Speech Enhancement and Denoising using Kalman Filters

Alexandra Mikhael
UCSD
San Diego, CA
amikhael@ucsd.edu

Abstract

Speech recognition is the key to develop a full closed loop speech prosthesis to help people with difficulty speaking regain their speech. However, external noise is a big challenge to overcome to achieve accurate results. To simulate this environment, noise corrupted speech signals are taken from the NOIZEUS dataset, which include signals corrupted with white noise, train noise and babble noise at different SNRs. Kalman filters are adaptive filters that are used in this work to reduce the noise in the input signal and enhance speech. Two filter parameters are tuned : the measurement noise covariance, R, and the process noise covariance, Q. An optimal value of R and Q are determined, and the algorithm achieves good noise reduction and speech enhancement.

1. Introduction

Speech recognition involves taking speech patterns from a person's voice, processing them through a computer, and then identifying the speech's content. It is interdisciplinary and draws on a variety of disciplines, including artificial intelligence, phonetics, linguistics, signal processing, and information theory. Speech enhancement plays an important role in improving the quality and intelligibility of speech under many application scenarios, such as teleconference systems, human-machine communication, and speech recognition.

Speech recognition is widely implicated in our technological life, such as iOS Siri, Google, Amazon Alexa, etc. However, due to the varieties of language dialogs and human speech accents, as well as the presence of noise in the environment, there are challenges in the development of speech prosthetics.

Many academics have worked on voice recognition technologies during the last few decades. Over the years of research and trials, Wiener and Kalman filtering has been proved to be one of the most efficient speech enhancements and denoising methods. [1].

1.1. Theory

The following phases can be used to subdivide the speaking process. First, the speaker's brain develops the neural impulses necessary for producing the intended speech or language information. Second, it is translated into language by the human brain, and the speaker then expresses the thought using a variety of volume, pitch, and timbre. After the speech is delivered, other listeners will be able to hear the sounds made by the speakers. Scientists can use various signal processing and statistical techniques to extract the speaker's speech attributes, like formants, and analyze its content.

2. Literature Review and Problem Statement

2.1.1 Linear Prediction Coefficients (LPC)

One of the most significant and popular speech analysis approaches is linear predictive analysis. This method's significance is rooted in both its relative speed of computation and its capacity to deliver precise estimates of the speech parameters. It is an all-pole model in the Z-transform domain since the linear predictive analysis method assumes that the speech can be described by a predictor model that only considers previous values of the output.

$$X(z) = H(z)E(z) = \frac{1}{1-\sum_{k=1}^{p} a_k z^{-k}} E(z).$$

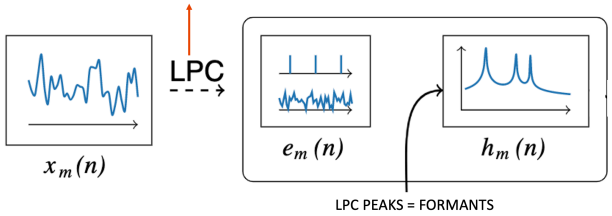

Figure 1: LPC method diagram

The order of the LPC model is crucial to obtain detailed coefficients of the input signal and to accurately extract the frequency components. We compared multiple orders of

LPC, ranging from order 3, 15, 30 to 50. The results show that the LPC of order 50 gives the best approximation of the input speech signal.

After running the algorithm, we get an LPC spectrum for each voiced frame as shown in figure 2.

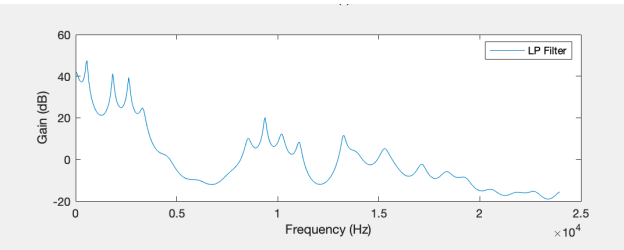

Figure 2: LPC Spectrum for one voiced frame

2.1.2 *Autoregressive Model of Speech*

In the Autoregressive model, where p is the order of the model, it is assumed that a sample at any given time depends on its prior p samples plus a stochastic component. By arguing that speech may be modeled as an all-pole, linear, time-varying filter triggered by either an impulse train of a certain pitch or noise, Linear Predictive Coding (LPC) [3] connects the AR model to speech production.

