Predictive Coding of Speech Signals and Subjective Error Criteria

EE6110:Adaptive Signal Processing Project

Abhishek Sekar
EE18B067

Indian Institute of Technology, Madras

December 2020

RoadMap

- A foundation for Speech Coding
- Manipulating the Error Criteria
- Ingredients of a generalized Adaptive Predictive Coder
- Simulation Results and Observation

Speech Coding

- **What :** 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]

Speech Coding

- **What :** 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]
- **Where :**Primarily Telecom and Multimedia industries

Speech Coding

- **What :** 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]
- **Where :**Primarily Telecom and Multimedia industries
- **Why :**One sample = 2 bytes , bitrate in a phone call is approximately 128Kbps !!
  Speech coding can remove redundancy and could potentially get it down to as low as 2.4Kbps.
  A factor of 53 times!

The Problem Statement

- Predictive coding introduces a prediction error which is realized as noise.
- We observe that the presence of this 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.

A basic predictive coder

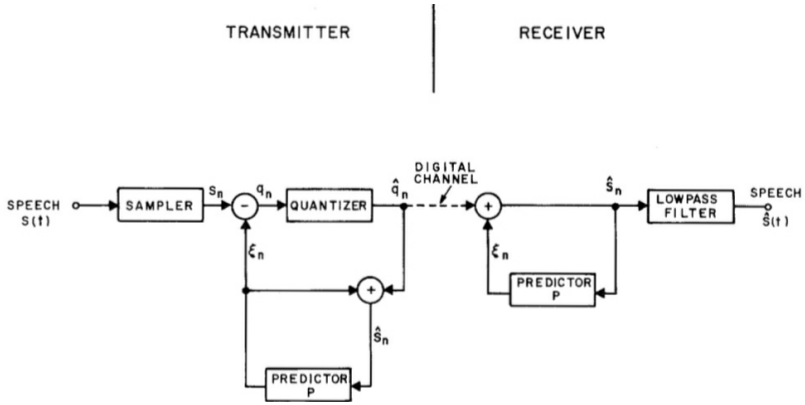


Figure: Schematic Diagram of a Predictive Coder[1]

Finding an expression for the prediction error:

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

$$\delta_n = \hat{q}_n - q_n$$

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

$$q_n = s_n - \xi_n$$

$$\hat{s}_n - s_n = \delta_n \tag{1}$$

A basic predictive coder

$$\hat{s}_n - s_n = \delta_n = \hat{q}_n - q_n$$

This tells us that the Error due to Quantization = Prediction Error.
The Quantization error is modelled as a white noise[8] and therefore, we have a constant noise intensity across frequencies.
How do we change that?

Manipulating the Error Criteria

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 $\hat{s}_n$,

$$q_n = s_n - \sum_{k=1}^{k=M} s_{n-k} a_k - \sum_{k=1}^{k=M} \delta_{n-k} a_k \tag{2}$$

Manipulating the Error Criteria

Replace one part with another M tap 'All Pole' filter F with its k^{th} coefficient being b_k

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

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

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

Writing this in the Fourier domain,

$$\hat{S} - S = \Delta \frac{1 - F}{1 - P} \tag{4}$$

F can be used to manipulate the error spectrum.

