore be approximated as

$$L(\boldsymbol{Z}(\boldsymbol{h})) = \sum_{j=1}^{T} \log(P(\boldsymbol{z}_j(\boldsymbol{h})\left|s_j\right.)) + \log(P(s_1, s_2, \ldots, s_T)) \quad (3)$$

where $s_1, s_2, s_3, \ldots, s_T$ represents the most likely state sequence. $P(\boldsymbol{z}_j(\boldsymbol{h})\left|s_j\right.)$ represents the probability of $\boldsymbol{z}_j(\boldsymbol{h})$ computed on the distribution of the j^{th} state, s_j, in this sequence. $P(s_1, s_2, s_3, \ldots, s_T)$ is determined by the state transition probabilities of the HMM.

Optimization of $L(\boldsymbol{Z}(\boldsymbol{h}))$ requires joint estimation of both $\boldsymbol{h}$ and the most likely state sequence $s_1, s_2, s_3, \ldots, s_T$. This can be performed by iteratively estimating the optimal state sequence for a given $\boldsymbol{h}$ using the Viterbi algorithm, and optimizing $\sum \log(P(\boldsymbol{z}_j(\boldsymbol{h})\left|s_j\right.))$ with respect to $\boldsymbol{h}$ for that state sequence. However, $\sum \log(P(\boldsymbol{z}_j(\boldsymbol{h})\left|s_j\right.))$ cannot be directly optimized and computationally expensive hill-climbing methods must be used to solve for $\boldsymbol{h}$. To reduce computational effort, we model state output distributions as Gaussians, and assume that to maximize $P(\boldsymbol{z}_j(\boldsymbol{h})\left|s_j\right.)$ it is sufficient to minimize the weighted distance $(\boldsymbol{z}_j(\boldsymbol{h}) - \mu_{s_j})^T \boldsymbol{W}(\boldsymbol{z}_j(\boldsymbol{h}) - \mu_{s_j})$ between $\boldsymbol{z}_j(\boldsymbol{h})$ and μ_{s_j}, the mean of the output distribution of s_j. Specifically, we assume that $\boldsymbol{W} = (IDCT)^T(IDCT)$, where $IDCT$ is the inverse discrete cosine transform matrix. This effectively transforms the maximization of $P(\boldsymbol{z}_j(\boldsymbol{h})\left|s_j\right.)$ into the minimization of the Euclidean distance between two log-spectral vectors. Under these assumptions, maximization of $\sum \log(P(\boldsymbol{z}_j(\boldsymbol{h})\left|s_j\right.))$ is equivalent to minimization of the objective function:

$$Q(\boldsymbol{h}) = \sum_{j=1}^{T} \left\| IDCT(\boldsymbol{z}_j(\boldsymbol{h}) - \mu_{s_j}) \right\|^2 \quad (4)$$

$Q(\boldsymbol{h})$ can be optimized with respective to $\boldsymbol{h}$ using hill-climbing methods such as the conjugate gradients method [6].

The entire algorithm for optimizing $\boldsymbol{h}$ from a calibration utterance is thus:

1. Construct an HMM for the transcription of the calibration utterance using HMM components from the speech recognizer.
2. Time-align the signals from the N microphones
3. Initialize $\boldsymbol{h}$ as $h_i[0] = 1/N$; $h_i[k]=0,\ k \neq 0$
4. Process signals using $\boldsymbol{h}$ to generate an output signal
5. Determine optimal state sequence through the utterance HMM using the array output.
6. Use optimal state sequence and (4) to estimate $\boldsymbol{h}$
7. If $Q(\boldsymbol{h})$ has not converged, go to step 4.

Note that time alignment of the signals is not critical. The estimated $\boldsymbol{h}$ is used to process all future utterances during recognition. If the calibration utterance is recorded simultaneously over a close-talking microphone, features derived from this cleaner signal can be used either to determine the optimal state sequence in step 4, or directly in (4) instead of the Gaussian mean vectors.

Filters derived from the calibration utterance are then used on newer utterances by the speaker.

3.2 Unsupervised Filter Estimation

In the unsupervised filter estimation algorithm all constraints are statistical. Thus, the speech recognizer is expected to provide both acoustic and linguistic statistical constraints. Thus, the HMM that is used to measure the likelihood of the output of the array is not merely the HMM for the correct transcription for the recorded utterance, but rather represents the entire expected language. Such an HMM must, of necessity, be very large, and the measurement and maximization of the likelihood of an utterance can be arbitrarily complex. As a result, we resort to an iterative algorithm to effect the optimization.

