

# Markov Chain Monte Carlo Methods for Noise Robust Feature Extraction Using the Autoregressive Model

Robert W. Morris, Jon A. Arrowood, Mark A. Clements

Center for Signal & Image Processing  
Georgia Institute of Technology, Atlanta, USA

{romorris, jon, clements}@ece.gatech.edu

## Abstract

In this paper, Markov Chain Monte Carlo techniques are applied to feature estimation for automatic speech recognition. By using these methods, it is possible to explore new possibilities in leveraging the autoregressive assumption for noise robust feature extraction. Two minimum mean square error estimators are compared that directly estimate the mean of the feature vectors. The first estimator uses the assumption that the speech is an autoregressive signal, while the second makes no assumptions about the speech spectrum. By creating samples from the posterior distribution, these methods also provide an elegant solution to finding feature variances. These variances can be used to create optimal temporal smoothers of the features as well as input for uncertainty observation decoding. Testing on the Aurora2 database shows that autoregressive modeling provides additional information to improve speech recognition performance. In addition, both smoothing and uncertain observation decoding improve performance in this method.

## 1. Introduction

The use of the autoregressive (AR) or linear predictive coding (LPC) model has been pervasive in speech processing areas such as speech coding and enhancement. Although these have been used as features in speech recognition, they were superseded by different cepstral representations such as mel-frequency cepstrum coefficients (MFCC). These were chosen not only because they increased clean performance, but because LPC based features were found to be less robust to noise [1].

It might seem strange to focus on LPC, the very element that made the features less noise robust. However, recent gains in computational resources have made Monte Carlo methods useful for enhancing noise-corrupted speech [2]. In these methods, the speech is assumed to be autoregressive with slowly varying parameters. These algorithms operate in the time domain and their evaluation has focused on waveform enhancement for listening purposes.

There has been recent interest in finding direct statistical estimators of speech features given noise corrupted speech [3]. In this paper, Markov chain Monte Carlo (MCMC) techniques are used to draw samples from the posterior distribution of the feature vectors as shown in Figure 1. This is accomplished in the frequency domain, which simplifies the computation by decorrelating the waveform. With the samples available, it is trivial to get both minimum mean square error (MMSE) estimates and variances of the feature vectors. The automatic generation of the variances is useful, since it can be applied directly to Uncertain Observation (UO) techniques [4, 5, 6].

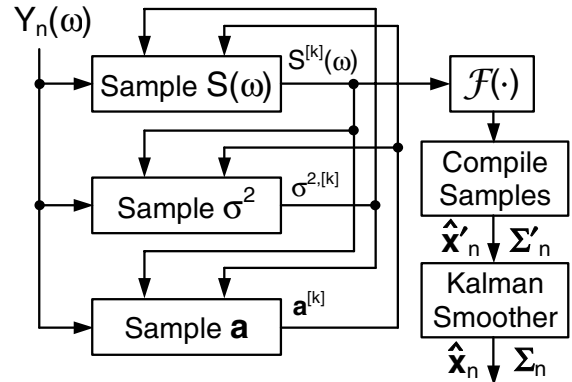


Figure 1: Block diagram of proposed algorithm under the AR assumption.

## 2. Signal Model

We consider two models of the speech signal: autoregressive (AR), and nonparametric (NP). In the AR model, we assume that the  $n$ th observed signal block  $y_n[t] = s_n[t] + v_n[t]$ ,  $t = 1, \dots, T$ , is the sum of an autoregressive speech signal,  $s_n[t]$ , and a random noise signal,  $v_n[t]$ , with power spectrum  $P_v(\omega)$ . The speech signal is expressed by

$$s_n[t] = \sum_{k=1}^p a_n[k]s[t-k] + e_n[t], \quad e_n \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}). \quad (1)$$

Under the NP model, we assume that the speech is a Gaussian process with power spectrum  $P_s(\omega)$ . The prior for this spectrum is given by an inverted gamma distribution

$$P_s(\omega) \sim \mathcal{IG}(\alpha, \beta). \quad (2)$$

The actual quantity of interest is the MFCC feature vector,  $\mathbf{x}_n$ , which is a function of the speech,  $s_n$ . Since direct estimation of AR parameters has been shown to produce erratic results [7], we also want to leverage knowledge about the time evolution of these feature vectors. The feature vectors are modeled by a dynamic linear model

$$\mathbf{x}_n = \mathbf{A}\mathbf{x}_{n-1} + \mathbf{w}_n + \mathbf{F}, \quad \mathbf{w}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}), \quad (3)$$

where the system matrices  $\mathbf{A}$ ,  $\mathbf{F}$ , and  $\mathbf{Q}$  are calculated using Aurora2 training data.

