NOISE ROBUST AURORA-2 SPEECH RECOGNITION EMPLOYING A CODEBOOK-CONSTRAINED KALMAN FILTER PREPROCESSOR

Venkatesh Krishnan, Sabato M. Siniscalchi, David V. Anderson and Mark A. Clements

Center for Signal and Image Processing, School of Electrical and Computer Engineering,
Georgia Institute of Technology, Atlanta, Georgia 30332, USA

ABSTRACT

In this paper, a speech signal estimation framework involving Kalman filters for use as a front-end to the Aurora-2 speech recognition task is presented. Kalman-filter based speech estimation algorithms assume autoregressive (AR) models for the speech and the noise signals. In this paper, the parameters of the AR models are estimated using a expectation–maximization approach. The key to the success of the proposed algorithm is the constraint on the AR model parameters corresponding to the speech signal to belong to a codebook trained on AR parameters obtained from clean speech signals. Aurora-2 noise-robust speech recognition experiments are performed to demonstrate the success of the codebook-constrained Kalman filter in improving speech recognition accuracy in noisy environments. Results with both *clean* and *multi-conditional* training are provided to show the improvements in the recognition accuracy compared to the base-line system where no pre-processing is employed.

1. INTRODUCTION

Recently the design of automatic speech recognition (ASR) systems for use in personal and mobile electronic devices has been seeing a tremendous growth. The design of robust ASR systems for use in mobile environments poses several research challenges. Firstly, these systems must perform without degradation in a variety of environmental conditions, where the input speech is corrupted by background noise. Secondly, the implementation of these systems is constrained by the limited resources available in wireless devices. In a distributed speech recognition (DSR) environment, features are extracted from the speech signal at the remote location and the recognition is performed in a centralized server.

One solutions to the problem of designing robust ASR systems is to employ noise suppression algorithms prior to the feature extraction by the DSR system. The use of Kalman filters (KF) for estimation of clean speech from noisy measurements has been widely explored [1] [2] [3]. Typically, the KF formulation for speech signal estimation assumes that the speech signal can be modeled as a p^{th} order autoregressive (AR) process. To accommodate non-white spectral characteristics of the noise corrupting the speech, the noise signal is also modeled as a q^{th} order AR process. The state of the KF is usually defined to include p consecutive speech and q consecutive noise samples. The KF then provides a minimum mean square error (MMSE) estimate of the KF state at a time instance t, given the noisy measurement and the AR model for the time evolution of the state of the Kalman filter. The estimate of the clean speech signal can be derived from the estimated KF state. In this paper, we present a Kalman filter based signal estimator and use it as a preprocessor to the Aurora speech recognition system to demonstrate its effectiveness.

The performance of such a KF system largely depends on the reliability of the estimates of the AR model parameters. Since the clean speech signal and the noise are unknown, standard procedures for AR model parameter estimation, such as the autocorrelation method, can not be employed. In the proposed KF preprocessor, the AR model parameters for the clean speech and the noise signals are obtained from codebooks, $\mathcal{C}_s$ and $\mathcal{C}_v$, containing suitably designed prototype AR parameters of the speech and noise signals respectively. These codebooks are trained using the standard k-means clustering of the AR parameters obtained from a database of clean speech and speech-free noise signals. During the operation of the KF, the appropriate AR parameters are selected from $\mathcal{C}_s$ and $\mathcal{C}_v$ every frame (10-40 msec duration) using an Expectation Maximization (EM) [4] algorithm.

The mathematical formulation of the proposed codebook-constrained KF (CCKF) preprocessor is presented in Section 2. A brief description of the Aurora-2 system for speech recognition with a simple back-end and the proposed CCKF in the front-end is provided in Section 3. The conclusions are given in Section 4.