In a p^{th} order autoregressive process, the current sample, x(k), depends on the linear combination of the previous p samples plus a stochastic or random component that represents noise, yielding an all-pole FIR filter using Gaussian noise as input.

$$\mathrm{x(k)} = -\sum_{m=1}^{p} a_m\, x(k-m) + n(k) \qquad (2.1)$$

where a_m are the linear prediction coefficients (LPCs) and n(k), the process noise, is a zero-mean Gaussian noise with variance σ_n^2.

The autocorrelation function at lag l is given by :

$$R_{xx}(l) = E[x(k)x(k-l)] \qquad (2.2)$$

and so, (2.1) can be rewritten as :

$$\sum_{m=0}^{p} a_m\, x(k-m) = n(k), a_0 = 1 \qquad (2.3)$$

Thus, we can get the autocorrelation and cross-correlation terms from the following equation :

$$\sum_{m=0}^{p} a_m R_{xx}(l-m) \; = R_{nx}(l) \qquad (2.4)$$

where $R_{nx}(l) = 0$ for all l except $l = 0$ where it equals σ_n^2.

This can be written in matrix form as follows :

$$\begin{bmatrix} R_{xx}(0) & \cdots & R_{xx}(1-p) \\ \vdots & \ddots & \vdots \\ R_{xx}(p-1) & \cdots & R_{xx}(0) \end{bmatrix} x \begin{bmatrix} a_1 \\ \vdots \\ a_p \end{bmatrix} = - \begin{bmatrix} R_{xx}(1) \\ \vdots \\ R_{xx}(p) \end{bmatrix} \qquad (2.5)$$

2.1.3 *Kalman Filter*

The Kalman filter was suggested by R. Kalman in his well-known study [1] as a filtering method to predict unknown states of a dynamic system. It essentially consists of a collection of recursive equations that minimize the mean squared error to estimate the state of a system. There are two phases for the Kalman filter algorithm's operation. The prediction phase is the initial stage. The filter here provides estimates of the current state variables. Additionally, it gives their probability. The second stage is where the filter learns the results of the subsequent measurement or subsequent portion of the signal, which may be distorted and noisy. It then uses them to update these estimates with the use of a weighted average, wherein greater weight is given to the estimate with the highest probability.

The Kalman filter is a recursive method. The Kalman gain, i.e. the weight given to measurements of the current-state estimate, is a crucial computation variable. This number is tuned to achieve a specified output. In essence, the Kalman gain indicates how much we should modify our estimate of a given value. As a result, we need to find the error covariance matrix at every given time instant to derive the Kalman gain matrix and minimize the variances of each item, given by the diagonal entries. [11]

Now, (2.1) can be re-written in matrix form as :

$$\begin{bmatrix} x(k-p+1) \\ \vdots \\ x(k) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_p & -a_{p-1} & -a_{p-2} & \cdots & -a_1 \end{bmatrix} \begin{bmatrix} x(k-p) \\ x(k-p+1) \\ \vdots \\ x(k-1) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} n(k) \qquad (2.6)$$

or $$\mathrm{X(k) = \varphi X(k-1) + D^{*}n(k)} \qquad (2.7)$$

where X(k) is the (p×1) state vector matrix, φ is the (p×p) state transition matrix that uses LPCs calculated from noisy speech, D is the (p×1) input matrix and n(k) is the noise corrupted input signal at the kth instant.

When the speech signal is corrupted by noise, the output y(k) is given by:

$$y(k) = x(k) + u(k) \quad (2.8)$$

where u(k) is the measurement noise, a zero-mean Gaussian noise with variance σ_u^2 . In vector form, we get :

$$y(k) = \mathbf{O}^*\mathbf{X}(k) + u(k) \quad (2.9)$$

where O is the (1×p) observation matrix given by O = [0 0 ⋯ 0 1]. The Kalman filter calculates $\hat{X}$ (k|k) which is the estimate of the state vector X(k), given corrupted speech samples up to instant k, by using the following equations:

$$\hat{X}(k|k-1) = \phi\hat{X}(k-1|k-1) \quad (2.10)$$

where $\hat{X}$ (k|k-1) is the a priori estimate of the current state vector X(k).

$$C(k|k-1) = \phi C(k-1|k-1)\phi^T + DQD^T \quad (2.11)$$

where C(k|k-1) is the error covariance matrix of the a priori estimate, given by $E[w_k^- w_k^{-T}]$ where w_k^- =X(k)-$\hat{X}(k|k-1)$. Q is the process noise covariance matrix, which in this case is σ_n^2. Similarly, R is the measurement noise covariance matrix, which is σ_u^2.

