Speech Enhancement Using A Recursive Filter

[1]k.S.N.Murthy, [2]m.Suman, [3]v.Sriram, [4]n.Sai Kotewararao,[5]b.Phani Krishna Rao

[1,2]Professors,Department of ECE, K L C E, Guntur DT, AP, India
[3,4,5]Students, Department of ECE, K L C E, Guntur DT, AP, India

Abstract–

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 not suitable in non-stationary noise environments.
In this paper noise is estimated using 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 of segmental SNR values for different environments. These results shows that proposed approach will produce enhanced speech with less residual noise.

Keywords-*recursive filter, time update measurement update, error co-variance*

1. INTRODUCTION

Speech is a form of communication in everyday life. It existed since human civilizations began and even till now, speech
is applied to high technological telecommunication systems. As applications like cellular and satellite technology are getting popular among mankind, human beings tend to demand more advance technology and are in search of improved applications. For this reason, researchers are looking closely into the four generic attributes of speech coding. They are complexity, quality, bit rate and delay. Other issues like robustness to transmission errors, multistageencoding/decoding and accommodation of non-voice signals such as data play an important role in coding of speech as well.

In order to understand these processes, both human and machine speech
has to be studied carefully on the structures and functions of spoken language: how we produce and perceive it and how speech technology may assist us in communication. Therefore in this project, we will be looking more into speech processing with the aid of a recursive Filter.

2. RECURSIVE FILTER

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

3. PROCESS

After going through some of the introduction and advantages of using this 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} ………………… …………….(1)

with a measurement $z \in \Re^m$ that is

z_k=Hx_k+v_k……………………(2)

The random variables and represent the process and measurement noise (respectively). We assume that they are independent of each other, white, and with normal probability distributions

P(w)-N(0,R)…………(3)

P(V)-N(0,R)……………(4)

Ideally, the process noise covariance Q and measurement noise covariance R matrices are assumed to be constant, however in practice, they might change with each time step or measurement.

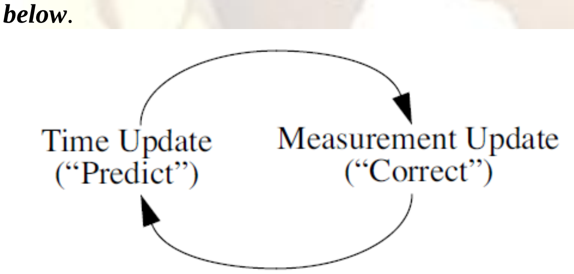
In the absence of either a driving function or process noise, the n×n matrix **A** in the difference equation **(1)** relates the state at the previous time step k-1 to the state at the current step k . In practice, **A** might change with each time step, however here it is assumed constant. The n×l matrix **B** relates the optional control input $u \in \Re^l$ to the state **x**. H which is a matrix in the measurement equation **(2)** which relates the state to the measurement, z_k. In practice H might change with each time step or measurement, however we assume it is constant.

4. ALGORITHM

This section will begin with a broad overview, covering the "high-level" operation of one form of this fillter. 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 just like the one shown in ***Fig below***.



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 4.1*** and ***Table 4.2***

$$\hat{x}_k = A\hat{x}_{k-1} + BU_k \ldots\ldots\ldots\ldots\ldots(5)$$

$$P_k = AP_{k-1}A^T + Q \ldots\ldots\ldots\ldots\ldots..(6)$$

Table 4.1:Time update equations

Once again, notice how the time update equations in ***Table 4.1*** project its state, **x** and covariance, P^kestimates forward from time step k-1 to step k. As mentionedearlier, the matrixes **A** and **B** are from **(1)**, while is from **(3)**. Initial conditions forthe filter are discussed in the earlier section.

$$K_k = \bar{P}_k H^T (H\bar{P}_k H^T + R)^{-1} \ldots\ldots\ldots.(7)$$

$$\hat{x}_k = \overline{\hat{x}_k} + K_k(z_k - H\overline{\hat{x}_k}) \ldots\ldots\ldots(8)$$

$$P_k = (I - K_k H)\bar{P}_k \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(9)$$

Table 4.2: Measurement update equations

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 above 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$\hat{x}_k$ 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 → it makes practical implementations much more feasible than (for example) an implementation of a Wiener filterwhich 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 of ***Fig 4.1***is combined with the equations from ***Table 4.1***and ***Table 4.2***, in ***Fig 4.2***as shown below, which offers a much more complete and clear picture of the operation of the recursive filter.