2. CODEBOOK CONSTRAINED KALMAN FILTER PREPROCESSOR

In this section, the mathematical formulation of the proposed speech signal estimator that uses a codebook constrained Kalman filter is presented. Let the noisy speech measurement at the time t be $y[t]$.

$$y[t] = s[t] + v[t] \tag{1}$$

In this paper, it is assumed that the speech signal $s[t]$ and the noise signal $v[t]$ can be modeled as Gaussian AR random processes [5]. They may be expressed as

$$s[t] = \sum_{j=1}^{p} \alpha_j s[t-j] + e[t], \quad v[t] = \sum_{j=1}^{q} \beta_j v[t-j] + u[t] \tag{2}$$

where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_p]$ are the p AR model parameters for the speech signal, and $\boldsymbol{\beta} = [\beta_1, \beta_2, ...\beta_q]$ are the q AR model parameters for the noise signal, $v[t]$. The signals $e[t]$ and $u[t]$ are independent Gaussian white noise signals with second order moments σ_e^2 and σ_u^2, respectively. Equation (2) can be written in vector-matrix notation as

$$\mathbf{x}[t] = \Phi \mathbf{x}[t-1] + G[t], \tag{3}$$

where

$$\begin{aligned} \mathbf{x}[t] =& [s[t-p+1], \ldots, s[t], v[t-q+1], \ldots, v[t]],^T \\ G[t] =& [0, \ldots, e[t], 0, \ldots, u[t]],^T \end{aligned} \tag{4}$$

and

$$\Phi = \begin{bmatrix} \Phi_s & \mathbf{0} \\ \mathbf{0} & \Phi_v \end{bmatrix} \text{where}$$

$$\Phi_s = \begin{bmatrix} 0 & 1 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_p & \alpha_{p-1} & \dots & \alpha_1 \end{bmatrix} \text{ and } \Phi_v = \begin{bmatrix} 0 & 1 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \beta_q & \beta_{q-1} & \dots & \beta_1 \end{bmatrix}. \quad (5)$$

Let the autocorrelation matrix of $G[t]$ be $\mathbf{\Sigma} = E\{G[t]G[t]^T\}$. $\mathbf{\Sigma}$ is a $(p+q)\times(p+q)$ matrix with σ_e^2, σ_u^2 in the (p,p) and $(p+q,p+q)$ locations respectively and 0 elsewhere. Also in (3), The observation, $y[t]$, is related to $\mathbf{x}[t]$ by

$$y[t] = \mathbf{M}\mathbf{x}[t], \quad (6)$$

where $\mathbf{M}$ is a $1 \times (p+q)$ vector with the 1 in the p^{th} and $p+q^{\text{th}}$ position and 0 elsewhere. The speech signal amplitude at time t can be derived from $\mathbf{x}[t]$ using $s[t] = \mathbf{M_1}\mathbf{x}[t]$, where $\mathbf{M_1}$ is a $1\times(p+q)$ vector with the 1 in the p^{th} position and 0 elsewhere.

2.1. The Kalman filter

If the AR model parameters, $\boldsymbol{\alpha}, \boldsymbol{\beta}$, and $\boldsymbol{\sigma}$ are known *a priori*, then a Kalman filter, whose state vector at t is $\mathbf{x}[t]$, can be employed to estimate the clean speech signal. The AR model parameters can be derived if the clean speech signal and the residual noise signals are known. Since in a practical system these signals are unknown, an algorithm for the ML estimation of these AR parameters is described in Section 2.2. In this section, we provide the Kalman filtering equations for obtaining the sample-by-sample minimum mean-squared error (MMSE) estimate of $s[t]$, assuming that the ML estimates of these AR parameters are available.

