Predictive Coding of Speech Signals and Subjective Error Criteria

EE6110 Course Project Report

Abhishek Sekar
Roll No: EE18B067

December 24,2020

1 Introduction

Speech coding refers to the process by which we digitize a speech signal and represent it with reasonable quality in as few bits as possible.[4] Numerous applications in the telecommunications and the multimedia industry have made this field a very important application of Signal Processing.
Some noise is present in the recovered speech signal after coding due to errors associated with the whole process.
We observe that the presence of noise does not affect our perception of the speech spectrum pertaining to formant regions(the peaks in spectrum resulting from acoustic resonance of the human vocal tract) while it clearly affects the clarity of the portions of spectra where the speech intensity is low.
This report talks about how we can manipulate the noise spectrum so as to reduce its *perceptible* distortion.

2 Noise in a Predictive Coder

Given next page is the block diagram[1] of a Predictive Coder set up. Described below are the various stages of the process.

Transmitter Side:

- First off, the input speech signal s(t) is sampled to give us a series of samples and the s_n shown in the figure is the n^{th} sample.
- The Quantizer quantizes the input based on its parameters and this quantized output $\hat{q}_n$ is combined with ξ_n,the previous estimate from the predictor and is sent as the input $\hat{s}_n$ to the predictor.
- The difference in the source speech sample s_n and the prediction ξ_n is fed as the input to the Quantizer.

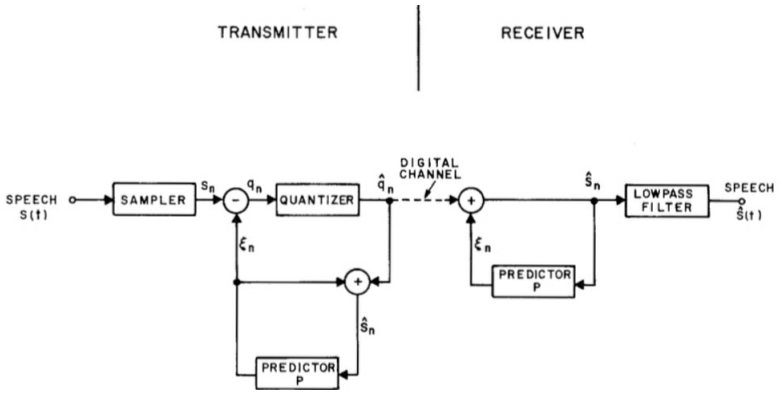

Figure 1: Schematic Diagram of a Predictive Coder

- **Receiver Side:** $\hat{q}_n$ is obtained and it goes through a similar predictor network as on the transmitter side and the regenerated sample is $\hat{s}_n$.

From the block diagram, we can try finding a relationship between the output sample $\hat{s}_n$ and the input s_n.

$$\hat{s}_n = \hat{q}_n + \xi_n \tag{1}$$

Now,taking the error introduced due to quantization as δ_n ($\hat{q}_n$ - q_n) and using it in the above equation, we have

$$\hat{s}_n = \delta_n + q_n + \xi_n$$

$$\mathrm{q}_n = s_n - \xi_n$$

$$\hat{s}_n = s_n + \delta_n \tag{2}$$

This tells us that, in the current set up, the error between the prediction and input is the same as that introduced by the quantizer. Assuming that the quantizer error is white[8], we have the noise spectra being constant for all frequencies.
Now let's see how we can change that.

3 Manipulating Noise Spectra

By assuming that the predictor is an M tap 'All Pole' filter with its k^{th} coefficient being a_k and elaborating on the expression of q_n we have,

$$\xi_n = \sum_{k=1}^{k=M} \hat{s}_{n-k} a_k$$

$$q_n = s_n - \sum_{k=1}^{k=M} \hat{s}_{n-k} a_k$$

Substituting for $\hat{s}_n$,

$$q_n = s_n - \sum_{k=1}^{k=M} s_{n-k} a_k - \sum_{k=1}^{k=M} \delta_{n-k} a_k \tag{3}$$

From this, we see that q_n can be split into two parts, namely, the error from prediction of s_n through filter P and the quantization error also filtered through filter P. One way to manipulate the error spectra is by allotting these two parts to different filters,ie, let the quantization error be filtered by another M tap filter,F.
Then we have,

$$q_n = s_n - \sum_{k=1}^{k=M} s_{n-k} a_k - \sum_{k=1}^{k=M} \delta_{n-k} b_k \tag{4}$$

