Enhancement of Compressed Speech Signal using Recursive Filter

M.Suman¹, Dr. Habibulla Khan², Dr. M.Madhavi Latha³, D.Aruna Kumari⁴

Associate Professor¹ Professor and Head²  Professor & Director-ITC³  Associate Professor⁴
¹,² Life Member, CSI
³,⁴ K. L University, Vaddeswaram, Guntur, Andhra Pradesh.
³ JNTU college of Engineering, Hyderabad

Abstract: Speech compression, enhancement and recognition in noisy, reverberant conditions is a challenging task. In this paper a new approach to this problem, which is developed in the framework of probabilistic random modeling. speech coding techniques are commonly used in low bit rate analysis and synthesis. Coding algorithms seek to minimize the bit rate in the digital representation of a signal without an objectionable loss of signal quality in the process. As the compression techniques that are used are Lossy compression technique and there is every possibility of loss in quality. Speech enhancement aims to improve speech quality by using various algorithms. This paper deals with multistage vector quantization technique used for coding (compression) of narrow band speech signals. The parameter used for coding of speech signals are the line spectral frequencies, so as to ensure filter stability after quantization. The code books used for quantization are generated by using Linde, Buzo and Gray (LBG) algorithm. The performance of quantization is measured in terms of spectral distortion measured in dB, Computational complexity measured in KFlops and Memory Requirements measured in Floats. From the results it can be proved that multistage vector quantization is having better spectral distortion performance, less computational complexity and memory requirements when compared to unconstrained vector quantization. The existing Speech enhancement techniques like spectral subtraction and Kalman filters performances are compared with the proposed recursive filter and approach yields significantly estimating the parameters like signal to noise ratio subjected to white Gaussian Noise and Real time noise signals.

Keywords- Linear predictive Coding, Multi stage vector quantization, Line Spectral Frequencies (LSF).

1. INTRODUCTION

One of the major components in speech enhancement is “noise estimation”. In earlier methods residual noise will be present in the enhanced speech signal because of inaccurate noise estimation and is not suitable in non-stationary noise environments. In this research noise is estimated using a recursive filter. Therefore in this research, we will be looking more into speech processing with the aid of a recursive Filter. In this estimation estimator is recursively updated in each frame so that non-stationary noise is tracked and estimated.

In performance comparison proposed approach we present the SNR for additive white Gaussian noise at different dB’s and with different environment noises. These results shows that proposed approach will produce enhanced speech with very less additive noise when compared to spectral subtraction and Kalman Filter.

2. SPEECH ENHANCEMENT

  Enhancement means the improvement in the value or quality of something. When applied to speech, this simply means the improvement in intelligibility and/or quality of a degraded speech signal by using signal processing tools [26]. By
speech enhancement, it refers not only to noise reduction but also to de-reverberation and separation of independent signals. This is a very difficult problem for two reasons:

- First, the nature and characteristics of the noise signals can change dramatically in time and between applications. It is also difficult to find algorithms that really work in different practical environments.
- Second, the performance measure can also be defined differently for each application. Two criteria’s are often used to measure the performance like quality and intelligibility. It is very hard to satisfy both at the same time.

Speech enhancement is an area of speech processing where the goal is to improve the intelligibility, quality and/or pleasantness of a speech signal. The most common approach in speech enhancement is noise removal, where by estimation of noise characteristics, noise components can be cancelled and retain only the clean speech signal. The basic problem with this approach is that if those noise parts of the Noisy speech signal noise is removed, they are also bounded to remove those parts of the speech signal that reassemble noise. In other words, speech enhancement procedures, often inadvertently, also corrupt the speech signal when attempting to remove noise. Algorithms must therefore compromise between effectiveness of noise removal and level of distortion in the speech signal.

Current speech processing algorithms can roughly be divided into three domains, spectral subtraction, sub-space analysis and filtering algorithms.