Let $\widetilde{\mathbf{x}}[t|\tau]$ be the best estimate of the state of the system $\mathbf{x}[t]$, using all available information till the time instance $\tau \leq t$. Let

$$\mathbf{P}[t|\tau] = E\left\{(\mathbf{x}[t] - \widetilde{\mathbf{x}}[t|\tau])(\mathbf{x}[t] - \widetilde{\mathbf{x}}[t|\tau])^T\right\}. \quad (7)$$

If $\widetilde{\Phi}$ is a matrix similar to (5), but constructed using the maximum likelihood (ML) estimates of the AR model parameters, then for a time-frame $t = t_1, t_1+1, \dots, t_2$, the Kalman filtering [2] equations are given by

$$\widetilde{\mathbf{x}}[t|t] = \Lambda[t]\widetilde{\mathbf{x}}[t-1|t-1] + \Delta[t]y[t], \quad (8)$$

$$\text{where} \quad \Delta[t] = \widetilde{\Phi}\mathbf{P}[t|t]\mathbf{M}^T\left[\mathbf{M}\mathbf{P}[t|t]\mathbf{M}^T\right]^{-1}, \quad (9)$$

$$\text{and} \quad \Lambda[t] = \widetilde{\Phi} - \Delta[t]\mathbf{M}. \quad (10)$$

$$\mathbf{P}[t+1|t+1] = \Lambda[t]\mathbf{P}[t|t]\widetilde{\Phi} + \widetilde{\mathbf{\Sigma}}. \quad (11)$$

$\Delta[t]$ is defined as the Kalman gain, and $\widetilde{\mathbf{\Sigma}}$ is the estimate of $\mathbf{\Sigma}$. The MMSE estimate of the clean speech at t is given by

$$\widetilde{s}[t] = \mathbf{M_1}\widetilde{\mathbf{x}}[t|t]. \quad (12)$$

2.2. Codebook-constrained ML estimation of AR parameters

The performance of the CCKF largely depends on the reliability of the estimates of the AR model parameters of the clean speech and the residual noise signals, but in a practical system, the true AR model parameters for use in the CCKF are unavailable. In this section, an iterative EM algorithm for obtaining the ML estimate of the AR model parameters from the noisy speech input to the CCKF for the time-frame $t_1 \leq t \leq t_2$ is presented. The CCKF framework with the EM algorithm is shown in Fig. 1. It may be noted that while the Kalman filter operates on a sample-by-sample basis, the AR model parameters used by the CCKF may be updated on a frame-by-frame basis since these parameters tend to be stationary over short periods of time (10–40 msec).

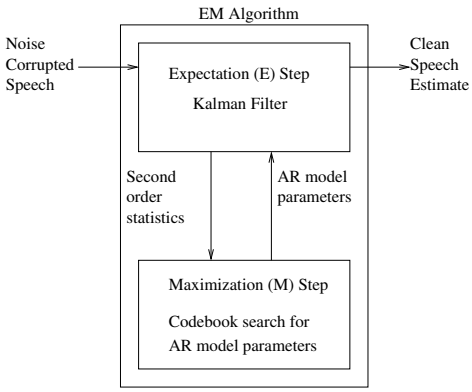

Fig. 1: Codebook-constrained Kalman filtering based speech estimation framework.

Let us define the frame $\mathbf{Y} \doteq \{y[t_1], y[t_1+1], \dots, y[t_2-1], y[t_2]\}$, $\mathbf{s} \doteq \{s[t_1], s[t_1+1], \dots, s[t_2-1], s[t_2]\}$, $\mathbf{V} \doteq \{v[t_1], v[t_1+1], \dots, v[t_2-1], v[t_2]\}$ for the time period $t_1 \leq t \leq t_2$, and the set of AR parameters for this frame be denoted $\Theta = \{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\sigma}\}$. If $f(\mathbf{Y};\Theta)$ is the PDF of $\mathbf{Y}$ parameterized on Θ, then the ML estimate of Θ is given by

$$\Theta_{ML} = \underset{\Theta}{argmax} \log[f(\mathbf{Y};\Theta)]. \quad (13)$$

Defining the *complete-data log-likelihood* function [4] as $\log[f(\mathbf{s},\mathbf{V};\Theta)]$, the i^{th} iteration of the EM algorithm can be described in the following two steps:

• ***The E step*** involves the evaluation of the cost function

$$Q(\Theta, \widetilde{\Theta}^{(i)}) = E\left[\log f(\mathbf{s},\mathbf{V};\Theta)|\widetilde{\Theta}^{(i)}, \mathbf{Y}\right]. \quad (14)$$

Since the PDF $f(\mathbf{s},\mathbf{V};\Theta)$ represents an AR Gaussian density, (14) can be expanded as

$$Q(\Theta, \widetilde{\Theta}^{(i)}) = -\frac{t_2 - t_1}{2}\log\frac{\sigma_s^2}{\sigma_u^2} - \sum_{t_1}^{t_2}\left[\frac{E\left\{(s[t] - \sum_{j=1}^{p}\alpha_j s[t-j])^2\right\}}{2\sigma_s^2} + \frac{E\left\{(v[t] - \sum_{j=1}^{q}\beta_j v[t-j])^2\right\}}{2\sigma_u^2}\right] \quad (15)$$

The second order statistics in (15) are obtained from the (7) and (8) [2]. The $\widetilde{\Phi}$ and $\widetilde{\Sigma}$ used by the Kalman filter (9) - (11) to evaluate (7) and (8) is constructed using $\widetilde{\Theta}^{(i)}$.

• ***The M step*** determines the set of AR parameters that maximizes the likelihood function

$$\widetilde{\Theta}^{(i)} = \underset{\Theta}{argmax}\, Q(\Theta, \widetilde{\Theta}^{(i)}). \quad (16)$$

The optimal AR parameters $\boldsymbol{\alpha}$ corresponding to the clean speech are constrained to belong to a suitably designed codebook $\mathcal{C}_s$. This

	Set A					Set B					Set C			Avg
	Subway	Babble	Car	Exhib.	Avg	Restau	Street	Airport	Station	Avg	SubM	StrM	Avg	
Clean	98.83	98.94	99.05	98.92	98.94	98.83	98.94	99.05	98.92	98.94	99.05	99.00	99.03	98.95
20 dB	97.24	97.82	98.21	96.88	97.54	97.14	97.52	97.67	97.69	97.51	96.93	97.76	97.35	97.49
15 dB	95.00	95.50	97.08	94.88	95.62	94.78	96.07	96.12	95.09	95.52	94.96	95.68	95.32	95.52
10 dB	89.90	90.27	92.63	88.92	90.43	89.1	90.99	90.84	91.82	90.69	88.70	89.75	89.23	90.29
5 dB	78.45	77.24	77.84	75.59	77.28	73.26	78.51	77.81	78.00	76.9	76.33	75.18	75.76	76.82
0 dB	50.38	47.34	42.56	49.55	47.46	46.70	49.12	51.18	44.96	47.99	45.75	45.16	45.46	47.27
-5 dB	19.96	16.75	13.45	20.89	17.76	17.29	19.07	18.16	15.92	17.61	16.58	18.74	17.66	17.68
Avg	82.19	81.63	81.66	81.16	81.66	80.20	82.44	82.72	81.51	81.72	80.53	80.71	80.62	81.48

(a)

	Set A					Set B					Set C			Avg
	Subway	Babble	Car	Exhib.	Avg	Restau	Street	Airport	Station	Avg	SubM	StrM	Avg	
Clean	-5.41	-9.28	0.00	-45.95	-15.16	-5.41	-9.28	0.00	-45.95	-15.16	-14.46	-9.89	-12.17	-14.56
20 dB	15.08	76.96	38.70	17.89	37.16	70.99	40.10	76.82	55.66	60.89	53.70	54.00	53.85	49.99
15 dB	40.97	83.82	74.50	48.65	61.98	79.51	66.84	85.17	73.83	76.34	63.92	60.40	62.16	67.76
10 dB	58.72	81.43	79.79	60.01	69.99	77.34	73.46	81.94	81.74	78.62	59.73	58.95	59.34	71.31
5 dB	59.08	70.48	68.00	59.93	64.37	63.47	66.27	70.57	70.68	67.75	52.06	49.79	50.92	63.03
0 dB	36.02	44.26	35.67	41.17	39.28	42.61	38.44	45.45	39.18	41.42	28.12	28.18	28.15	37.91
-5 dB	10.42	16.65	7.11	15.07	12.31	16.50	11.44	13.60	10.42	12.99	4.23	8.50	6.36	11.39
Avg	41.97	71.39	59.33	45.53	54.56	66.79	57.02	71.99	64.22	65.00	51.50	50.27	50.88	58.00