Where b_k corresponds to the k_{th} coefficient of filter F.
Using the expression for $\hat{s}_n$, we can show that this can indeed be used to manipulate the error spectra.

$$\hat{s}_n = \delta_n + q_n + \xi_n$$

$$\hat{s}_n = \delta_n + s_n - \sum_{k=1}^{k=M} s_{n-k} a_k - \sum_{k=1}^{k=M} \delta_{n-k} b_k \sum_{k=1}^{k=M} \hat{s}_{n-k} a_k$$

$$\hat{s}_n - \sum_{k=1}^{k=M} \hat{s}_{n-k} a_k - (s_n - \sum_{k=1}^{k=M} s_{n-k} a_k) = \delta_n - \sum_{k=1}^{k=M} \delta_{n-k} b_k$$

Writing this in the Fourier domain, we have the following relationships between the transforms(expressed as capital letters)

$$\hat{S} - S = \Delta \frac{1-F}{1-P} \tag{5}$$

Therefore,this tells us that we can indeed change the noise spectra by implementing a filter F with different parameters when compared to P.

4 Choices of Filter F

Since Δ can be modeled as a white noise, the noise spectrum practically depends on $\frac{1-F}{1-P}$. However,the average power spectrum of the contribution $\frac{1-F}{1-P}$ is 0. The proof is given below.

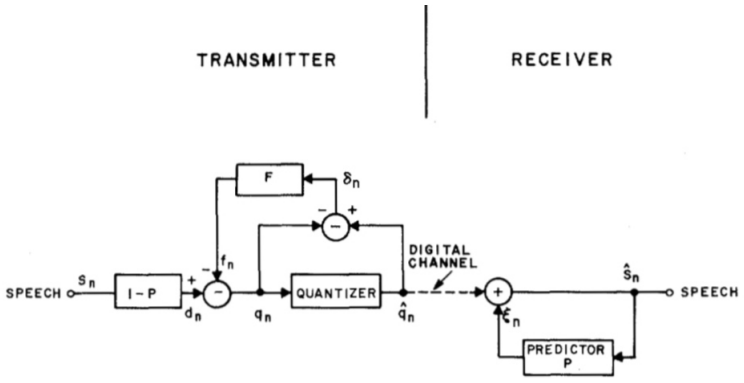

Figure 2: Schematic Diagram of a Predictive Coder discussed above[1]

$$1 - F(z) = \sum_{k=1}^{k=M} b_k z^{-k}$$

Expressing this as a product of it's roots r_k for k between 1 and M,

$$1 - F(z) = \Pi_{k=1}^{k=M}(1 - r_k z^{-1})$$

$$log(1 - F(z)) = \sum_{k=1}^{k=M} log(1 - r_k z^{-1})$$

Since the roots are inside the unit circle for a stable filter, we can write the logarithm as a polynomic function of z^{-1}.

$$log(1 - F(z)) = \sum_{n=1}^{n=\infty} k_n z^{-n} = \sum_{n=1}^{n=\infty} k_n e^{-2\pi j f T_n}$$

Where k_n is the sum of the n^{th} power of the roots.
From this we get the required relation,

$$\int_0^{fs} log(1 - F(e^{2\pi j f T}) df = \sum_{n=1}^{n=\infty} k_n \int_0^{fs} e^{-2\pi j f T_n} df = 0$$

Similarly,since Ps(z) is also an all pole filter we can show that the same relation holds for it as well. Therefore, since the average log power spectrum of $\frac{1-F}{1-P}$ is a linear combination of log(1-F(z)) and log(1-P(z)), it's average value is also 0.

Therefore,all that can be done is the redistribution of noise from a certain frequency level to another one. This property can be utilized to reduce the portions of noise in regions of the spectra where the intensity of the speech signal is low and redistribute it to formant regions of the spectra since their perception remains undisturbed even in the presence of additional noise,ie the noise is masked by the speech signal. This redistribution can be appropriately done by determining F by using a weighting across frequencies. From eqn(5), the only term we can control to redistribute noise is 1-F.We can therefore define our condition as the following minimization problem.

$$\min N_f = \int_{f=0}^{f=fs} G(f)^2 W(f) df \tag{6}$$

Where fs is the sampling frequency and

$$G(f) = (1 - F)^2$$

By taking into account that the contribution of the filters to the average logarithm of power spectrum of Noise is 0,we can find the G(f) that minimizes N_f. It satisfies the relation given below.

$$log(G(f)) = -log((W(f)) + \frac{1}{fs} \int_{f=0}^{f=fs} log(W(f) df \tag{7}$$

Therefore once we choose the weighting function, we can determine the filter coefficients of F.