1. Spectral subtraction algorithms operate in the spectral domain by removing, from each spectral band, that amount of energy which corresponds to the noise contribution. While spectral subtraction is effective in estimating the spectral magnitude of the speech signal, the phase of the original signal is not retained, which produces a clearly audible distortion known as “ringing”.

2. Sub-space analysis operates in the autocorrelation domain, where the speech and noise components can be assumed to be orthogonal, whereby their contributions can be readily separated. Unfortunately, finding the orthogonal components is computationally expensive. Moreover, the orthogonality assumption is difficult to motivate.

3. Finally, filtering algorithms are time-domain methods that attempt to either remove the noise component (Wiener filtering) or estimate the noise and speech components by a filtering approach (Kalman filtering).

3. DRAWBACKS OF SPECTRAL SUBTRACTION METHOD:

1. Presence of Residual Noise (Musical Noise): It is obvious that the effectiveness of the noise removal process is dependent on obtaining an accurate spectral estimate of the noise signal. The better the noise estimate, the lesser the residual noise content in the modified spectrum. However, since the noise spectrum cannot be directly obtained, it is forced to use an Average estimate of the noise.

Hence there are some significant variations between the estimated noise spectrum and the actual noise content present in the instantaneous speech spectrum.. However, due to the limitations of the single –channel enhancement methods, it is not possible to remove this noise completely, without compromising the quality of the enhanced speech.

2. Roughening of Speech due to the noisy phase: The phase of the Noise-corrupted signal is not enhanced before being combined with the modified spectrum to regenerate the enhanced time signal. This is due to the fact that the presence of noise in the phase information does not contribute immensely to the degradation of the speech quality.

This is especially true at high SNRs (>15dB). However, at low SNRs (<0dB), the noisy phase can lead to a perceivable roughness in the speech signal contributing to the reduction speech quality. Most speech enhancement algorithms, including the spectral subtraction methods, try to optimize noise removal based on mathematical models of the speech and noise signals. However, speech is a subtle form of communication and is heavily dependent on the relationship of one frequency with another. Hence, while conventional speech enhancement algorithms can increase the speech quality of the noisy speech by increasing the SNR, there is no significant increase in speech intelligibility.

4. DISADVANTAGES OF KALMAN FILTER:

Among the filter disadvantages we can find that it is necessary to know the initial conditions of the mean and variance state vector to start the recursive algorithm. There is no general consent over the way of determinate the initial conditions. The Kalman filter development, as it is found on the original document, is supposed a wide knowledge about probability theory, specifically with the Gaussian condition for the random variables, which can be a limit for its research and application. When it is developed for autoregressive models, the results are conditioned to the past information of the variable under study. In this sense the prognostic of the series over the time represents the inertia that the system actually has and they are efficient just for short time term.

This recursive Filter is an estimator for what is called the “linear quadratic problem”, which focuses on estimating the instantaneous “state” of a linear dynamic system perturbed by white noise. Statistically, this estimator is optimal with respect to any quadratic function of estimation errors.

5. RECURSIVE PROCESS:

After going through some of the introduction and advantages of the filter, we will now take a look at the process. The process commences with the addresses of a general problem of trying to estimate the state of a discrete-time controlled process that is governed by a linear stochastic difference equation:

\[ x_k = Ax_{k-1} + Bu_k + w_{k-1} \]  

with a measurement that is

\[ z_k = Hx_k + v_k \]

\[ x_k = Ax_{k-1} + Bu_k + w_{k-1} \]  

\[ z_k = Hx_k + v_k \]
The random variables represent the process and measurement noise (respectively). We assume that they are independent of each other, white, and with normal probability distributions

\[
P(w) = \text{N}(0, R) \quad \text{(3)}
\]

\[
P(V) = \text{N}(0, R) \quad \text{(4)}
\]

Ideally, the process noise covariance \( R \) and measurement noise covariance \( K \) matrices are assumed to be constant, however in practice, they might change with each time step or measurement. In the absence of either a driving function or process noise, the \( n \times n \) matrix \( A \) in the difference equation \( (1) \) relates the state at the previous time step \( k-1 \) to the state at the current step \( k \). In practice, \( A \) might change with each time step or measurement, however we assume it is constant.