(b)

Table 1: (a)Aurora2 word recognition accuracy with CCKF front-end when trained on *clean* data. (b) Relative improvement over baseline

	Set A					Set B					Set C			Avg
	Subway	Babble	Car	Exhib.	Avg	Restau	Street	Airport	Station	Avg	SubM	StrM	Avg	
Clean	98.34	98.64	98.51	98.61	98.53	98.34	98.64	98.51	98.61	98.53	98.53	98.94	98.74	98.57
20 dB	97.88	98.28	98.51	98.40	98.27	97.30	97.70	98.33	98.12	97.86	97.45	97.88	97.67	97.99
15 dB	96.32	97.13	98.15	97.47	97.27	95.95	97.22	97.20	97.35	96.93	96.25	96.92	96.59	97.00
10 dB	94.04	94.92	96.42	95.68	95.27	93.21	94.86	95.71	95.12	94.73	93.55	95.10	94.33	94.86
5 dB	87.66	88.15	88.85	89.02	88.42	86.37	88.51	89.35	88.24	88.12	86.86	86.79	86.83	87.98
0 dB	66.35	65.63	66.27	70.56	67.20	63.43	65.96	72.83	66.49	67.18	67.06	67.08	67.07	67.17
-5 dB	33.47	26.57	30.60	37.06	31.93	25.73	32.50	38.09	31.35	31.92	31.87	33.4	32.64	32.06
Avg	88.45	88.82	89.64	90.23	89.28	87.25	88.85	90.68	89.06	88.96	88.23	88.75	88.49	89.00

(a)

	Set A					Set B					Set C			Avg
	Subway	Babble	Car	Exhib.	Avg	Restau	Street	Airport	Station	Avg	SubM	StrM	Avg	
Clean	-17.73	5.26	8.39	6.84	0.69	-15.60	13.53	9.79	3.42	2.79	9.42	22.56	15.99	4.59
20 dB	16.97	31.55	28.98	35.57	28.27	22.03	7.53	30.13	36.75	24.11	12.45	16.12	14.28	23.81
15 dB	-1.79	11.67	32.61	24.68	16.79	10.42	30.03	25.00	37.00	25.61	16.57	43.90	30.24	23.01
10 db	-4.27	-7.73	19.11	26.15	8.31	13.31	17.32	36.57	27.80	23.75	0.32	29.77	15.05	15.83
5 dB	-7.10	5.51	11.55	13.15	5.78	12.84	16.02	18.04	24.80	17.93	28.89	23.86	26.38	14.76
0 db	-2.05	9.26	27.86	18.57	13.41	10.10	7.90	19.83	20.32	14.54	40.12	26.58	33.35	17.85
-5 db	9.53	-0.91	13.37	17.72	9.93	0.22	7.05	11.67	12.27	7.80	16.92	11.77	14.35	9.96
Avg	0.35	10.05	24.02	23.62	14.51	13.74	15.76	25.91	29.33	21.19	19.67	28.05	23.86	19.05

(b)

Table 2: (a)Aurora2 word recognition accuracy with CCKF front-end for *multi-conditional* training (b) Relative improvement over baseline

codebook is designed by the standard K-means clustering of AR parameters derived from a database of clean speech signals. The other AR parameters, $\boldsymbol{\beta}_k$'s and $\boldsymbol{\sigma}$, are estimated by an unconstrained maximization of the likelihood function [2]. Although $\boldsymbol{\alpha}$ can also be estimated as described in [2], we observed that the perceptual quality of the estimated clean speech is remarkably better when it is constrained to belong to the codebook $\mathcal{C}_s$.