$$K(k) = C(k|k-1)O^T(O\,C(k|k-1)O^T + R)^{-1} \quad (2.12)$$

where K(k) is the Kalman gain for the kth instant. The measurement update equations are thus given by :

$$\hat{X}(k|k) = \hat{X}(k|k-1) + K(k)(y(k) - O\hat{X}(k|k-1) \quad (2.13)$$

$$C(k|k) = (I - K(k)O)C(k|k-1) \quad (2.14)$$

3. Proposed Approach

In this paper, we will implement a Kalman filter where the two filter parameters that need to be tuned are the measurement noise covariance, R in (2.12), and the process noise covariance, Q in (2.11). Accurate estimation of these parameters can greatly boost filter performance for speech enhancement and denoising.

3.1. Dataset Description

The speech datasets used in this work come from two sources. The first dataset comes from the TIMIT database. It is an acoustic-phonetic continuous speech corpus. TIMIT contains broadband recordings of 630 speakers of eight major dialects of American English, each reading ten phonetically rich sentences. The TIMIT corpus includes time-aligned orthographic, phonetic and word transcriptions as well as a 16-bit, 16kHz speech waveform file for each utterance. Corpus design was a joint effort among the Massachusetts Institute of Technology (MIT), SRI International (SRI) and Texas Instruments, Inc. (TI). The speech was recorded at TI, transcribed at MIT and verified and prepared for CD-ROM production by the National Institute of Standards and Technology (NIST). [12] This dataset is non-noisy and will serve as a benchmark for the implementation.

The second dataset is the NOIZEUS dataset. It is a noisy speech corpus for evaluation of speech enhancement algorithms [13]. The noisy database contains 30 IEEE sentences (produced by three male and three female speakers) corrupted by eight different real-world noises at different SNRs. The noise was taken from the AURORA database and includes suburban train noise, babble, car, exhibition hall, restaurant, street, airport, and train-station noise. This dataset will be used to test the efficiency of the implementation in terms of denoising and speech enhancement.

3.2. Speech Detection

Speech being non-stationary and having high redundancy of samples during silent periods (non-speaking), pre-processing the speech signal is crucial to extracting meaningful information and reducing algorithm processing time.

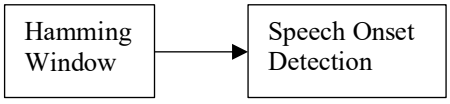

Figure 3: Pre-processing steps diagram

3.2.1 Hamming Window

Speech is a non-stationary signal, which means that it varies with time and cannot be considered as linear time invariant. However, it has been shown that a frame of 20 - 30 ms could be considered as time invariant for analysis [14]. For the purpose of this work, we compared two windowing methods : Hamming window and Hanning window. The Hamming window [15] can be described as follows :

$$w(n) = \begin{cases} 0.54 - 0.46 * cos\dfrac{2\pi n}{N-1}, 0 \le n \le N-1 \\ 0, otherwise \end{cases} \quad (3.1)$$

The Hanning window can be described as :

$$w(n) = \begin{cases} 0.5 - 0.5 * cos\dfrac{2\pi n}{N-1}, 0 \le n \le N-1 \\ 0, otherwise \end{cases} \quad (3.2)$$

The length of the window is crucial for frequency analysis since if the window is large in the time domain, it will be narrow in the frequency domain and vice versa.

Our window length is 10 ms and the comparison between the two windowing options is shown in figure 4 and figure 5.

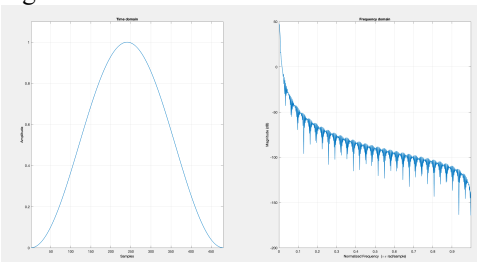

Figure 4: Hamming window

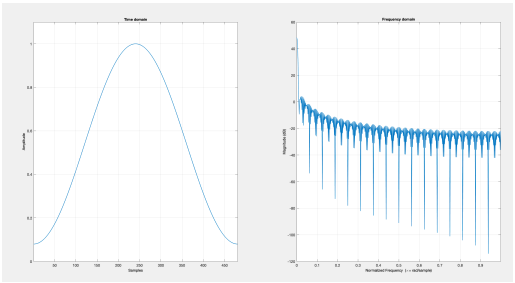

Figure 5: Hanning window

Both windows have the characteristics of low pass and symmetry. The main lobe of hamming window is the widest, has the lowest side lobe level, has relatively stable spectrum for speech signal, and it helps to enhance the characteristics of the central section of signal.