In each iteration, we process the signal using the current estimate of the filter parameters and perform speech recognition on the output of the array. The recognizer's output is a string of words, that is then assumed to be the true transcription for the utterance. An HMM can be constructed for this transcription and filter parameters can be optimized as they were in Section 3.1. The entire algorithm for estimating filters can be stated as follows:

1. Time-align the signals from the N microphones
2. Initialize $\boldsymbol{h}$ as $h_i[0] = 1/N$; $h_i[k]=0,\ k \neq 0$
3. Process signals using $\boldsymbol{h}$ to generate an output signal.
4. Perform speech recognition on the array output to obtain a word sequence

5. Construct an HMM for the recognized word sequence using HMM components from the speech recognizer.
6. Determine optimal state sequence through the HMM using the current array output.
7. Use optimal state sequence and (4) to estimate $\boldsymbol{h}$
8. If $Q(\boldsymbol{h})$ has not converged, go to step 3.

It is important to note that unlike the calibration algorithm, the unsupervised filter estimation algorithm is applied to every recorded utterance individually. The estimated filters are hence utterance specific, although experimental evidence suggests that they do indeed generalize to other utterances by the same speaker, from the same location.

4. Experimental Evaluation of Beamforming

The proposed beamforming algorithm was evaluated on two microphone array databases recorded at CMU. The first database, TMS8, consists of 140 utterances (10 speakers each with 14 unique utterances), recorded in a noisy speech lab, using an 8 microphone horizontal linear array placed a distance of 1 meter from the speaker. The second corpus, TMS15, was recorded in a reverberant conference room with a talk radio playing, using a 15 element log-linear array with a unit spacing of 4 cm. The distance of the speaker from the array varied from 1m to 3m. The utterances in both sets are comprised of alphanumeric strings and strings of command words. Each microphone array recording also has a close-talking microphone control recording for reference.

Rather than subjective tests, or measurements of SNR, the evaluation metric we use is the automatic speech recognition performance obtained from the output of the array, as compared to that obtained with unprocessed noisy recordings. Superior processing of the array recordings must result in better recognition performance. As a comparator, we also report results obtained from simple delay-and-sum beamforming.

The CMU SPHINX-III speech recognition system with context-dependent continuous HMMs (8 Gaussian/state) trained on clean speech using 7000 utterances from the WSJ0 training set was used in all experiments. In all experiments, all microphone array filters were 50 point FIR filters.

For the calibration experiments one utterance from each speaker was used to estimate filter parameters, and the rest were processed using the estimated parameters. In the unsupervised case, filter parameters were estimated afresh for each utterance.

The recognition results for some of the databases used is shown in Figure 2. We observe that signals processed both by the calibration and unsupervised algorithms result in significant improvements over delay and sum processing. Additionally, we observe that the unsupervised algorithm is often more effective than the calibration algorithm, although the calibration algorithm uses deterministic language constraints whereas the unsupervised algorithm uses only statistical constraints. This is attributable to the fact that the unsupervised beamforming is performed individually on every utterance, and thus computes array parameters that are specific to the spatial and frequency characteristics of both the utterance and the background noise, if any. On the other hand, the calibration algorithm optimizes filters on a calibration utterance and applies it to future utterances from the speaker, and is thus dependent on the similarity of spatial and frequency characteristics of the calibration and test utterances, for effectiveness.

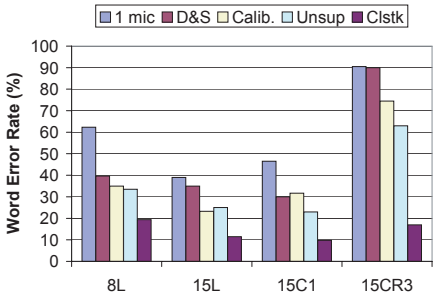

Figure 2. Word error rates for the microphone array databases using a single microphone, signals obtained from delay-and-sum array processing, and the calibration beamforming algorithm, and recordings from a close-talking microphone.