The \( n \times l \) matrix \( B \) relates the optional control input to the state \( x \). \( H \) which is a matrix in the measurement equation \( (2) \) which relates the state to the measurement, \( z \). In practice \( H \) might change with each time step, however we assume it is constant.

6. RECURSIVE ALGORITHM

This section will begin with a broad overview, covering the "high-level" operation of one form of this filter. After presenting this high-level view, I will narrow the focus to the specific equations and their use in this discrete version of the filter. Firstly, it estimates a process by using a form of feedback control loop whereby the filter estimates the process state at some time and then obtains feedback in the form of (noisy) measurements. As such, these equations for this filter fall into two groups: "Time Update equations" and "Measurement Update equations".

The responsibilities of the time update equations are for projecting forward (in time) the current state and error covariance estimates to obtain the priori estimates for the next time step. The measurement update equations are responsible for the feedback i.e. for incorporating a new measurement into the priori estimate to obtain an improved posteriori estimate.

The time update equations can also be thought of as "predictor" equations, while the measurement update equations can be thought of as "corrector" equations. By and large, this loop process of the final estimation algorithm resembles that of a predictor-corrector algorithm for solving numerical problems. As the time update projects the current state estimate ahead in time, the measurement update adjusts the projected estimate from the time update by an actual measurement at that particular time. The specific equations for the "time" and "measurement" updates are presented below in Table 6.1 and Table 6.2.

\[
x_k = Ax_k-1 + Bu_k \quad \text{(5)}
\]

\[
P_k = AP_{k-1}A^T + \quad \text{(6)}
\]

Once again, notice how the time update equations in Table 6.1 project its state, \( x \) and covariance, \( P_k \) estimates forward from time step \( k-1 \) to step \( k \). As mentioned earlier, the matrices \( A \) and \( B \) are from \( (1) \), while is from \( (3) \). Initial conditions for the filter are discussed in the earlier section.

\[
K_k = P_kH(HP_kH^T+R)^{-1} \quad \text{(7)}
\]

\[
x_k = x_k + (z_k - Hx_k) \quad \text{(8)}
\]

\[
P_k = (I - K_kH)P_k \quad \text{(9)}
\]

By referring to above data, it is obvious that the first task during the measurement update is to compute the gain, \( k_k \). By comparing \( (7) \) in the table below and the previous section, notice the equations are the same. Next, is to actually measure the process in order to obtain \( z_k \) and then to generate a posteriori state estimate \( x_k \) by incorporating the measurement as in \( (8) \). Once again, notice the repeated equation of \( (8) \) here for completeness. Finally, the last step is to obtain a posteriori error covariance estimate via \( (9) \).

Thus, after each time and measurement update pair, this loop process is repeated to project or predict the new time step priori estimates using the previous time step posteriori estimates. This recursive nature is one of the very appealing features of this filter that it makes practical implementations much more feasible than (for example) an implementation of a kalman filter which is designed to operate on all of the data directly for each estimate. Instead, this filter recursively conditions the current estimate on all of the past measurements. The high-level diagram is combined with the equations from Table 6.1 and Table 6.2, and in Table 6.2 as shown below, which offers a much more complete and clear picture of the operation of the recursive filter.