Schematic Diagram of the Modified Predictive Coder

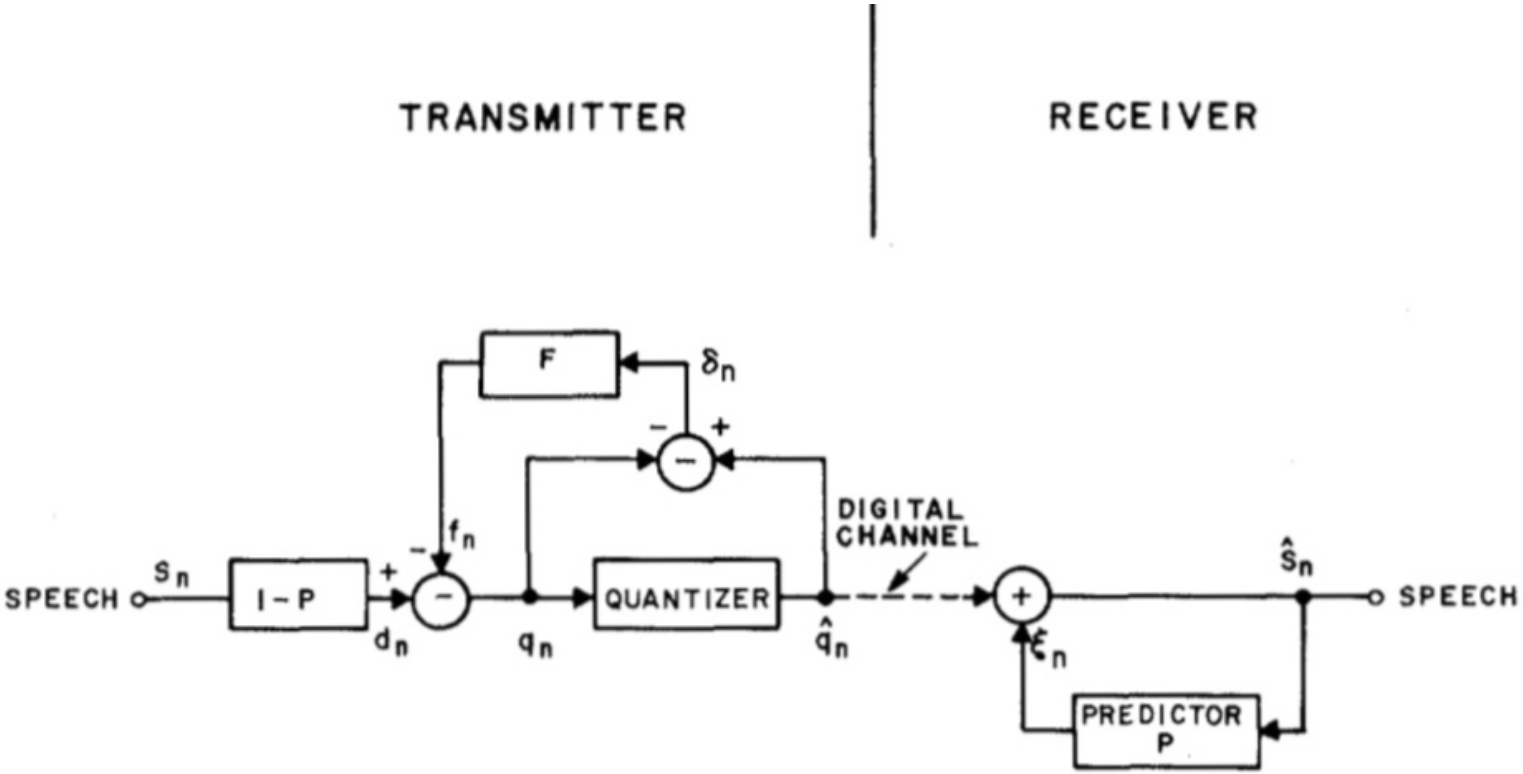


Figure: Schematic Diagram of a Predictive Coder discussed above[1]

Redistribution of Noise

With the introduction of F, we cannot reduce average noise power.

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

Expressing this as a product of its 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}.

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

Redistribution of Noise

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

$$\int_0^{fs} log(1 - F(e^{2\pi jfT})df = \sum_{n=1}^{n=\infty} k_n \int_0^{fs} e^{-2\pi jfT_n} df = 0$$

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)), its average value is also 0.
The Noise spectrum power can only be redistributed and not reduced.

Redistribution of Noise

The formant peaks in the speech spectrum mask the noise.
Hence we can remove noise from the parts of the speech spectrum with low intensity and redistribute it to the peaks.