3.2.2 Speech Onset Detection

In order to reduce the processing time and errors in the Kalman filter's parameter tuning, we developed a way to detect the onset of speech production. By doing so, we can run the algorithm only on the voiced part of the input signal (speech part), which will resolve the issue of detecting random states in the silence part.

The detection algorithm is based on the energy and zero-crossing rate of the input signal. In fact, during speech, the energy of the signal is high compared to periods of silence and by setting a threshold, we can detect the onset of speech production and set a classification algorithm that will detect periods of silence, unvoiced or voiced frames. Once a frame is detected to be voiced (i.e speech), the speech enhancement/Kalman filter algorithm will then run on this specific frame.

Average energy can be defined as:

$$E = \sum_{m=0}^{N-1} [w(m)x(n-m)]^2 , 0 \le m \le N-1 \quad (3.3)[16]$$

where x(n) is the speech signal, N the length of frame, m is the frame shift, w(m) is the window function which expressed as $w(m) = \begin{cases} 1, & 0 \le m \le N-1 \\ 0, & otherwise \end{cases}$

Another metric that has been used to identify the onset of speech is zero-crossing rate (ZCR). It shows how many times the x-axis is crossed by a frame of voiced signal. One of the most straightforward techniques for time domain speech analysis is zero crossing analysis. ZCR can be defined as [17] :

$$ZCR = \frac{1}{2} \sum_{m=0}^{N-1} |sgn[x(m) - sgn[x(m-1)]|. w(n-m) \quad (3.4)$$

Unvoiced sounds have a larger ZCR than voiced sounds because their energy is more concentrated in the upper frequencies. Therefore, we can discriminate between voiced and unvoiced frames using the ZCR and energy of the signal.

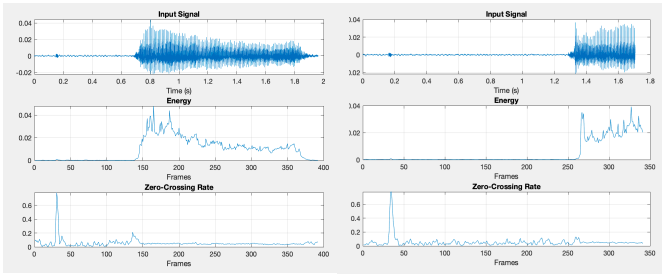

Figure 6 : Energy and Zero-Crossing rate of the two trials

3.3. Filter Tuning : Optimum Filter Parameters

The two filter parameters that need to be tuned are the measurement noise covariance, R in (2.12) and the process noise covariance, Q in (2.11). For the AR model of speech, R and Q are scalar quantities the values of which are the variances of process noise (σ_n^2), and measurement noise (σ_u^2) respectively.

3.3.1 Measurement Noise Covariance, R

In [10], the autocorrelation function of noisy measurement leads to the equation:

$$\sigma_u^2 = \frac{\sum_{i=1}^{p} a_i [R_{yy}(i) + \sum_{k=1}^{p} a_k R_{yy}(|i-k|)}{\sum_{i=1}^{p} a_i^2} \quad (3.5)$$

However, this formula leads to very high values of R. Instead, what we use in the algorithm are the previously found voiced frames over which we compute the PSD (Power Spectral Density) truncated within the frequency range [100Hz 2000Hz] where the spectral components of human speech are located.

3.3.2 Process Noise Covariance, Q

This parameter is harder to find accurately since it depends on the process model. In [8], two metrics, robustness J_1 and sensitivity J_2 were defined as well as Q_c for voiced frames and Q_2 for silent frames ($Q_2 < Q_c$).

Exactly as described in [7], let A_k and B be two terms defined for every frame as :

$$A_k = O(\phi C(k-1|k-1)\phi^T)O^T \quad (3.7)$$
$$B = O(DQD^T)\,O^T \quad (3.8)$$

Here, A_k, B and R are scalars. Furthermore, R is constant across frames since it is the variance of the noise corrupting the speech, whereas B is varied across frames to better represent the dynamics of the process.