### 3. Algorithm

#### 3.1. Feature Estimation

Our goal is to create an estimator of the true MFCC vector,  $\mathbf{x}_n$ , from the acoustic waveform,  $\mathbf{y}_n$ . The MMSE estimates are  $\mathbb{E}\{\mathbf{x}_n|\mathbf{y}_n, \text{AR}\}$  and  $\mathbb{E}\{\mathbf{x}_n|\mathbf{y}_n, \text{NP}\}$ . Because of the difficulties in finding these directly, we perform Monte Carlo integration on samples from the distributions. Since  $\mathbf{x}_n$  is a function of  $\mathbf{s}_n$ , the problem changes to generating samples of  $\mathbf{s}_n|\mathbf{y}_n, \text{AR}$  and  $\mathbf{s}_n|\mathbf{y}_n, \text{NP}$ . Unfortunately, these distributions are difficult to compute as well. However, the distributions for  $\mathbf{a}|\sigma^2, \mathbf{s}, \mathbf{y}_n$ ,  $\sigma^2|\mathbf{a}, \mathbf{s}, \mathbf{y}_n$ ,  $\mathbf{s}|(P_s, \sigma^2, \mathbf{y}_n)$ , and  $P_s|\mathbf{s}, \mathbf{y}_n$  can be derived analytically. If they are arranged in a Gibbs sampler as shown in Figure 1 and sampled sequentially, these samples converge in distribution to the desired random variables  $\mathbf{a}|\mathbf{y}_n, \sigma^2|\mathbf{y}_n, \mathbf{s}|\mathbf{y}_n$ , and  $P_s|\mathbf{y}_n$ . To simplify the calculation of the distributions, we work with the waveforms entirely in the frequency domain. This yields two samplers for our assumptions: AR and NP. The AR sampler is given by

$$\begin{aligned} \mathbf{a}^{[k]}|\sigma^{2,[k-1]}, \mathbf{s}^{[k-1]}, \mathbf{y}_n &\sim \mathcal{N}\left(\hat{\mathbf{a}}^{[k]}, C_{\mathbf{a}}^{[k]}\right), \\ \sigma^{2,[k]}|\mathbf{a}^{[k]}, \mathbf{s}^{[k-1]}, \mathbf{y}_n &\sim \text{IG}\left(\frac{L}{2} - 1, \frac{L}{2}\hat{\sigma}^{2,[k]}\right), \\ S^{[k]}(\omega)|\mathbf{a}^{[k]}, \sigma^{2,[k]}, Y_n(\omega) &\sim \mathcal{N}\left(\hat{S}^{[k]}(\omega), C_S^{[k]}(\omega)\right), \end{aligned} \quad (4)$$

while the NP sampler is given by

$$\begin{aligned} P_s^{[k]}(\omega)|S^{[k-1]}(\omega), Y_n(\omega) &\sim \text{IG}\left(\alpha + \frac{1}{2}, \beta + \frac{|S^{[k-1]}(\omega)|^2}{2}\right), \\ S^{[k]}(\omega)|P_s^{[k]}(\omega), Y_n(\omega) &\sim \mathcal{N}\left(\hat{S}^{[k]}(\omega), C_S^{[k]}(\omega)\right). \end{aligned} \quad (5)$$

This leaves several elements for definition:

$$\hat{\mathbf{a}}^{[k]} = -\left(\mathbf{R}_s^{[k-1]}\right)^{-1} \mathbf{r}_s^{[k-1]}, \quad (6)$$

$$C_{\mathbf{a}}^{[k]} = \sigma^{2,[k-1]} \left(\mathbf{L}\mathbf{R}_s^{[k-1]}\right)^{-1}, \quad (7)$$

$$\hat{\sigma}^{2,[k]} = r_{s,0}^{[k-1]} - 2\mathbf{a}^{[k],T} \mathbf{r}_s^{[k-1]} + \mathbf{a}^{[k],T} \mathbf{R}_s^{[k-1]} \mathbf{a}^{[k]}, \quad (8)$$