Time update ("predict")

1. Project the state head

$$\overline{\hat{x}_k} = f(\hat{x}_{k-1}, u_k, 0)$$

2. Project the error covariance ahead

$$\bar{P}_k = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T$$

Measurement update ("correct")

1. Compute the gain

$$K_k = \bar{P}_k H_k^T (H_k \bar{P}_k H_k^T + V_k R_k V_k^T)^{-1}$$

2. Update estimate with measurement

$$\hat{x}_k = \overline{\hat{x}_k} + K_k(z_k - h(\overline{\hat{x}_k}, 0))$$

3. Update the error covariance

$$P_k = (I - K_k H_k)\bar{P}_k$$

Fig: Complete operation of the filter

5. 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:

$$y_k = \left[\frac{1}{1-\sum_{i-1}^{N} \alpha Z}\right] W_k \ldots\ldots\ldots(10)$$

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

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

…………………….. (11)

where,

k→ Number of iterations;
$\mathbf{y_k}$ → current input speech signal sample;
$\mathbf{y_{k-N}}$→ (N-1)th sample of speech signal;
$\mathbf{a_N}$ → Nth filter coefficient; and
$\mathbf{w_k}$ → 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

$$H_k = X H_{k-1} + \overline{W_k} \ldots\ldots\ldots\ldots(12)$$

$$y_k = g H_k \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(13)$$

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

$$H_k = \begin{pmatrix} y_k \\ y_{k-1} \\ y_{k-2} \\ \vdots \\ y_{k-N+1} \end{pmatrix}$$

$$\overline{w}_k = \begin{pmatrix} w_k \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

g=(1 0 ... 0)

X is the system matrix, $\mathbf{H_k}$consists of the series of speech samples; $\overline{W}_k$is the excitation vector and **g**, the output vector. The reason of (k-N+1)th iteration is due to the state vector, $\mathbf{H_k}$, consists of N samples, from the kth iteration back to the (k-N+1)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_{k-1}^Tas the row vector:

$$\boldsymbol{H_{k-1}^T} = -[\boldsymbol{y_{k-1} y_{k-2} \ldots\ldots y_{k-N}}]$$
.... (14)

And$\mathbf{z_k} = \mathbf{y_k}$. Then (11) and (14) yield

$$\boldsymbol{z_k = H_{k-1}^T X_k + W_k}$$
.... (15)

Where$\mathbf{X_k}$will always be updated according to the number of iterations, k.

Note that when the k = 0, the matrix $\mathbf{H_{k-1}}$ is unable to be determined. However, when the time $\mathbf{z_k}$is detected, the value in matrix $\mathbf{H_{k-1}}$ is known. The above purpose is thus sufficient enough for defining the recursive filter, which consists of:

$$X_k = [1 - K_k H_{k-1}^T] X_{k-1} + K_k Z_k$$
..............(16)

Where I= $\begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix}$

With $K_k = P_{k-1}H_{k-1}[H_{k-1}^T P_{k-1} H_{k-1} + R]$ ………………….(17)

Where K_k *is the filter gain*

$P_{k-1}$is the priori error covariance matrix

R is the measurement noise covariance

$$P_k = P_{k-1} - P_{k-1}H_{k-1} [H_{k-1}^T P_{k-1} H_{k-1} + R] H_{k-1}^T P_{k-1} + Q$$

…………………….(18)