Two extreme values of F

Sr.No.	W(f)	F
1	Constant	0
2	$\|1\text{-Ps}\|^{-2}$	Ps

5 Ingredients of a Generalized Predictive Coder for Speech Signals

Given next page is a generalized Predictive Coder for speech signals. As you can see in the figure,the setup is very similar to what we had before, except there's a new Prediction Filter Pd.The input speech signal is sent through a lowpass filter and sampled. The quantized information $\hat{q}_n$ is encoded at the transmitter side and decoded at the receiver side.Lastly, the estimated speech samples go through a de-emphasis filter to give rise to the output of the whole process.

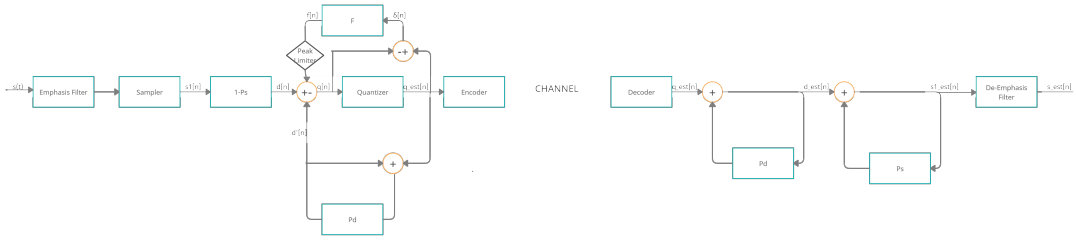

Figure 3: Schematic Diagram of a Generalized Adaptive Predictive Coder for Speech

5.1 Prediction using Short Time Spectral Envelope of Speech(Ps)

The predictor Ps is also called the formant predictor since it predicts the spectral envelope of the speech signal. It is of the form,

$$Ps(z) = \sum_{k=1}^{k=M} a_k z^{-k}$$

Where, z represents a delay of one sample (in Z transform notation). The number of taps M,depends on the bandwidth of the signal we're trying to estimate. It is roughly of the below form.[7]

$$M = 2B.W + 2(\text{or}4) \tag{8}$$

Therefore coding in the telecommunication industry where B.W is roughly 3.4 Khz we have M = 10 or 12. This predictor has to be adaptive since the speech signal changes rapidly. What's usually done in practice is to update the predictor coefficients every few *frames*(Small portions,typically a few ms long) of the speech signal while multiple frames of the speech signal is used to train the filter.[7] The coefficients of the predictor are chosen in such a manner that they reduce the MSE.The cost function can be expressed as,

$$J = E[e_n^2] = E[(s_n - \sum_{k=1}^{k=M} a_k s_{n-k})^2]$$

This resembles a M+1 dimensional bowl shaped surface with M degrees of freedom. Therefore this has a unique minima which we can find by differentiation. Differentiating with respect to a_i,

$$\frac{\partial J}{\partial a_i} = 2E[(s_n - \sum_{k=1}^{k=M} a_k s_{n-k})(-s_{n-i})] = 0$$

$$E[s_n s_{n-i}] = \sum_{k=1}^{k=M} a_k E[s_{n-k} s_{n-i}]$$

Denoting ϕ_{ij} as E$[s_{n-i} s_{n-j}]$, we have

$$\phi_{0i} = \sum_{k=1}^{k=M} a_k \phi_{ki}$$

Expressing this in Matrix form leads us to the normal equation.[3]

$$\Phi a = \Psi \tag{9}$$

Where, ψ[i] is ϕ[0i]. This can be solved recursively in so many different ways. Given next page is one such method which uses the properties of the Covariance matrix Φ.[5]

The reason this method is preferred is because its computational complexity is very low. Another good technique uses the Cholesky Decomposition of the covariance matrix and proceeds by calculating reflection coefficients.[2]

The Levinson Durbin Recursion Algorithm

- Initialization: l=0,$J_0 = \psi[0]$
- Recursion for l :1,2,...,M
  - Evaluating Reflection Coefficient.

$$k_l = \frac{1}{J_{l-1}}(\psi[l] + \sum_{i=1}^{i=l-1} a_i^{(l-1)}\psi[l-i]) \tag{10}$$