On the whole, the experiments indicate that optimizing the beamformer using the detailed statistical information about speech, that is present in a speech recognizer, can result in highly effective beamforming. The results with the unsupervised algorithm in particular, on the 15CR3 data, which has a background talk radio corrupting the recordings, show that the algorithm is able to lock onto a speech signal in the presence of other structured signal sources.

5. Beamforming for speaker separation

The beamforming algorithms describe in Section 3 can also be extended to the situation where there are multiple speakers speaking simultaneously and the array processing scheme must selectively extract the signal from one of the speakers. In this situation, however, allowance must be made for the fact that the competing signals - signals from other speakers than the one we wish to extract - also match the statistical constraints presented by the recognizer. As a consequence, although one may expect the objective function used for filter parameter estimation - *i.e.* the likelihood of the output of the array as measured by the recognizer - to have multiple local optima, one for each speaker, the iterative algorithms presented in Section 3 are usually unable to arrive at these optima.

It thus becomes necessary to explicitly model the fact that there are multiple speech sources that are simultaneously active. Assuming that the multiple sources are independent of each other, the joint probability for the multiple sources is simply the product of the probability distributions of the individual sources. The probability distribution of any single speech source is modelled by a speech recognizer.

Once again, we note that the speech recognizer in fact represents a combination of two independent sets of statistical constraints: acoustic constraints that are modelled by HMMs, and linguistic constraints that are modelled by a grammar or an N-gram language model. In this paper, in the context of speaker separation, we only address the specific instance where the recognizer provides only statistical acoustic constraints, and all linguistic constraints are deterministic. i.e., we assume that we know the exact word sequence uttered by each of the speakers. We must emphasize that this is not indeed the final goal of our

work - ideally all constraints must be statistical. Nevertheless, the algorithm we present still address two issues: 1. it affirms the hypothesis that a speech recognizer can be used to guide a beamformer, and 2. the filter parameters are likely to generalize to other utterances by the same speakers from the same locations.

From the known word sequences for each speaker we construct an HMM for that word sequence using components from the recognizer. The constructed HMMs represent the probability distribution for the speakers. The joint distribution for all the speakers can be shown to be a cross product of the HMMs for the individual speakers, *i.e.* a factorial HMM, or FHMM.

In an FHMM each state is a composition of one state from the HMMs for each of the speakers, reflecting the fact that the individual speakers may have been in any of their respective states, and the final output is a combination of the output from these states. Figure 3 illustrates the dynamics of an FHMM for two speakers

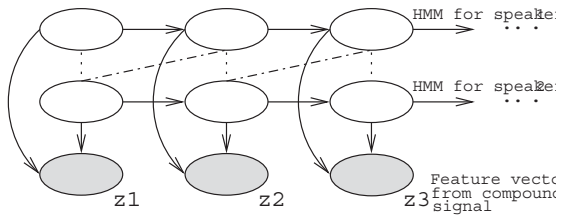

Figure 3. The dynamics of a factorial HMM for two speakers. The signal for each speaker follows the dynamics dictated by the HMM for that speaker, independently of the other speaker. The final output, however, is a combination of the outputs of the two HMMs.

For simplicity, we focus on the two-speaker case. Extension to more speakers is straightforward. Let S_i^k represent the i^{th} state of the HMM for the k^{th} speaker (where k is either 1 or 2). Let $S_{i,j}^{k,l}$ represent the factorial state obtained when the HMM for the k^{th} speaker is in state i and that for the l^{th} speaker is in state j. The output density of $S_{i,j}^{k,l}$ is a function of the output densities of its component states:

$$P(X|S_{i,j}^{k,l}) = f(P(X|S_i^k)(P(X|S_j^l))) \tag{5}$$

The precise nature of the function $f(.)$ is unknown. This is because the relative signal levels of the various speakers is unknown, even at the outset. As the algorithm iteratively improves the beamformer for a specific speaker, the levels of the competing speakers in the output of the array is further reduced to by an unknown degree. At each stage of the algorithm, $f(.)$ must reflect the degree of mixing of the various speakers in the current output of the array, so that the factorial HMM can be appropriately constructed. Clearly, then, it is difficult, if not impossible, to determine $f(.)$ in an unsupervised manner.

We do not attempt to estimate $f(.)$. Instead, we begin with the simplifying assumption that the HMMs for the individual speakers have Gaussian state output distributions (in order for this assumption to be valid, the recognizer used must also model HMM states with Gaussians). We assume that the state output density for any state of the FHMM is also a Gaussian whose mean is a linear combination of the means of the state output densities of the component states.