$$r_{s,i}^{[k]} = \frac{1}{L} \sum_{l=0}^{L-1} \left|S^{[k]}(\omega_l)\right|^2 e^{ji\omega_l}, \quad (9)$$

$$\hat{S}^{[k]}(\omega) = \frac{P_s^{[k]}(\omega)}{P_s^{[k]}(\omega) + P_v(\omega)} Y_n(\omega), \quad (10)$$

$$C_S^{[k]}(\omega) = \frac{P_s^{[k]}(\omega)P_v(\omega)}{P_s^{[k]}(\omega) + P_v(\omega)}, \quad (11)$$

where  $\mathbf{R}_s$  and  $\mathbf{r}_s$  are constructed from the sequence  $r_{s,i}$  in the same manner as in the autocorrelation method, and  $L$  is the block length. For comparison, both methods form samples of the speech signal in the same manner using the noise PSD and a sample of the speech PSD,  $P_s^{[k]}(\omega)$ . This is represented in the AR sampler by the LPC coefficients:

$$P_s^{[k]}(\omega) = \frac{\sigma^{2,[k]}}{\left|1 + \sum_{l=1}^p a_l^{[k]} e^{-j\omega l}\right|^2}. \quad (12)$$

For each block, the samplers in Eqns. 4 and 5 create a sequence of  $K$  samples from the posterior distributions. Since the Markov chain requires several iterations to converge in distribution, we only consider samples  $K_b$  through  $K$ , where  $K_b$  is

the number of ‘‘burn-in’’ samples. By using these samples, we are able to get the conditional expectation of any function of the speech signal by transforming the samples and averaging. Regardless of which sampler is used, the initial MFCC parameter estimates are

$$\hat{\mathbf{x}}'_n = \frac{1}{K - K_b} \sum_{k=K_b+1}^K \mathcal{F}\left(\mathbf{S}_n^{[k]}\right), \quad (13)$$

$$\Sigma'_n = \frac{1}{K - K_b} \sum_{k=K_b+1}^K \left(\mathcal{F}\left(\mathbf{S}_n^{[k]}\right) - \hat{\mathbf{x}}'_n\right) \left(\cdot\right)^T, \quad (14)$$

$$[\mathcal{F}(\mathbf{S})]_j = \sum_{k=0}^K d_{j,k} \log \left( \sum_{l=0}^L w_{k,l} |S(\omega_l)| \right), \quad (15)$$

where  $w_{k,l}$  are the filter bank weights and  $d_{j,k}$  are the DCT coefficients for  $\mathcal{F}$ , the spectrum to MFCC transformation.

Finally, one can combine the underlying model in Eqn. 3 with the observation equation  $\hat{\mathbf{x}}'_n = \mathbf{x}_n + \mathbf{v}_n$ ,  $\mathbf{v}_n \sim \mathcal{N}(\mathbf{0}, \Sigma'_n)$ . It is straightforward to use this model with a fixed-lag Kalman smoother to get the final estimates,  $\hat{\mathbf{x}}_n$  and  $\Sigma_n$ .

#### 3.2. Noise Estimation

In both of the methods, an estimate of the noise PSD is required. We collect this information for the experiments using one of two techniques: taking an ideal noise PSD by windowing the nearest  $L_1$  frames of the actual noise signal, or by averaging over the initial  $L_2$  signal frames before speech has begun.

#### 3.3. Uncertain Observations

An interesting consequence of the speech enhancement scheme described above is that it allows for arbitrary PDF descriptions of the unseen clean feature to be generated with relative ease. As several techniques have recently been developed to use PDF descriptions of features for robust recognition in place of the standard points in feature space [4, 5, 6], the Monte Carlo feature enhancement methods of this paper are ideal for this decoding style.