    Note: a_i is not raised to (l-1), it is the value of a_i at the $(l-1)^{th}$ iteration.
  - Linear Predictor Coefficients for l^{th} order predictor.

$$a_l^{(l)} = -k_l a_i^{(l)} = a_i^{(l-1)} - k_l a_{l-i}^{(l-1)} \text{for i:1,2,..,l-1} \tag{11}$$

  - MMSE at l^{th} iteration

$$J_l = J_{l-1}(1-k_l^2) \tag{12}$$

    From this,we can observe that $|k|$ has to be less than 1 for stability or equivalently all poles of the predictor has to lie inside the unit circle which matches our assumption before.
- Output : -a and Jmin

It was observed that the if we find the coefficients in the above manner then we get a very high Power Gain for the filter Ps.[1] This is because, the regeneration filter roughly resembles the reciprocal of the speech spectrum(differing by a scaling constant). The low pass filter we use cuts the speech spectrum off and therefore due to this, at the cutoff frequency 1-Ps(z) takes a very high value. Since

$$\text{Power Gain} = \int [1 - Ps(e^{2\pi j f T}]^2 df$$

it ends up taking a very high value.The missing components of speech spectrum cutoff by the lowpass filter end up contributing towards very low eigenvalues in Ps and this is another explanation as to why its Power Gain is too high.
This is fixed by the below regularization,

$$\hat{\phi[ij]} = \phi[ij] + \lambda e_{min}\mu_{i-j} \tag{13}$$

$$\hat{\psi[i]} = \psi[i] + \lambda e_{min}\mu_i \tag{14}$$

Where the terms with the hat represent the regularized matrix elements.λ is a constant between 0.01 and 0.10 while e_{min} corresponds to the MMSE computed without regularization. μ_i refers to the autocorrelation of white noise samples i samples apart passed through a high pass filter(inverse filter of the low pass filter chosen usually). For the choice of the high pass filter given below,

$$[\frac{1}{2}(1 - z^{-1})]^2$$

μ_i was observed to be,

Values of μ_i

μ_i	Value
μ_0	$\frac{3}{8}$
μ_1	$\frac{-1}{4}$
μ_2	$\frac{1}{16}$
μ_k k >2	0

5.2 Prediction using the Periodic Nature of Voiced Speech(Pd)

The adjacent pitch periods in voiced speech show considerable similarity. This quasi periodic nature observed in speech persists albeit to a lower extent in the difference signal d_n obtained after prediction through Ps. Thus the redundant features observed in d_n can be further removed by using the filter Pd. This will reduce the channel capacity which improves the speed of the coder. The expression of a first order pitch predictor with three predictor coefficients is given below.

$$Pd(z) = \beta_0 z^{-M} + \beta_1 z^{-M-1} + \beta_2 z^{-M-2} \tag{15}$$

Where M is a constant determined by computing the maximum correlation coefficient between d_n and d_{n-M} across multiple values of M. M essentially is of the order of 2 to 20ms and is usually a multiple of the pitch period of the voiced speech signal. Once M is obtained, the filter coefficients can be easily estimated.[9] The expression for β_i is given below.

$$\beta_i = \frac{\phi[0][M+i]}{\phi[M+i][M+i]} \tag{16}$$

Here $\phi[i][j]$ refers to the correlation between d_{n-i} and d_{n-j}.

5.3 The Quantizer

Representation of a large set of elements with a much smaller set of elements is called quantization.They are also called Analog to Digital converters.This representation ensures that fewer data has to be stored or transmitted,which helps towards a cost efficient solution. However, quantization introduces errors which are undesirable and therefore a quantizer has to be chosen appropriately

in a manner minimizing these errors and also taking into account the trade-off between coder complexity and increasing the number of quantizer levels to minimize this error. There's two major types of quantizers applied in Speech Coding and they're uniform quantizers and optimal non-uniform quantizers.[6]

5.3.1 Uniform Quantizers

As the name suggests, the quantizer levels are distributed uniformally. Let y_i for i:1,2,..,N be the quantizer outputs or the *Codebook* elements and similarly x_i the input samples in ascending order.Then we have,

$$y_{i+1} - y_i = \Delta \text{ for i: } 1,2,...,\text{N-1} \tag{17}$$

$$y_N = x_{N-1} + \frac{\Delta}{2} \tag{18}$$

Where Δ is a constant called the *step size* of the uniform quantizer.

$$\Delta = \frac{\max(x) - \min(x)}{N} \tag{19}$$

Uniform Quantizers are good when the input sample is somewhat uniform or linear as well.[8]Hence they are used for speech signals for small frame sizes(when they are without rapid changes in the spectrum).