J_1 and J_2 as well as n_q, a controlling parameter, are defined in [18] as follows :

$$J_1 = [(A_k + B + R)^{-1} R] = \frac{\sigma_w^2}{A_k+\sigma_w^2+\sigma_u^2} \quad (3.9)$$
$$J_2 = [(A_k + B)^{-1} B] = \frac{\sigma_u^2}{A_k+\sigma_u^2} \quad (3.10)$$
$$n_q = log_{10}(B) = log_{10}(\sigma_u^2) \quad (3.11)$$

For each frame, $n_q = n \times \log_{10} Q$ and so for each value of n, corresponding Q, J_1, J_2 values are determined. Large values of Q may indicate robust filter performance, whereas lower values of Q indicate sensitive filter performance. A trade-off is chosen as Q_c is the intersection of J_1 and J_2. We select a range of values around Q_c moving along the J_2 curve, but the value of Q is fixed between silent and voiced frames.

The Kalman filter equations (2.10 to 2.14) are executed on each voiced frame. Iterative Kalman filtering is done, and LPCs are calculated from a posteriori state estimates, X(k|k). Finally, overlap and add the state estimates to obtain the enhanced speech output.

4. Results and Discussions

4.1.1 Quantitative Results

We require certain assessment measures in order to compare the enhanced speech to the original clean speech and objectively assess its quality. SNR, Segmental SNR, and Frequency Weighted Segmental SNR are examples of common objective measurements mentioned in [18].

Segmental SNR is among them and, in contrast to several other techniques, is more compatible with subjective preference rating [18]. Therefore, we base our assessment of the effectiveness of our algorithm on the difference between the segmental SNR of noisy and improved speech. Segmental SNR is given by the formula:

$$SegSNR = \frac{1}{N}\sum_{i=1}^{N} 10log_{10}[\frac{\sum_{n\in framek}|s(n)|^2}{\sum_{n\in framek}|\hat{s}(n) - s(n)|^2}] \quad (4.1)$$

where s(n) is the noise-free signal and sˆ(n) is the enhanced speech signal. N is the number of frames and n denotes the samples in the k^{th} frame. The Segmental SNR is expressed in Decibels (dB) where a higher value usually indicates more noise removal from enhanced speech.

The SNR and PESQ table can be found in Appendix A.

Another metric for speech evaluation is the Perceptual Evaluation of Speech Quality (PESQ) test which is discussed by Hu and Loizou in [17]. It is a collection of standards that includes a test procedure for an automated evaluation of the speech quality as perceived by a telecommunication system user. ITU-T recommendation P.862 (02/01) standardizes it. A high PESQ number denotes the speech enhancement algorithm's excellent performance. Figure 7 shows the PESQ assessment block diagram.

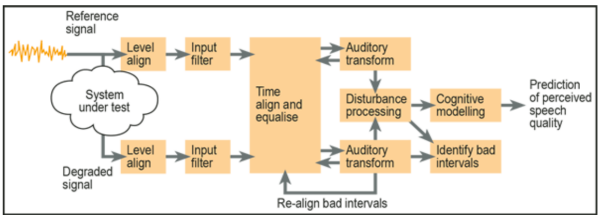

Figure 7 : PESQ Block Diagram

PESQ provides information regarding the perceptual quality of improved speech, whereas segmental SNR indicates the degree of noise reduction. Rarely, an extremely high segmental SNR can be misleading, resulting from considerable noise and speech spectrum reduction. The enhanced speech will then have a low PESQ, showing that the high segmental SNR was caused by a loss of intelligibility. In order to assess speech enhancement methods, these parameters are therefore complementary to one another.

Segmental SNR and PESQ tests were carried out on a sample of speech corrupted with three different types of noise (white, train and babble) and tested with multiple

values of Q around Q_c. The Segmental SNR plots are given in figure 8 and the PESQ plots are given in figure 9.

It is observed that for white noise $Q > Q_c$ gives better results. For train and babble noise, the value of Q that gives best performance depends on the SNR of noise corrupted speech. Q_c performs better when the speech SNR is low. The optimum performance is achieved for moderate SNR when $Q = Q_c$ and for low SNR when $Q > Q_c$. This is because the a priori state estimate should be trusted more in low SNR. In other words, a lower value of Q satisfies the need for a more sensitive performance. Robustness is prioritized since it increases the measurement's credibility for speech with high SNR levels. Therefore, a larger value of Q yields better outcomes. A trade-off between sensitivity and robustness yields the optimum performance for moderate levels of noise where $Q = Q_c$.