Figure: Robin Hood

F is like the Robin Hood of filters!

Appropriate F

The noise redistribution can be done by determining F by using a weight across frequencies.

$$\min N_f = \int_{f=0}^{f=fs} G(f)^2 W(f) df \tag{5}$$

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.

$$log(G(f)) = -log((W(f)) + \frac{1}{fs} \int_{f=0}^{f=fs} log(W(f) df \tag{6}$$

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

Appropriate F

Two extreme values of F

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

Where Ps is the predictor. F is chosen between these two extremes.

$$F = \alpha z^{-1} Ps \tag{7}$$

$$F = \alpha Ps \text{ where } |\alpha| < 1 \tag{8}$$

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.
In practice, it has to be ensured that 1-F is a minimum phase transfer function.

A generalized Adaptive Predictive Coder

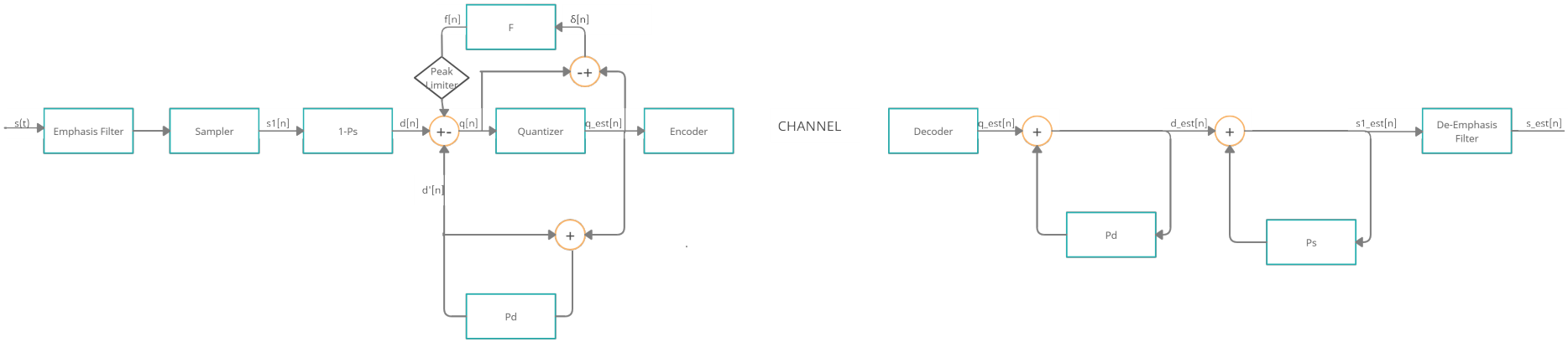


Figure: Schematic Diagram of a Generalized Adaptive Predictive Coder for Speech

Ps:Prediction using Short Time Spectral Envelope

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

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

The number of taps M,depends on the bandwidth of the signal we're trying to estimate. [7]

$$M = 2B.W + 2(\text{or}4) \tag{9}$$

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]

Estimating the Coefficients

The coefficients of the predictor are chosen in such a manner that they minimize the MSE.

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

This has a unique minima. Differentiating with respect to a_i,

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

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

Estimating the coefficients

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]

$$\Phi a = \Psi \tag{10}$$

Where, ψ[i] is ϕ[0i]. This can be computed recursively.[5, 2].

Levinson Durbin Recursion

The Levinson Durbin Recursion Algorithm

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

$$k_l = \frac{1}{J_{l-1}}(\psi[l] + \sum_{i=1}^{i=l-1} a_i^{(l-1)}\psi[l-i]) \tag{11}$$

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.

$$a_l^{(l)} = -k_l a_i^{(l)} = a_i^{(l-1)} - k_l a_{l-i}^{(l-1)} \text{ for i:1,2,..,l-1} \tag{12}$$

  - MMSE at l^{th} iteration

$$J_l = J_{l-1}(1 - k_l^2) \tag{13}$$

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

Regularization

It was observed that Ps computed in the above fashion had a large power gain.[1]
This is because, the regeneration filter roughly resembles the reciprocal of the speech spectrum.
The lowpass filter cuts the speech spectrum off and due to this, at the cutoff frequency 1-Ps(z) takes a very high value.