5.3.2 Optimal Quantizers

This school of quantizers try minimizing the mean square error between the input and the quantized output to the maximum extent. The codebook is estimated in compliance with two major conditions,namely the nearest neighbour condition and the centroid condition.[6]

- **Nearest Neighbour Condition:** Basically the input gets assigned to the element of the codebook which is the closest (in terms of distance or minimum error) to it. The set of elements are grouped into cells partitioned by which codebook element it gets mapped to.

  $$R_i = \{x : d(x, y_i)\text{is the smallest distance}\} \tag{20}$$

  R_i is the optimum partition cell corresponding to element y_i of the codebook.

- **Centroid Rule:** The definition of a centroid is as follows, the centroid of R_i is y_i provided it exists. They can be computed if we know the probability distribution of the parent random variable X of the input.

  $$y_i = \frac{\int_{R_i} x f_x dx}{\int_{R_i} f_x dx} \tag{21}$$

The algorithm which finds the codebook of the optimal quantizer is called the Lloyd Max Algorithm.[6]
When dealing with speech signals, we don't know the statistics of the signal and therefore we implement an adaptive variant of the Lloyd Max Algorithm a type of the K-Means algorithm.

Adapted Lloyd Max Algorithm as a Recursion

- **Initialization:** Elements of the initial codebook are selected randomly from the given set of inputs, the training dataset.
- **Recursion till the MSE converges within a tolerance level**
  - For the given codebook Y_m using the nearest neighbour condition partition the input data into optimal partition cells.
  - Using the optimal partition cells,implement the Centroid condition to find an updated value of the codebook. Since the statistics of the input is unknown, the centroid of the cell can be approximated by the mean of the elements of the cell.

    $$y_i = \frac{1}{N} \sum_{x_k \in R_i} x_k \tag{22}$$

    Where N is the number of elements in the cell.
  - Compute the Mean Square Error for the updated codebook and continue the recursion if the new error isn't within a tolerance level of the older error.

5.4 Revisiting F

It was observed that the predictive coder set up was unstable when 1-F was not a minimum phase transfer function.[1] To improve stability, a peak limiter was brought into the set up which prevented f_n from going higher than twice the rms error in the prediction of the speech samples.
Coming to the potential values of F, something between the two extreme values discussed in the last section could be chosen for F.

$$F = \alpha z^{-1} Ps \tag{23}$$

$$F = \alpha Ps \text{ where } |\alpha| < 1 \tag{24}$$

Given above are two potential choices.They might not be the most optimal choices of F. The optimal choice of F should be determined by extensive experimentation on different speech signals.

6 Simulations:Discussion and Results

6.1 Experimental Setup Specifics

The experimental setup is the same as the adaptive predictive coder shown in figure 3.The figure is shown again below for reference.

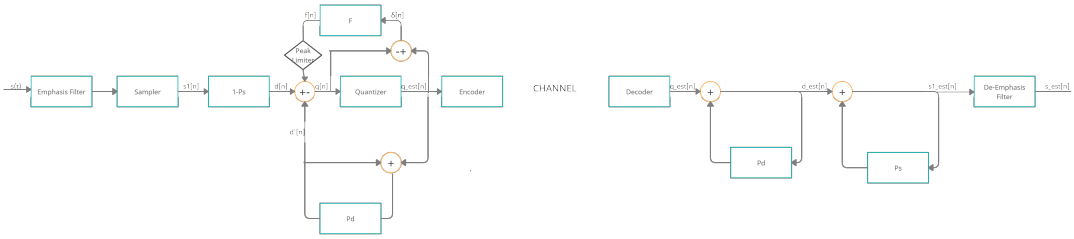

Figure 4: Schematic Diagram of a Generalized Adaptive Predictive Coder for Speech

The simulation was attempted in python.Firstly, audio signals (in the .wav format) of duration roughly 1 second was recorded at the audio sampling rate given in the table. This was further sampled at $\frac{1}{6}$ the original sampling frequency which was also above the nyquist rate for telecommunications. This was then sent through the pre-emphasis filter and the feedback loop comprising Pd,F and the Quantizer was processed sample by sample. The filter and the Quantizer parameters were updated every 10ms. The filter parameters were found using the algorithms discussed before and as for the Quantizer, a gaussian white noise with variance equal to the rms error of the previous cycle's prediction was used as the training data. The Lloyd Max Algorithm was implemented using K-Means in python with a very low tolerance level.