4.1.2 Qualitative Results

One metric to evaluate qualitative results of our implementation is by studying the spectrograms of the clean speech, noisy speech, and enhanced speech. The spectrogram is a 3D diagram that shows the Short Time Fourier Transform (STFT) of a non-stationary signal, with depth of color representing amplitude in decibels (dB) and time and frequency on the x and y axes, respectively. The results are shown in figure 8 for a speech signal corrupted by train noise at 0dB SNR. More results can be found in Appendix B. The spectrograms clearly show that a lower order model removes noise more effectively than a higher order model. However, the improved output is more understandable thanks to the higher order models' preservation of more spectral components.

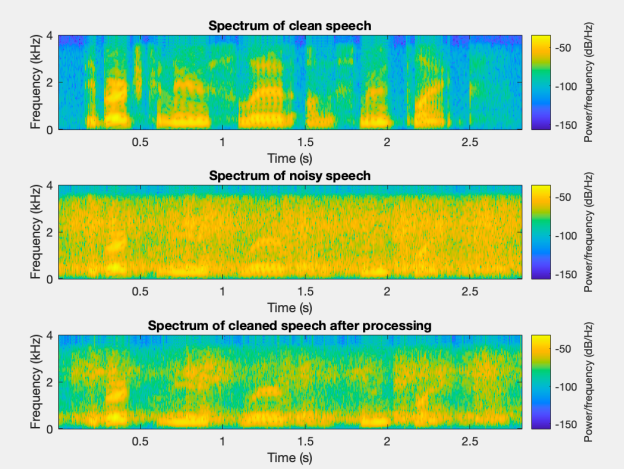

Figure 8 : One example of Spectrograms clean speech, corrupted speech with train noise at 0dB SNR, and enhanced speech

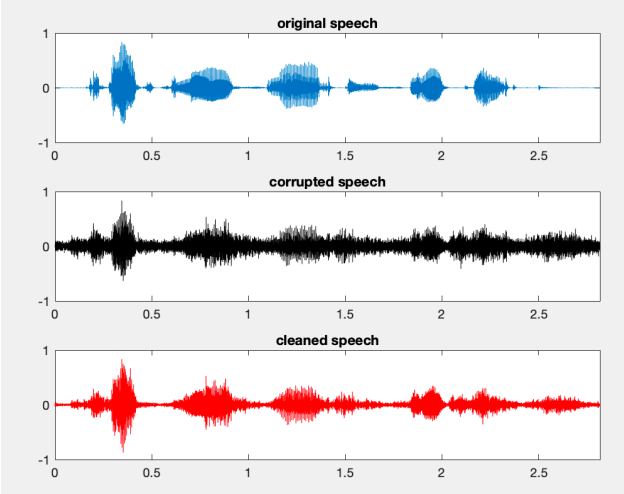

Figure 9 : Speech Signals for clean speech, corrupted speech with train noise at 0dB SNR, and enhanced speech

5. Conclusion

In conclusion, we first developed and algorithm to detect onset of speech and whether a given signal portion is voiced or unvoiced using the energy and ZCR. We then developed a Kalman filter implementation, using LPC and parameter tuning, that is capable of denoising speech signal. The types of noise added to the speech were white noise, train noise and babble noise at different SNRs. According to segmental SNR and PESQ, results show that the algorithm was successful at enhancing speech. However, some aspects are still to be worked on such as the real-time speech processing aspect as well as the order of the model.

APPENDIX A : SNR, PESQ and Process Noise Covariance for speech signals corrupted with different types of noise.