We define $M_{i,j}^{k,l}$, the mean of the Gaussian state output density of $S_{i,j}^{k,l}$, as:

$$M_{i,j}^{k,l} = A^k M_i^{\ k} + A^l M_j^l \tag{6}$$

where M_i^k represents the D-dimensional mean vector for S_i^k and A^k is a $D \times D$ weighting matrix. We also assume that the covariance matrix for all states of the factorial HMM is the same. The A^k matrices for all the speaker and the global covariance matrix are unknown and must be estimated from the current estimate of the speaker's signal. The estimation is performed using the expectation maximization (EM) algorithm. In the expectation (E) step of the algorithm, the *a posteriori* probabilities of the various factorial states, and thereby the *a posteriori* probabilities of the states of the HMMs for the speakers, are found. The factorial HMM has as many states as the product of the number of states in its component HMMs and direct computation of the E step is prohibitive. We therefore take the variational approach to the estimation. For further details of our implementation of the algorithm, we refer the reader to [7].

Once the A^k matrices and the covariance are estimated, the entire factorial HMM for the mixed signal that is output by the array is constructed. Note now that array filter parameters must now be estimated such that the likelihood of the output on the HMM for the desired speaker is optimized (and not the likelihood on the FHMM). we approximate the optimization as follows: we find the best state sequence through the factorial HMM, to represent the current output of the array. Each state in this sequence represents a compounding of one state from each of the component HMMs. Hence, the best state sequence for the desired speaker can now be extracted from the best state sequence for the factorial HMM. Filter parameters are now optimized using this state sequence using a procedure analogous to that used in Section 3. The overall filter estimation procedure is as follows:

1. Construct an HMM for the transcription of each speaker using HMM components from the speech recognizer.
2. Initialize $\boldsymbol{h}$ as $h_i[0] = 1/N$; $h_i[k]=0,\ k \neq 0$
3. Process signals using $\boldsymbol{h}$ to generate an output signal
4. Learn A^k and state covariance matrices for the FHMM for all the speakers, using the output of the array
5. Determine optimal state sequence through the FHMM using the array output
6. Extract the state sequence for the desired speaker from the optimal state sequence through the FHMM.
7. Use optimal state sequence and (4) to estimate $\boldsymbol{h}$
8. If $Q(\boldsymbol{h})$ has not converged, go to step 3.

6. Experiments on Speaker Separation

Experiments were run to evaluate the proposed speaker separation algorithm. Simulated mixed-speaker recordings were generated using utterances from the test set of the Wall Street Journal(WSJ0) corpus. Room simulation impulse response filters were designed for a room $4m \times 5m \times 3m$ with a reverberation time of 200msec. The microphone array configuration consisted of 8 microphones placed around an imaginary $0.5m \times 0.3m$ flat panel display on one of the walls. Two

speakers were situated in different locations in the room and 8-channel recordings were created for the mixtures.

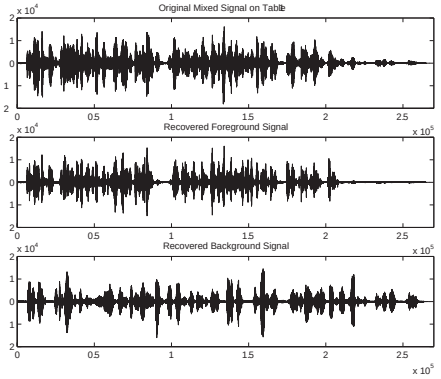

Figure 4. Example waveforms for signals extracted from a mixture of two speakers. The top panel shows a mixed signal, and the bottom two panels show the signals extracted for the two speakers.

Figure 4 shows example waveforms extracted from a mixture of two signals using the proposed algorithm.

Target Speaker	Clean target	Delay & Sum	FHMM Beamformer
Speaker 1	36dB	-11dB	38dB
Speaker 2	24dB	12dB	23dB

Table 1: Speaker to speaker ratio (SSR) in dB for a mixture of two speakers where the first speaker is in the background. The table shows the SSR of the signal derived when filters are optimized with full knowledge of the desired clean signal, when simple delay and sum beamforming is used to extract the desired speaker, and the proposed FHMM based method is used.