Specifications of the experimental setup

Sr.No.	Variable/Filter	Value
1	Audio Sampling Rate	44.1 KHz
2	Sampling Rate for the coder	7.35 KHz
3	Emphasis Filter	1-0.4z^{-1}
4	De-Emphasis Filter	(1-0.4z^{-1})$^{-1}$
5	Ps taps	12
6	λ regularization	0.05
7	F filter	Ps*0.5 z^{-1}
8	Pd order	3
9	Quantizer Model	Optimal Quantizer 4 levels
10	Quantizer Tolerance	10^{-5}
11	Updating Period	10ms

Note:The setup here is very different compared to the original paper's setup.

6.2 Plots and Discussion

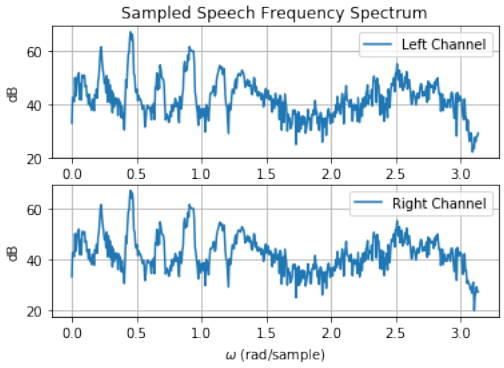

Figure 5: Frequency Spectrum of Sampled Audio for the word "Hello"

The left and right channels refer to the two parts to the audio signal(what we'd essentially hear in the left and right side whilst using a headphone). This is automatically done by the recording software to improve spatial emulation of sound.

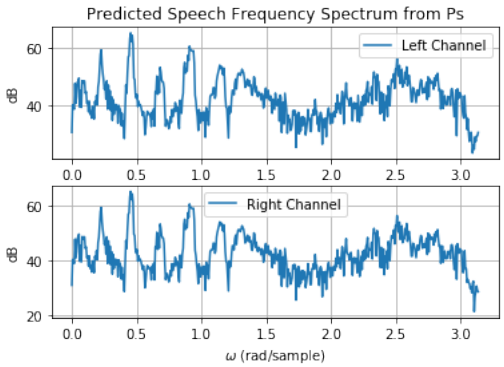

Figure 6: Predicted Speech Spectrum using Ps for the word "Hello"

This plot depicts the predicted speech signal using just the formant predictor Ps. From the plot, it's clear that for the prediction is very close to the original signal.

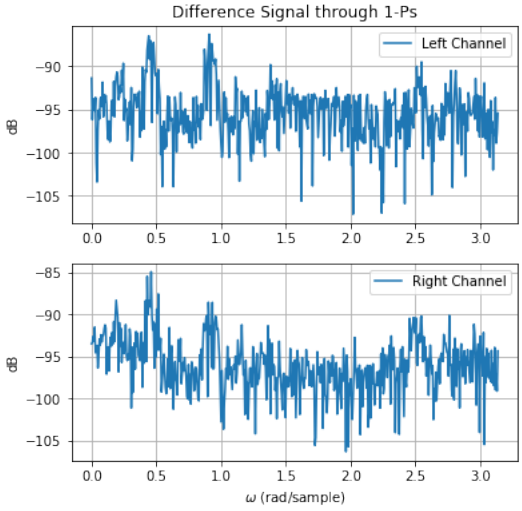

Figure 7: Difference Signal Spectrum through 1-Ps for the word "Hello"

The difference signal takes very low values to the tunes of -100dB which further reiterates the performance of the predictor with respect to speech frames in its domain.