3. AURORA2 NOISY SPEECH RECOGNITION

Given the need to have a common platform where researchers could test their noise pre-processing and ASR algorithms, and compare their results fairly, the Aurora DSR Working Group defined a set of connected digit string recognition experiments called the Aurora-2 task [6].

The Aurora-2 task defines two training modes: (a) *clean training mode* in which the recognition engine is trained on clean data alone and (b) *multi-conditional training* where training is done using both clean and noisy data. Three testing sets are provided for the evaluation of the Aurora-2 task. Each set has 4 subsets of 1001 utterances obtained from the TIDigits test database. The first testing set, *set A* contains four sets of 1001 sentences, corrupted by subway, babble, car, and exhibition hall noises, respectively, at different SNR levels. The noise types included in this set are the same as those in the multi-conditional training. The second set, *set B* contains 4 sets of 1001 sentences each, corrupted by restaurant, street, airport, and train station noises at different SNR levels. These noise types are different from the ones used in the *multi-conditional* training. The test *set C* contains 2 sets of 1001 sentences, corrupted by subway, and street and airport noises. The data *set C* was filtered with the MIRS filter [7] before the addition of noise in order to evaluate the robustness of the algorithm under convolutional distortion mismatch.

3.1. Front-end noise suppression using CCKF

The performance of the CCKF described in Section 2 when incorporated in the front-end of the Aurora-2 task is evaluated in this section. For the CCKF noise pre-processor, 10^{th} order AR models were used

for the speech and the noise signals. It was found that a 10^{th} order model was sufficient for most noise types and increasing the model order did not yield any improvements. The AR model parameters corresponding to the speech signal were constrained to belong to a codebook containing 4096 prototype 10^{th} order AR parameter vectors. This codebook was trained using 100000 AR parameter vectors derived from clean speech files from the TIMIT training database using the k-means algorithm [8]. Three iterations of the EM algorithm as described in Sections 2.2 were performed. The estimate of the clean speech signal obtained from the Kalman filter (during the E step of the final iteration) is used as the input to the Aurora-2 system.

Aurora-2 speech database along with the ETSI Mel-cepstrum DSR (WI007) standard front-end version 2.0 were used for evaluating the ASR performance of the CCKF. The standard front-end extracts a 39-dimensional feature vector from the estimate of the speech signal. The feature vector consists of the 12 MFCCs (MFCC of order 0 is not included)logarithmic frame energy, and their first and second order derivatives. Once the features are extracted, cepstral mean and variance normalization [9] across an utterance are performed. If $\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_J$ are the cepstral feature vectors obtained corresponding to an utterance, then the normalized features are given by

$$\widehat{\mathbf{F}}_j = \sigma_F^{-1/2} \left(\mathbf{F}_j - \mu_F \right), \tag{17}$$

where μ_F is the mean vector obtained by averaging the feature vectors and σ_F is a diagonal variance matrix containing the variances of each component of the feature vector over the utterance. Mean normalization accounts for channel distortion by forcing the features to have zero mean over an utterance. Variance normalization tries to smooth the difference between noisy and clean utterance by making the variance unitary over an utterance. The back-end consists in a whole word left-to-right continuous density hidden Markov model (CDHMM) where a single word is represented by 18 states, and each state has three diagonal-covariance Gaussian mixtures. The search engine of HTK 3.0 toolkit was used to perform the experiments.

With the Aurora-2 task setup described above, two sets of experiments were performed. In the first experiment, the training was performed using features extracted from the *clean training* database. The testing speech files belonging to *set A*, *set B*, and *set C* were enhanced using the CCKF. In Table 1, the word recognition accuracy (in percentage) are shown for different noise types and levels when the CCKF is used in the front-end over the baseline case where no enhancement is used in the front end.