For this paper, we investigated using the Monte Carlo feature enhancement algorithms in Section 3.1 as a front-end to the Uncertain Observation HMM decoding algorithm described in [4]. Instead of calculating the state  $j$  output probability,  $b_j(\mathbf{x})$ , for a frame of speech by finding the probability of a single point,  $\mathbf{x}$ , in space representing that speech frame, this decoding algorithm finds the probability of all possible observations, weighted by their respective likelihoods. Thus, decoding is in general specified as

$$\text{Pr}[\mathbf{y}_n | q_t=j, \mathcal{W}_n] = \int_{-\infty}^{\infty} f_n(\vartheta) b_j(\vartheta) d\vartheta \quad (16)$$

where  $f_n(\mathbf{x})$  is some PDF describing  $\text{Pr}[\mathbf{x}_n | \mathbf{y}_n, \mathcal{W}]$ , the likelihood of unobserved clean speech feature vector  $\mathbf{x}_n$  given noisy observation  $\mathbf{y}_n$  and noise model  $\mathcal{W}_n$ .

For the particular case of a  $K$  mixture Gaussian speech model and a single Gaussian speech observation PDF with mean  $\hat{\mathbf{x}}_n$  and covariance  $\Sigma_{jk}$ , the decoding algorithm simplifies to

$$\sum_{k=1}^K c_{jk} \mathcal{N}(\mu_{jk}, \Sigma_{jk} + \Sigma_n) |_{\hat{\mathbf{x}}_n} \quad (17)$$

It is worth noting that a more complex system can be obtained easily by extending to an  $I$  mixture Gaussian describing

the enhanced speech vector  $\mathbf{x}_n$ :

$$\sum_{i=1}^I c_i \sum_{k=1}^K c_{jk} \mathcal{N}(\mu_{jk}, \Sigma_{jk} + \Sigma_{in}) |_{\hat{\mathbf{x}}_{in}} \quad (18)$$

as the method of Section 3.1 readily extends to providing mixture Gaussian feature estimates.

## 4. Experimental Results

The algorithm parameters were all adjusted for the following experiments. The number of iterations  $K$  and  $K_b$  are set to 1000 and 200 for the NP algorithm, but are raised to 4000 and 500 for the AR algorithm. For the NP algorithm  $\alpha = 0.15$  and  $\beta = 0$ , while in the AR algorithm  $p = 10$ .

### 4.1. Example Iteration

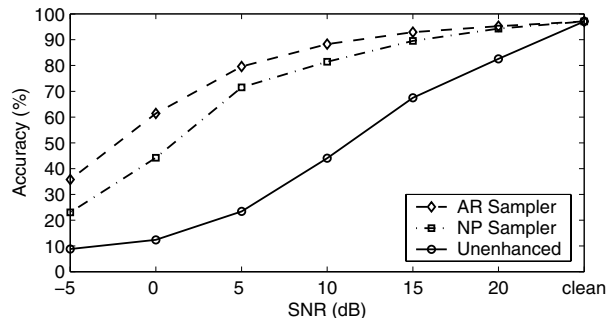
One can gain intuition about the algorithm by viewing results from a single block of noisy speech. In Figure 3, the results of running the Gibbs sampler under the AR condition on a vowel segment corrupted by car noise from Set A, noise condition 3 at -3 dB SNR, are displayed. In Figure 3a, the spectrum of the noisy and clean waveforms are plotted, along with the an estimate of the noise PSD. In the regions of the formant peaks, the signal and noise energy are comparable, while the region between 500 and 1500 Hz is completely dominated by noise.

The Gibbs sampler creates a sequence of samples of the LPC spectrum, given by  $\mathbf{a}^{[k]}$  and  $\sigma^{2,[k]}$ . In Figure 3b, the spectrum generated by ten samples of these parameters are plotted along with the LPC spectra calculated from the clean and noisy waveforms. These samples are highly variable, but tend to be clustered around the clean spectrum. In Figure 3c, the mean and variance estimates of the speech log spectrum are displayed. The distribution is represented by two gray bands representing the regions within one and two standard deviations from the mean value. This effectively finds the shape of the spectral envelope, without attempting to estimate the fine structure. Of course, we are interested in features for speech recognition, so the filter bank coefficients are included in Figure 3d, where each filter bank output is plotted at the location of its center frequency. Again, the estimated parameters are represented using the calculated means and variances. One can see that the shape of the filter bank outputs is well estimated by this method.

### 4.2. Speech Recognition Experiments

The enhancement techniques of Section 3.1 were also tested using the Aurora2 digit database. Experiments were designed to compare the performance between the AR-constrained system and the nonparametric system and to show further improved recognition performance when feature variance estimates are used in the Uncertain Observation algorithm. Models were trained using the Aurora2 clean-training scenario. For all experiments, recognition was performed using only the static MFCC coefficients and log energy, giving 13 dimensional feature vectors instead of the standard 39. This was because the increased feature variance from the enhancement algorithm caused the derivative features to hurt performance.