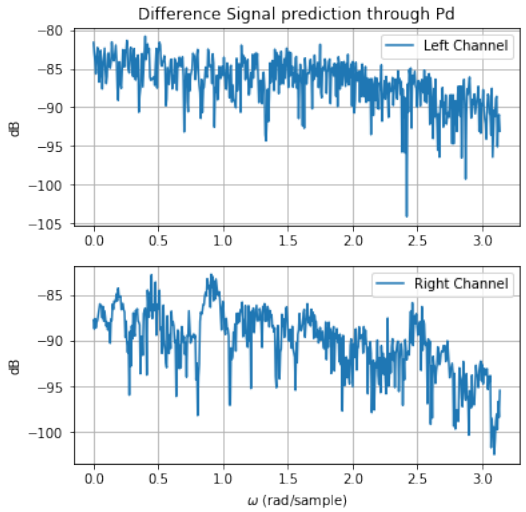

Figure 8: Difference Signal Spectrum predicted using Pd for the word "Hello"

This plot shows that the prediction made by the pitch predictor Pd isn't good since there is a lot of variation between this and the previous plot.

Discussion 1

- From these plots it is clear that the spectral envelope is predicted well while the pitch predictor doesn't perform as well.
- The good prediction of the spectral envelope can be attributed to the way it was done. The predictor was trained using speech frames of size 5ms.(The period of update). Due to this, when the input signal was sent into the predictor, since a large portion of the input directly corresponded to the training data, the prediction ended up being very good.
- The bad prediction of the pitch predictor could be due to the following.
  - The pitch predictor fares well only for the voiced portions of the speech signal (the portion of speech produced by the vocal tract) while the input signal is a mix of voiced and unvoiced speech(background noise)[5]
  - The training data of the filter consisted of samples within a 5ms timeframe.This might not have been larger than the period of the voiced portions leading to incorrect prediction.
  - With the difference signal reaching very low values, the computation might not have been sensitive enough leading to incorrect prediction.

- The presence of λ normalization indeed reduced the average power gain of the predictor Ps to less than 2dB.
- Given below are the prediction gains for both the filters(average across left and right channels over multiple experiments)

Prediction Gains

Sr.No.	Filter	Gain
1	Ps	67.86 dB
2	Pd	-26.34 dB

The prediction gain is computed as

$$P.G = 10log(\frac{P_i}{P_e}) \tag{25}$$

Where P_s and P_e refer to the power of the input signal and the prediction error respectively.

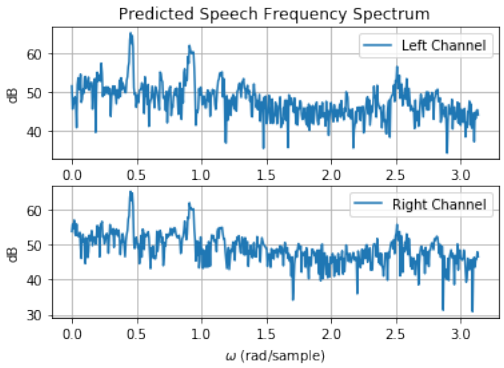

Figure 9: Predicted Frequency Spectrum for the word "Hello" using 3-bit PCM

This plot shows the predicted speech spectrum when it is predicted just using an 8 level optimal quantizer without any adaptive prediction filters, ie the predicted output is essentially $\hat{q}$.

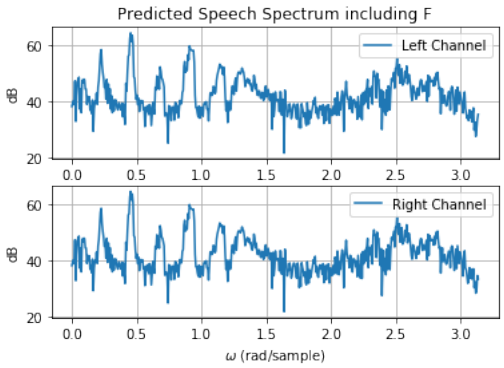

Figure 10: Predicted Frequency Spectrum for the word "Hello" using filter F

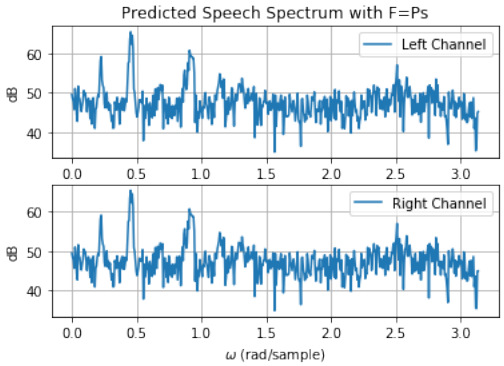

Figure 11: Predicted Frequency Spectrum for the word "Hello" using filter F=Ps

Discussion 2

- Note: The three plots were not shown in the same graph since the spectrums had a lot of overlap despite the vast difference in the SNR.
- The first plot is just for comparison purposes and to point out the areas in which the predictive coder performs better than a conventional PCM (just quantizer) coder.As per the original paper,their results stated that the predictive coder performs better than a 7-bit log PCM coder. In this plot, we see some of the drawbacks in a PCM coder, namely abundant noise spread almost uniformally across the whole spectrum.Comparing this with the original speech spectrum, we observe that the formant peaks are captured well but the low intensity portions of speech are corrupted by noise.
- **Comparison between the two predictive coder plots:** The output signals were processed to obtain an approximate SNR of about 30dB which

is around what was obtained in the original paper.