**Table 6.1: Time update equations**

<table>
<thead>
<tr>
<th>Time update (“predict”)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Project the state head</td>
</tr>
<tr>
<td>( x_k = f(x_{k-1}, u_k, O) )</td>
</tr>
<tr>
<td>2. Project the error covariance ahead</td>
</tr>
<tr>
<td>( P_k = Ap_k-1A^T + W_{k-1}Q_{k-1}W_{k-1}^T )</td>
</tr>
</tbody>
</table>

**Table 6.2: Measurement updates equations**

<table>
<thead>
<tr>
<th>Measurement update (“correct”)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Compute the gain</td>
</tr>
<tr>
<td>( K_k = P_kH(HP_kH^T + V_kR_kV_k^T)^{-1} )</td>
</tr>
<tr>
<td>2. Update estimate with measurement</td>
</tr>
<tr>
<td>( x_k = x_k + K_k(z_k - h(x_k, 0)) )</td>
</tr>
<tr>
<td>3. Update the error covariance</td>
</tr>
<tr>
<td>( P_k = (I - K_kH)P_k )</td>
</tr>
</tbody>
</table>
7. IMPLEMENTATION:

From a statistical point of view, many signals such as speech exhibit large amounts of correlation. From the perspective of coding or filtering, this correlation can be put to good use. The all pole, or autoregressive (AR), signal model is often used for speech. The AR signal model is introduced as:

\[ y_k = [1/1-\Sigma^N_{i=1} aZ] W_k \]  

Equation (10) can also be written in this form as shown below:

\[ y_k = a_1 y_{k-1} + a_2 y_{k-2} + \ldots + a_N y_{k-N} + W_k \]  

(11)

where,

\[ k \rightarrow \text{Number of iterations}; \]
\[ y_k\rightarrow \text{current input speech signal sample}; \]
\[ y_{k-N}\rightarrow (N-1)\text{th sample of speech signal}; \]
\[ a_n\rightarrow \text{Nth filter coefficient}; \]
\[ w_k\rightarrow \text{excitation sequence (white noise)}.

In order to apply this filtering to the speech expression shown above, it must be expressed in state space form as

\[ H_k=XH_{k-1}+W_k \]  

(12)
\[ y_k=gH_k \]  

(13)

\[ X = \begin{pmatrix} a_1 & a_2 & \ldots & a_{N-1} & a_N \\ 1 & 0 & \ldots & 0 & 0 \\ 0 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & 0 \end{pmatrix} \]

\[ H_k = \begin{pmatrix} y_k \\ y_{k-1} \\ \vdots \\ y_{k-N+1} \\ W_k \\
0 \\
0 \\
\end{pmatrix} \]

\[ g = (1 \ 0 \ \ldots \ 0) \]

\[ X \] is the system matrix; \( H_k \) consists of the series of speech samples; \( W_k \) is the excitation vector and \( g \), the output vector. The reason of \( (k-N+1) \text{th iteration is due to the state vector, } H_k, \text{ consists of } N \text{ samples, from the } k \text{th iteration back to the } (k-N+1) \text{th iteration. The above formulations are suitable for this filter.} \]

As mentioned in the previously, this filter functions in a looping method. Here we denote the following steps within the loop of the filter.

Define matrix \( H^T_{k-1} \) as the row vector:

\[ Hk-1T=\begin{bmatrix} yk-1yk-2 \ldots yk-N \end{bmatrix} \]  

(14)

and \( z_k= y_k \).

Then (11) and (14) yield \( z_k=Hk-1TXk+W_k \)  

(15)

Where \( X_k \) will always be updated according to the number of iterations, \( k \)

Note that when the \( k = 0 \), the matrix \( H_{k-1} \) is unable to be determined. However, when the time \( z_k \) is detected, the value in matrix \( H_{k-1} \) is known. The above purpose is thus sufficient enough for defining the recursive filter, which consists of: \( X_k= [1-Kk-1 TX_{k-1} + KkZ_k] \)  

(16)

\[ L = \begin{bmatrix} 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ 0 & 0 & \ldots & 1 \end{bmatrix} \]

With \( K_k=P_{k-1}H_{k-1}[H^T_{k-1}P_{k-1}H_{k-1}+R] \)  

(17)

Where \( K_k \) is the filter.

\( P_{k-1} \) is the priori error covariance matrix.