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

The missing components of speech spectrum cutoff by the lowpass filter end up contributing towards very low eigenvalues in Ps.
This is fixed by the below regularization,

$$\hat{\phi[ij]} = \phi[ij] + \lambda e_{min} \mu_{i-j} \tag{14}$$

$$\hat{\psi[i]} = \psi[i] + \lambda e_{min} \mu_i \tag{15}$$

Regularization

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

Pd:Prediction using the Periodic Nature of Voiced Speech

The 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. A 3 Tap Pitch Predictor is shown below.

$$Pd(z) = \beta_0 z^{-M} + \beta_1 z^{-M-1} + \beta_2 z^{-M-2} \tag{16}$$

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.

$$\beta_i = \frac{\phi[0][M+i]}{\phi[M+i][M+i]} \tag{17}$$

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

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 tradeoff between coder complexity and the number of quantizer levels to minimize this error.

Types of Quantizers:Uniform Quantizers

There's two major types of quantizers applied in Speech Coding and they are uniform quantizers and optimal non-uniform quantizers.[6]

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,

$$y_{i+1} - y_i = \Delta \text{ for i: } 1,2,\ldots,\text{N-1} \tag{18}$$

$$y_N = x_{N-1} + \frac{\Delta}{2} \tag{19}$$

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

$$\Delta = \frac{\max(x) - \min(x)}{N} \tag{20}$$

Uniform Quantizers are good when the input sample is somewhat uniform.[8]Hence they are used for speech signals for small frame sizes.

Types of Quantizers: 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.

$$R_i = \{x : d(x, y_i) \text{is the smallest distance}\} \tag{21}$$

  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.

$$y_i = \frac{\int_{R_i} x f_x dx}{\int_{R_i} f_x dx} \tag{22}$$

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

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.

$$y_i = \frac{1}{N} \sum_{x_k \in R_i} x_k \tag{23}$$

    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.

Experimental Set Up

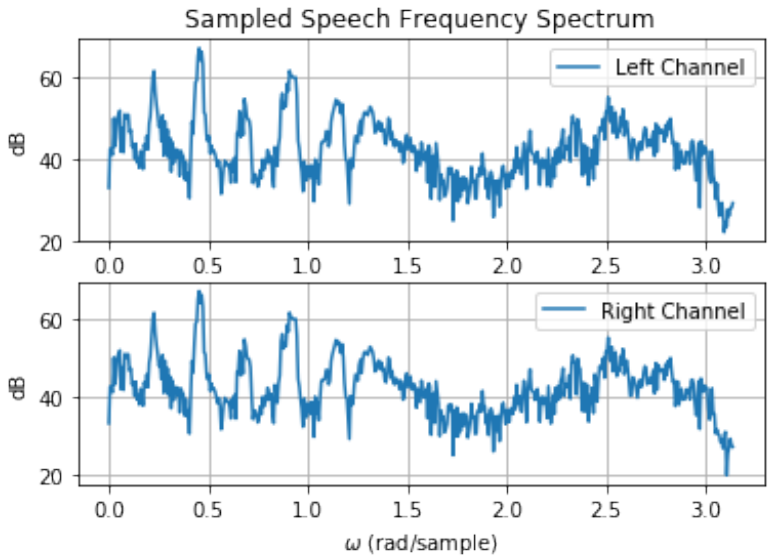
The simulation was attempted in python.Firstly, audio signals (in the .wav format) of duration roughly 1 second was recorded at the audio sampling rate. The filter parameters were found using the algorithms discussed before and as for the Quantizer, a gaussian white noise with variance equal to the rms 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