  - When the filter F is used we observe that the output is very similar to the input signal.It almost looks like the input signal has been scaled down. The regions with lesser intensity are also captured pretty well with reasonable accuracy.
  - When the filter F is absent and Ps is used instead, we observe that the formant peaks are sharper and much closer to the original speech spectrum. However, this comes at the cost of more noise at the lower intensity regions.
  - These graphs confirm what we had expected in the earlier part of this report and F indeed reduces the perceptible noise.

- Note: The graphs obtained do not exactly correspond to the outputs that I had gotten. The SNR of the predictor in my simulation did not end up being good enough (around -1 dB) to observe the differences between the plots with and without F.This could be due to the following reasons.

  - Generally, a larger set of training data (typically 30 - 60ms) is used for prediction.[7] I could not use something this big due to computational constraints. Also the input audio was too small for this training data size, this again,could not be larger owing to computational constraints.
  - Speech coding is done in a frame by frame manner.[7] However, I implemented it in a sample by sample manner.This coupled with the previous reason, might have led to incorrect predictions from both Ps and Pd since in the recursion, signals from the previous time frames also came into contact with predictors of the present time frame. The predictors might not have fared well for these cases owing to the small training data size.
  - The extremely poor performance of Pd might have dominated the predictor leading to overall poor prediction of the output.
  - However, despite these negatives, the predicted speech is somewhat decipherable when played.

- Given below are the overall SNR(averaged across left and right channels) of the three predictions.

Signal to Noise ratio of the Coders

Sr.No.	Predictor Type	SNR
1	3 bit PCM	10.46 dB
2	Adaptive Predictor with F	32.10 dB
3	Adaptive Predictor without F	30.54 dB

The coder SNR is computed in the same manner as the prediction gain.

- The Quantizer performs well in all three cases, with its gain being around 10dB.

7 Conclusion

The processed plots confirm the result obtained in the original paper, that the perceptible noise can be reduced with the presence of an additional filter F. The original paper[1] came out sometime in the year 1978, just when the ideas regarding predictive coding of speech were taking form. This paper at that time, was quite revolutionary and helped in making good progress in this field. Even now, roughly 40 years after the paper was published and with the advent of VoIP and advanced predictive coders, the theory expressed in this paper is still largely relevant.
This versatility of the Adaptive Predictive Coders and the lesser channel capacity have made them very popular over basic PCM coders despite their higher computational complexity.

References

[1] B. Atal and M. Schroeder. Predictive coding of speech signals and subjective error criteria. In *ICASSP '78. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 573–576, 1978.

[2] B. S. Atal and Suzanne L. Hanauer. Speech analysis and synthesis by linear prediction of the speech wave. *The Journal of the Acoustical Society of America*, 50(637), 1971.

[3] B. S. Atal and M. R. Schroeder. Adaptive predictive coding of speech signals. *The Bell System Technical Journal*, 49(8):1973–1986, 1970.

[4] Wai . C Chu. *Introduction,Speech Coding Algorithms*, chapter 1, pages 1–32. John Wiley Sons, Ltd, 2003.

[5] Wai . C Chu. *Linear Prediction,Speech Coding Algorithms*, chapter 4, pages 91–142. John Wiley Sons, Ltd, 2003.

[6] Wai . C Chu. *Scalar Quantization,Speech Coding Algorithms*, chapter 5, pages 143–160. John Wiley Sons, Ltd, 2003.

[7] Mark Hasegawa-Johnson and Abeer Alwan. Speech coding: Fundamentals and applications. *Encyclopedia of Telecommunications*, 2003.

[8] J. Max. Quantizing for minimum distortion. *IRE Transactions on Information Theory*, 6(1):7–12, 1960.

[9] R. P. Ramachandran and P. Kabal. Pitch prediction filters in speech coding. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(4):467–478, 1989.