In the second set of experiments, the recognition accuracy was evaluated when the back-end was trained using multi-conditional training data. In this case, the training data was also enhanced with the proposed CCKF. The testing speech files belonging to *set A*, *set B*, and *set C* were enhanced using the CCKF. Three iterations of the proposed CCKF algorithm were performed and the speech and the noise processes were assumed be 10^{th} order AR processes. In Table 2, the word recognition accuracy (in percentage) for different noise types and levels when the CCKF is used in the front-end over the baseline case where no enhancement is used in the front end are shown.

The performance of the proposed CCKF in improving speech recognition accuracy can be compared easily with other noise preprocessing algorithms using the Aurora experiments. Typically, we have observed that the performance of the proposed CCKF is either better or comparable to several state-of-the art noise preprocessing systems used in the Aurora experiments with a simple back-end. In [10], an Adaptive Sub-Band Spectral Subtraction (ASBSS) based front-end processing is proposed. Additionally, a noise pre-processor that combines the ASBSS with a Kalman filter is also presented. For both cases, the proposed CCKF achieves more than 10% improvement over the two methods for both clean and multi-conditional training. The performance improvement with the proposed CCKF may be attributed to the fact that constraining the AR parameters to a codebook trained on clean speech enables the noise pre-processor to select speech-like spectral parameters. In general, the word recognition accuracy was found to be better in the multi-conditional training case than the clean training case due to match in the training and testing condition. The improvement in recognition accuracy was found to be significant compared to the baseline for almost all noise conditions except babble and subway. It may be noted that babble noise is speech-like and, therefore, constraining the AR parameters to a codebook trained on clean speech does not improve the estimate of clean speech. In most cases, the improvements are better for low SNRs compared to higher SNRs.

4. CONCLUSIONS

In this paper, a codebook constrained Kalman filter based speech estimation framework was presented for use as an enhancement front-end in an automatic speech recognition system. The estimate of autoregressive model parameters required by the Kalman filter were constrained to belong to a codebook trained on such parameters obtained from clean speech. Simulation results from the Aurora-2 experiments were provided to demonstrate the improvements in the speech recognition accuracy compared to the base-line system for both *clean* and *multi-conditional* training.

5. REFERENCES

[1] K. K. Paliwal and A. Basu, "A speech enhancement method based on Kalman filtering," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1987, pp. 177–180.

[2] J. D. Gibson, B. Koo, and S. D. Gray, "Filtering of colored noise for speech enhancement and coding," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 39, pp. 1732–1742, 1991.

[3] S. Gannot, D. Burshtein, and E. Weinstein, "Iterative and sequential Kalman filter-based speech enhancement algorithms," *IEEE Transactions on Signal Processing*, vol. 6, no. 4, pp. 373–385, 1998.

[4] A. Dempster, N. Laird, and D. Rubin, "Maximim likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, vol. 39, pp. 1–38, 1977.

[5] Y. Ephraim, "Statistical model based speech enhancement systems," *Proceedings of the IEEE*, vol. 80, no. 10, pp. 1526–1555, 1992.

[6] D. Pearce and H. G. Hirsch, "The aurora experimental framework for the performance evaluatation of speech recognition system under noisy conditions," in *Proceedings of the 6th International Conference on Spoken Language Processing*, vol. 4, 2000, pp. 29–32.

[7] ITU-T Recommendation G.712, "Transmission performance characteristics of pulse code modulation channels," International Telecommunications Union, Geneva, Switzerland," ITU-T Rec.G712, 1996.

[8] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 1991.

[9] S. Vitabile, G. Pilato, G. Vassallo, S. M. Siniscalchi, A. Gentile, and F. Sorbello, "Neural classification of HEP experimental data," in *Proceeding of the Italian Workshop on Neural Nets (WIRN 2004)*, 2004.

[10] M. Fujimoto and Y. Ariki, "Evaluation of noisy speech recognition based on noise reduction and acoustic model adaptation on the Aurora2 tasks," in *International Conference on Spoken Language Processing*, vol. 1, 2002, pp. 465–468.