Where P_k is the posteriori error co-variance Matrix

Q= $\begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix}$

Thereafter the reconstructed speech signal, $\mathbf{Y}_k$after filtering will be formed in a manner similar to (11):

$$\boldsymbol{Y_k = a_1 Y_{k-1} + a_2 Y_{k-2} + \ldots.. a_N Y_{k-N} + W_k}$$

Since the value of $\mathbf{y}_k$is the input at the beginning of the process, there will be no problem forming $\mathbf{H}^{\mathbf{T}}_{\mathbf{k-1}}$. In that case a question rises, how is $\mathbf{Y}_k$ formed? The parameters$\mathbf{w}_k$ and $\{a\}_{i-1}^N$are determined from application of this filter to the input speech signal $\mathbf{y}_k$. That is in order to construct $\mathbf{Y}_k$, we will need matrix **X** that contains the filtering coefficients and the white noise, $\mathbf{w}_k$ which both are obtained from the estimation of the input signal. This information is enough to determine $\mathbf{HH_{k-1}}$
where

$$\mathrm{HH}_{k-1} = \begin{bmatrix} Y_{k-1} \\ Y_{k-2} \\ Y_{k-N+1} \end{bmatrix}$$

Thus, forming the equation (19)mentioned above.

RESULTS
The SNR values for different types of noises

Input	O/P SNR for input=0dB	O/P SNR for input=5dB	O/P SNR for input= 10dB
airport	1.1520	4.9623	5.7959
babble	0.7560	4.8810	5.6148
car	2.4997	4.8747	4.3537
exhibition	2.3022	4.8301	4.5720
restaurant	0.8976	4.9388	5.0451
station	0.5629	3.3547	5.1048
street	1.8141	3.7160	5.033