To get an upper bound on performance, experiments were first run using an ideal noise estimator. A smoothed spectral version of the actual noise signal was used as the noise estimate in the enhancement algorithm. Under these ideal conditions, the enhancement algorithm that assumes an underlying autoregressive model outperforms the unconstrained non-parametric



**Figure 2:** Comparison of AR and NP Gibbs samplers, averaged over Sets A and B, across SNRs.

enhancement algorithm. The performance of the AR and NP algorithms can be compared with the baseline Aurora2 recognition system from results in Table 1. Results are given for each algorithm both before and after the final Kalman smoothing filter.

Enhancement method	Set A Acc (%)	Set B Acc (%)	Set C Acc (%)	Overall Acc (%)
Unenhanced	45.82	50.09	40.88	46.54
AR Sampler	77.40	79.88	74.97	77.90
AR Sampler*	75.69	78.81	71.46	76.09
NP Sampler	71.46	71.74	64.20	70.12
NP Sampler*	61.52	62.19	61.69	61.82

**Table 1:** Results of the AR and NP enhancement algorithms using an ideal noise estimator. The \* denotes results prior to the final Kalman smoothing filter.

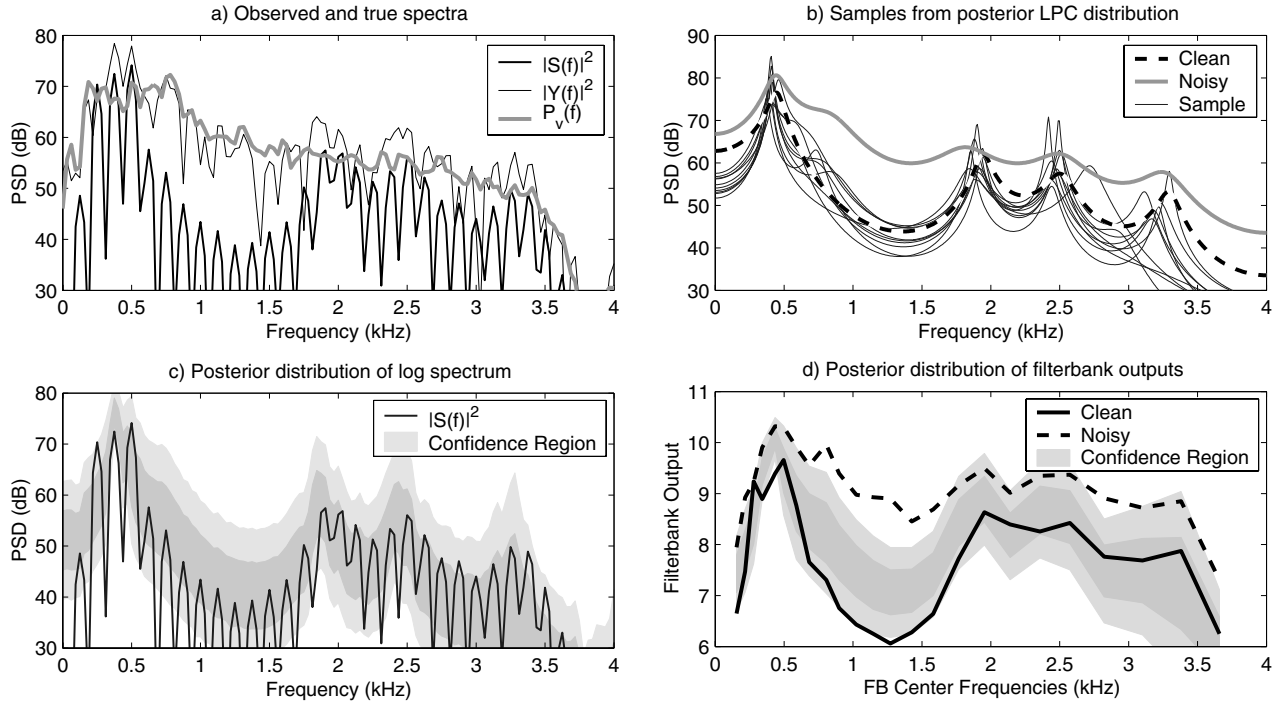
As expected, the Set A and Set B results in Table 1 are similar, since no use is made of stereo training data, and hence, no assumption is made about the noise. Since neither method makes allowances for convolutional distortion as found in Set C, the performance is lower than that of Sets A and B. In Figure 2, the results from the AR and NP algorithms are plotted across the range of Aurora2 SNRs. These results are averaged over Sets A and B only, as Set C is outside of the types of distortion for which this algorithm is designed.