Noise Type	*Order*	*SNR (dB)*	*Best Q_i*	*SegSNR Noisy (dB)*	*SegSNR Process (dB)*	*PESQ*
White	*53*	*0*	Q_4 =0.00016374	-8.819208	0.740564	1.896065
White	*15*	*0*	Q_4 =0.00012732	-8.819208	1.655220	1.807339
White	*50*	*5*	Q_4 =9.7156e-005	-6.319208	1.828233	2.095206
White	*15*	*5*	Q_4 =9.8624e-005	-6.319208	3.002766	1.692509
White	*48*	*10*	Q_5 =0.00010170	-3.819208	3.488442	2.223566
White	*15*	*10*	Q_5 =0.00012086	3.819208	4.349849	2.018423
Train	31	*0*	Q_1 =0.00029121	-7.905742	-0.856497	1.961122
Train	15	*0*	Q_1 =0.00017351	-7.905742	-0.443979	1.897596
Train	*32*	*5*	Q_3 =0.00045214	-3.557488	1.114640	2.138670
Train	*15*	*10*	Q_6 =0.0190400	1.481678	2.635199	2.379670
Train	*29*	*10*	Q_6 =0.0180990	1.481678	2.455842	2.553597
Train	*15*	*5*	Q_1 =0.00010111	-3.557488	1.198366	2.026621
Babble	*15*	*0*	Q_1 =0.00061337	-8.295660	-2.491006	1.734588
Babble	*30*	*0*	Q_1 =0.00060997	-8.295660	-3.416224	1.808091
Babble	*15*	*5*	Q_3 =0.0014059	-3.685627	-0.583802	2.088812
Babble	*31*	*5*	Q_3 =0.0014178	-3.685627	-0.884745	2.215525
Babble	*15*	*10*	Q_2 =0.001072	1.270681	1.625915	2.411438
Babble	*29*	*10*	Q_5 =0.0076442	1.270681	1.564816	2.594426

APPENDIX B : Speech audio signal and spectrogram before and after enhancement for different types of noise at different SNRs.

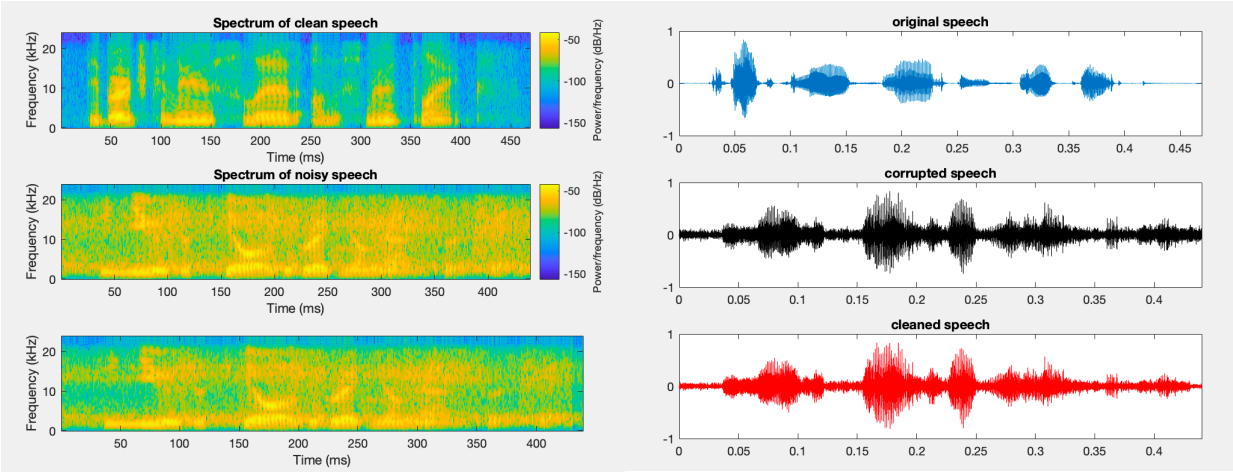

Figure S1 : a) Spectrograms of clean speech, corrupted speech with train noise at 5dB SNR, and enhanced speech, b) Corresponding audio signals