\( R \) is the measurement noise covariance

\[ P_k=P_{k-1} \]  

(18)

[\begin{bmatrix} H^T_{k-1}P_{k-1} \end{bmatrix} + H^T_{k-1}P_{k-1}+Q]

Where \( P_k \) is the posteriori error co-variance matrix

\[ Q = \begin{bmatrix} 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ 0 & 0 & \ldots & 1 \end{bmatrix} \]

Thereafter the reconstructed speech signal, \( Y_k \) after filtering will be formed in a manner similar to (11):

\[ Y_k = a_1 Y_{k-1} + a_2 Y_{k-2} + \ldots + a_N Y_{k-N} + W_k \]  

(19)

Since the value of \( y_k \) is the input at the beginning of the process, there will be no problem forming \( H^T_{k-1} \). In that case a question rises, how is \( Y_k \) formed? The parameters \( w_k \) and \( \{a\} \) are determined from application of this filter to the input speech signal \( y_k \). That is in order to construct \( Y_k \), we will need matrix \( X \) that contains the filtering coefficients and the white noise, \( w_k \) which both are obtained from the estimation of the input signal. This information is enough to determine \( HH_{k-1} \)

\[ HH_{k-1} = \begin{bmatrix} y_{k-1} \\ y_{k-2} \\ \vdots \\ y_{k-N+1} \end{bmatrix} \]

Thus, forming the equation (19) mentioned above.
8. RESULTS:

Table 1: 8.1 SNR with white Gaussian Noise

<table>
<thead>
<tr>
<th>Noise Type</th>
<th>Signal to Noise ratio( in dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>White Gaussian noise level</td>
<td>After Compression using MSVQ</td>
</tr>
<tr>
<td>2db</td>
<td>-16.6607</td>
</tr>
<tr>
<td>5db</td>
<td>-11.8891</td>
</tr>
<tr>
<td>10db</td>
<td>-8.7907</td>
</tr>
<tr>
<td>15db</td>
<td>-14.8324</td>
</tr>
<tr>
<td>20db</td>
<td>-31.1697</td>
</tr>
<tr>
<td>25db</td>
<td>-36.9003</td>
</tr>
</tbody>
</table>

Table 2: SNR with Real Time Noise

<table>
<thead>
<tr>
<th>Type of Real-Time Noise</th>
<th>SNR in dB</th>
<th>SNR in dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>After Compression using</td>
<td>Enhancement Using spectral subtraction</td>
<td>Enhancement using Kalman filter</td>
</tr>
<tr>
<td>Factory</td>
<td>-23.3076</td>
<td>-10.0212</td>
</tr>
<tr>
<td>Fire engine</td>
<td>-22.2793</td>
<td>-4.9626</td>
</tr>
<tr>
<td>Machine gun</td>
<td>-17.6370</td>
<td>-10.7221</td>
</tr>
<tr>
<td>Volvo Bus</td>
<td>-19.7961</td>
<td>-10.5267</td>
</tr>
<tr>
<td>Destroyer</td>
<td>-19.0281</td>
<td>-9.6162</td>
</tr>
<tr>
<td>Ambulance</td>
<td>-6.2175</td>
<td>2.7582</td>
</tr>
<tr>
<td>Pink</td>
<td>-16.7981</td>
<td>-9.1724</td>
</tr>
<tr>
<td>Traffic</td>
<td>-20.7846</td>
<td>-11.9805</td>
</tr>
</tbody>
</table>

9. CONCLUSIONS

In this research, an implementation of employing this recursive filtering to speech processing had been developed. As has been previously mentioned, the purpose of this approach is to reconstruct an compressed speech signal by making use of the accurate estimating ability of this filter. True enough, simulated results had proven that this filter indeed has the ability to estimate accurately. Furthermore, the results have also shown that this recursive filter could be tuned to provide optimal performance.

10. ACKNOWLEDGMENT

We sincerely thank Dr. M. Satya Sai Ram for continuous guidance and support.

11. REFERENCES