Fig1: Below result is for the input speech

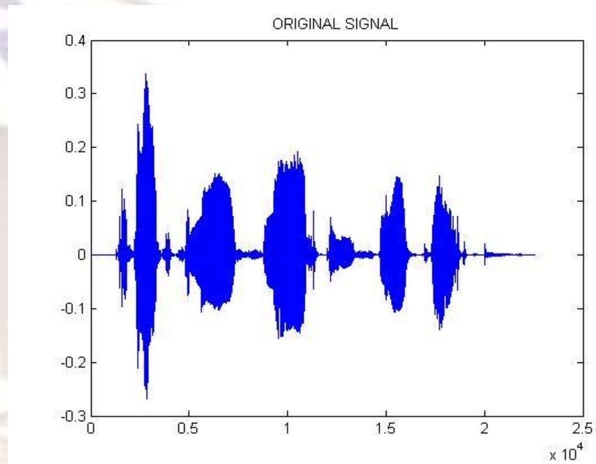


Fig2. Noise added

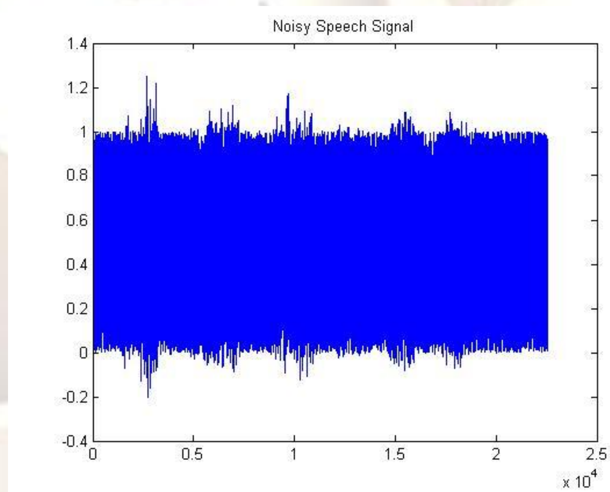


Fig3. Estimated signal

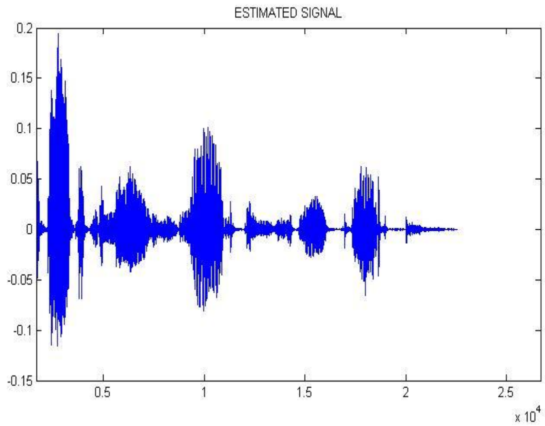


Fig 4: combined plot

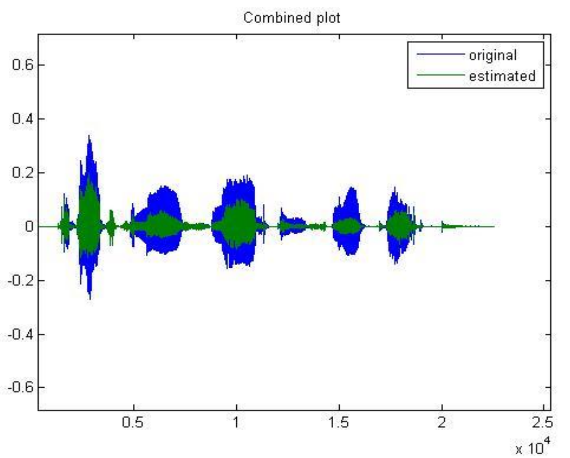
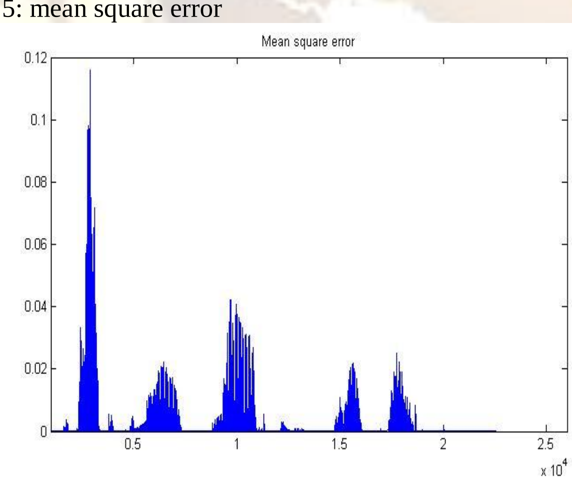


Fig 5: mean square error



CONCLUSION

Inthis project, 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 output 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.

ACKNOWLEDGEMENT

We would like to thank our respected teachers and colleagues for their guidance and support. We are thankful to our HOD, ECE Department. We also thank Principal& Chairman,KoneruLakshmaiah University for providing us most suitable environment for research and development.

REFERENCES:

[1] E. Grivel, M. Gabrea, and M. Najim, “Speech Enhancements a Realization Issue,” *Signal Processing*, vol. 82, pp. 1963–1978, Dec. 2002.

[2] M. Gabrea, E. Grivel, and M. Najim, “A single microphone kalman filter-based noise canceller,” *IEEESignal Processing Lett.*, vol. 6, pp. 55–57, Mar. 1999.

[3] J. D. Gibson, B. Koo, and S. D. Gray, “Filtering of colored noise for speech enhancement and coding,”*IEEE Trans. Signal Processing*, vol. 39, pp. 1732–1742, Aug. 1991.

[4] H. Morikawa and H. Fujisaki, “Noise reduction of speech signal by adaptive kalman filtering,” *Special issue in Signal Processing APII-AFCET*, vol. 22, pp. 53–68, 1988.

[5] M. Gabrea, “Robust adaptive kalman filtering based speech enhancement algorithm,” in *Proc.ICASSP’04*, 2004.

[6] M. Gabrea and D. O’Shaugnessy, “Speech signal recovery in white noise using an adaptive kalman filter, “in*Proc. EUSIPCO’00*, 2000, pp. 159–162.

[7] B. A. Anderson and J. B. Moore, *Optimal Filtering*, NJ:Prentice-Hall, Englewood Cliffs, 1979.

[8] B. D. Kovacevic, M. M. Milosavljevic, and M. Dj. Veinovic, “Robust recursive ar speech analysis,” *SignalProcessing*, vol. 44, pp. 125–138, 1995.

[9] R. K. Mehra, “On the identification of variances and adaptive kalman filtering,” *IEEE Trans. AutomaticControl*, vol. AC-15, pp. 175–184, Apr. 1970.

[10] T. Kailath, “An innovations approach to least squares estimation, part i: Linear filtering in additive white noise,” *IEEE Tran. Automatic Control*, vol. AC-13, pp. 646–655, Dec. 1968.