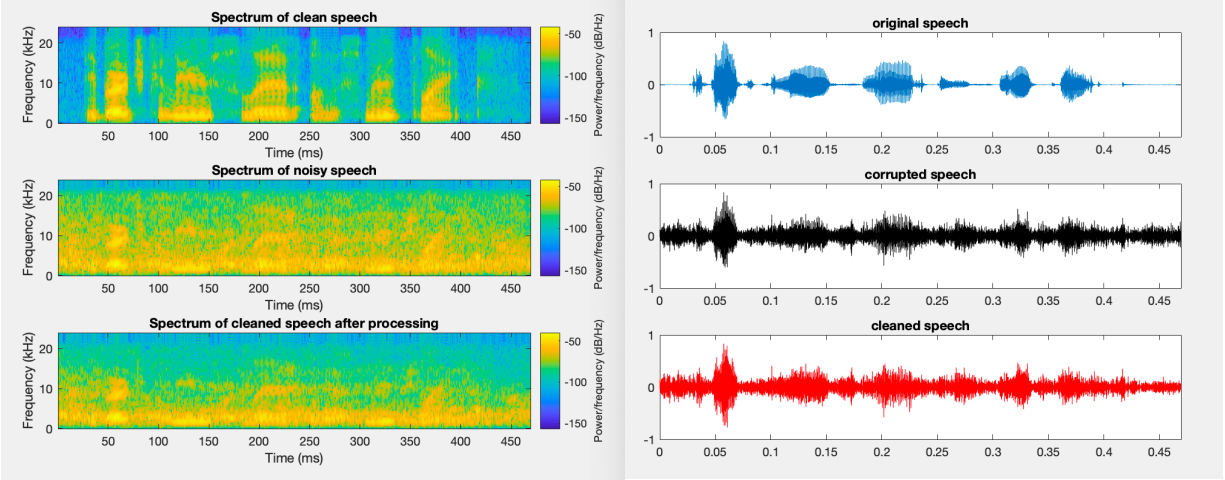

Figure S2 : a) Spectrograms of clean speech, corrupted speech with babble noise at 0dB SNR, and enhanced speech, b) Corresponding audio signals

References

[1] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. Transactions of the ASME–Journal of Basic Engineering, 82(Series D):35–45, 1960.

[2] Nehe, N. S., & Holambe, R. S. (2012). DWT and LPC based feature extraction methods for isolated word recognition. EURASIP Journal on Audio, Speech, and Music Processing, 2012(1).

[3] B.S Atal. Speech analysis and synthesis by linear prediction of the speech wave. Journal of the Acoustical Society of America, 47(1):65, 1970.

[4] Shahriar, S., & Hoq, M. N. (2016). Evaluation of LPC trajectory for Vowel-Consonant-Vowel sequence. 2016 19th International Conference on Computer and Information Technology.

[5] Noelia Alcaraz Meseguer. (2008). Speech Analysis for Automatic Speech Recognition. 87.

[6] Q. Xin och P. Wu, ”Research and Practice on Speaker Recognition Based on GMM,” Computer and Digital Engineering, p. vol 37(6), 2009. Y. Deng, X. Jing, H. Yang,

[7] Orchisama Das, Bhaswati Goswami, and Ratna Ghosh. Application of the tuned kalman filter in speech enhancement. In 2016 IEEE First International Conference on Control, Measurement and Instrumentation (CMI), pages 62–66. IEEE, 2016.

[8] M. Saha, R. Ghosh, and B. Goswami, “Robustness and sensitivity metrics for tuning the extended kalman filter,” IEEE Trans. Instrum. Meas., vol. 63, no. 4, pp. 964–971, 2014.

[9] A. Sulakhe, S. Mukherjee, B. Vishwakarma and K. S, "Audio Watermark Insertion and Enrichment of Speech via Kalman Filter and LSB," *2023 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE)*, Bengaluru, India, 2023, pp. 1123-1126

[10] K.K. Paliwal and A. Basu. A speech enhancement method based on kalman filtering. In Proc. ICASSP, volume 12, 1987.

[11] Garofolo, John S., et al. TIMIT Acoustic-Phonetic Continuous Speech Corpus LDC93S1. Web Download. Philadelphia: Linguistic Data Consortium, 1993.

[12] Hu, Y. and Loizou, P. (2007). “Subjective evaluation and comparison of speech enhancement algorithms,” *Speech Communication*, 49, 588-601.

[13] D. MANDALIA, P. GARETA, ”Speaker Recognition Using MFCC and Vector Quantization Model,” Institute of Technology, Nirma University, 2011.

[14] Y. Yang, ”Research on Endpoint Detection of Speech,” Computer Systems & Applications , p. vol 21(6), 2012.

[15] Y. Cai, ”Research on end point detection of speech signal,” University of Jiangnan, nr TN912.3 Master thesis, 2008.

[16] Mousumi Saha, Ratna Ghosh, and Bhaswati Goswami. Robustness and sensitivity metrics for tuning the extended kalman filter. Instrumentation and Measurement, IEEE Transactions on, 63(4):964–971, 2014.

[17] Bernard Grundlehner, Johan Lecocq, Radu Balan, and Justinian Rosca. Performance assessment method for speech enhancement systems. Citeseer.

[18] B. Schwerin and K. Paliwal. Using stft real and imaginary parts of modulation signals for mmse-based speech enhancement. Speech Communication, 58:49–68, 2014.

[19] Y. Hu and P. Loizou. Evaluation of objective quality measures for speech enhancement. IEEE Transactions on Speech and Audio Processing, 16:229–238, 2008.