For a more realistic example, the noise PSD was estimated from nonspeech frames prior to the beginning of the utterance. The results for this method, shown in Table 2 for the AR Gibbs sampler, illustrate the necessity for an accurate noise PSD estimate. Future work toward a practical implementation would necessitate a more advanced noise estimation algorithm, where  $P_v(\omega)$  would be tracked in time and included in the sampler iterations.

Enhancement method	Set A Acc (%)	Set B Acc (%)	Set C Acc (%)	Overall Acc (%)
Unenhanced	45.82	50.09	40.88	46.54
AR Sampler	66.53	65.06	59.22	64.48

**Table 2:** AR enhancement results using nonspeech frames for noise PSD estimation.

The final set of experiments were performed again using the



**Figure 3:** Example processing using the AR assumption on a block of noisy speech from Set A, noise condition 3, block SNR = -3 dB.

ideal noise PSD estimate and the AR Gibbs sampler, but this time the second order statistics of the clean feature estimates are used in the Uncertain Observation algorithm described in Section 3.3. The results, shown in Table 3, are averaged between Sets A and B only, and again use only static MFCC features. The results are compared with both baseline recognition using only the AR sampler means and the ideal case that the true feature variance is known (i.e., the exact value of  $(\mathbf{y}_n - \mathbf{x}_n)^2$  is used as the diagonal of matrix  $\Sigma_n$ ).

Decoding Method	Accuracy (%)
Standard Decoding	79.3
UO using AR sampler variances	80.8
UO using ideal variances	87.4

**Table 3:** Comparison between standard HMM decoding, UO decoding using variances from the AR sampler, and UO using ideal variances.

## 5. Conclusion

We have presented a new algorithm for estimating feature vectors for speech recognition based on MMSE criterion and knowledge of the power spectrum density of the corrupting noise by using Markov Chain Monte Carlo methods. We found that this method was effective when an accurate estimate of the noise PSD was available, improving recognition of noisy speech. A side effect of this style of enhancement algorithm is the ability to easily estimate feature variance, which can be exploited during decoding to further improve recognition. Also worth noting is an improvement due to a Kalman smoothing filter applied to enhanced feature vectors prior to recognition. We also found that using the underlying assumption that the speech spectrum

is autoregressive allowed for improved enhancement, even at low signal to noise ratios.

## 6. References

- [1] C. R. Jankowski Jr., H.-D. H. Vo, and R. P. Lippmann, "A comparison of signal processing front ends for automatic word recognition," *IEEE Trans. Speech and Audio Proc.*, vol. 3, pp. 286–293, 1995.
- [2] W. Fong, S. J. Godsill, A. Doucet, and M. West, "Monte Carlo smoothing with application to audio signal enhancement," *IEEE Trans. Signal Proc.*, vol. 50, pp. 438–449, 2002.
- [3] L. Deng, J. Droppo, and A. Acero, "Log-domain speech feature enhancement using sequential MAP noise estimation and a phase-sensitive model of the acoustic environment," in *Proc. ICSLP*, 2002, pp. 1813–1816.
- [4] J. A. Arrowood and M. A. Clements, "Using observation uncertainty in HMM decoding," in *Proc. ICSLP*, 2002, pp. 1561–1564.
- [5] N. Yoma and M. Villar, "Speaker verification in noise using a stochastic version of the weighted Viterbi algorithm," *IEEE Trans. Speech and Audio Proc.*, vol. 10, pp. 158–166, 2002.
- [6] L. Deng, J. Droppo, and A. Acero, "Exploiting variances in robust feature extraction based on a parametric model of speech distortion," in *Proc. ICSLP*, 2002, pp. 2449–2452.
- [7] R. W. Morris, M. A. Clements, and J. S. Collura, "Autoregressive parameter estimation of speech in noise," in *IEEE Speech Coding Workshop*, 2002, pp. 181–183.