Target Speaker	Clean target	Delay & Sum	FHMM Beamformer
Speaker 1	35dB	1dB	21dB
Speaker 2	19dB	1dB	18dB

Table 2: SSR of signals derived using three different processing methods on a mixture of two signals of equal energy.

Tables 1 and 2 show a speaker-to-speaker measure for the extracted signals. In the two-speaker case, this measure is the ratio of the energy of the signal from the desired speaker to that from the competing speaker in the output of the array, expressed in dB. The three columns in the table show what can be achieved by (1) optimizing the filters to minimize the error between the array output and the clean uncorrupted recordings of the desired speaker (these signals are available in our simulations) (2) using delay-and-sum processing with perfect knowledge of the speaker's location, and (3) with the proposed beamformer.

It is evident that the proposed method is highly effective at separating the speakers. In the case where the signal levels of the two speakers are comparable, the algorithms are able to improve the SSRs by 20dB. For the case where the signal levels of the speakers are different, the results are more dramatic - the SSR of the background speaker in table 1 improved by 38dB. The signal separation obtained with the FHMM-based methods is, in most cases, is comparable to that obtained when beamformer parameters are optimized with prefect knowledge of the desired signal. This indicates that replacing the deterministic constraints present in the perfect desired signal with the statistical constraints in the speech recognizer does not result in any degradation of performance.

7. Conclusions and Future Work.

In this paper, we have postulated that the design of a microphone array beamformer for speech signals can be greatly aided by utilizing detailed statistical models for speech. We show that evan an off-the-shelf speech recognition system can provide sufficient statistical constraints on the output of the beamformer to enable us to pick out and enhance a speech signal in a noisy environment. The proposed approach can also be used to separate speech signals from multiple speakers. In the latter case, however we must account for the fact that all the speech signals in a mixture may match the statistical constraints of the speech recognizer by explicitly estimating factorial HMMs for mixtures of speech signals, from the speech recognizer. In this latter case we have only utilized statistical acoustic constraints from the recognizer, and assumed knowledge of the word sequences uttered by the speakers. Under these constraints we are able to separate speakers, even when the signal from one of them one of them is 20dB below that from the other.

The beamforming algorithms presented in this paper have been studied in great detail and have found to be effective on varied data [8]. Nevertheless, they can only be considered preliminary - they are computationally expensive, and in the case of speaker separation make the rather serious assumption that word sequences uttered by the speakers are known. Future work will address the issue of speeding up the computation, as well as that of incorporating statistical language constraints for speaker separation.

Acknowledgements

The authors thank Daniel P. W. Ellis and Richard Stern for many useful suggestions. In particular, we thank Dan Ellis for suggesting the SSR measure to evaluate speaker separation performance.

References

[1] D. H. Johnson and D. E. Dudgeon, *Array Signal Processing: Concepts and Techniques*. New Jersey: Prentice Hall, 1993.

[2] Blitzer, J., Simmer, K.U. and Kammeyer, K.D. (1999), "Theoretical Noise Reduction Limits of the Generalized Sidelobe Canceller (GSC) for Speech Enhancement,", *Proc. IEEE Conf. on Acoustic. Speech and Signal Proc.*, Phoenix AZ.

[3] S. Nordholm, I. Clasesson, and M. Dahl, "Adaptive microphone array employing calibration signals: an analytical evaluation," *IEEE Trans. on Speech and Audio Proc*., vol. 7, pp. 241-252, May 1999.

[4] Aichner, R. Herbordt, W., Buchner, H., and Kellerman, W. (2003), "Least-squared error beamforming using minimum statistics and multichannel frequency domain adaptive filtering,", *Proc. Intl. workshop on Acoustic echo and noise control*, Kyoto Japan.

[5] M. S. Brandstein and H. F. Silverman, "A practical methodology for speech source localization with microphone arrays," *Computer Speech and Language*, vol. 11, pp. 91-126, April 1997.

[6] E. Polak, *Computational methods in Optimization*, New York: Academic Press, 1971.

[7] Reyes, M.J., Raj, B., Ellis, D.P.W.E (2003), "Multi-channel source separation by factorial HMMs," *Proc. IEEE conf. on Acoutics Speech and Signal Proc*., Hong Kong.

[8] Seltzer, M. L. (2003), *Microphone array processing for robust speech recognition*, Ph.D dissertation, Department of Electrical and Computer Engineering, Carnegie Mellon University, July 2003.