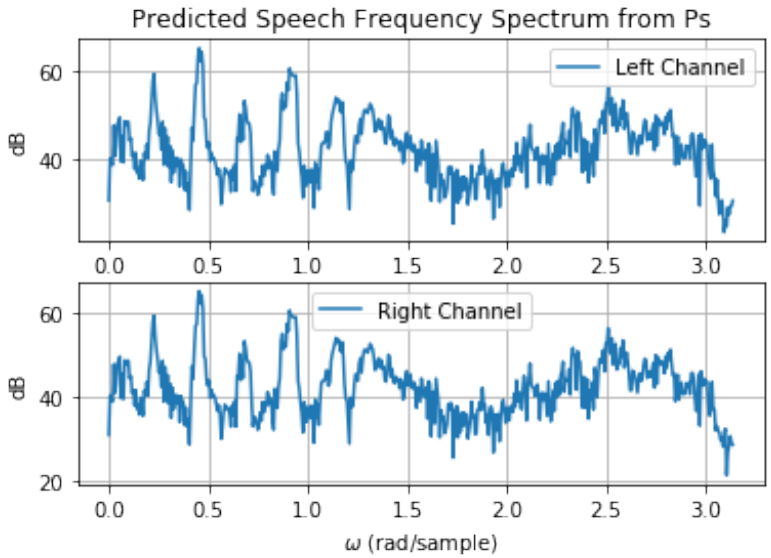
Sr.No.	Variable/Filter	Value
1	Audio Sampling Rate	44.1 KHz
2	Sampling Rate for the coder	7.35 KHz
3	Emphasis Filter	1-0.4z^{-1}
4	De-Emphasis Filter	(1-0.4z^{-1})$^{-1}$
5	Ps taps	12
6	λ regularization	0.05
7	F filter	Ps*0.5 z^{-1}
8	Pd order	3
9	Quantizer Model	Optimal Quantizer 4 levels
10	Quantizer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

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

Initial Plots



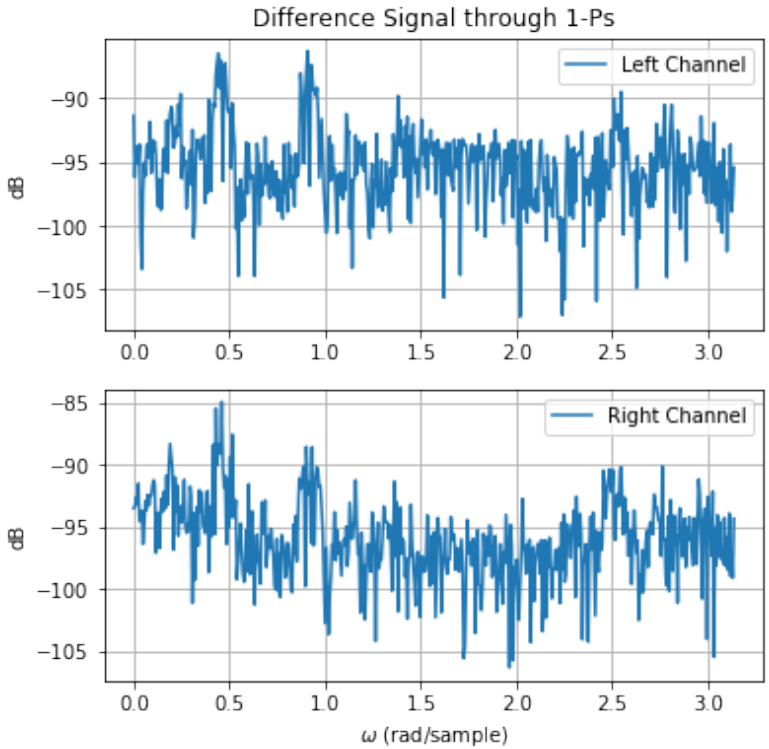
(a) Samples



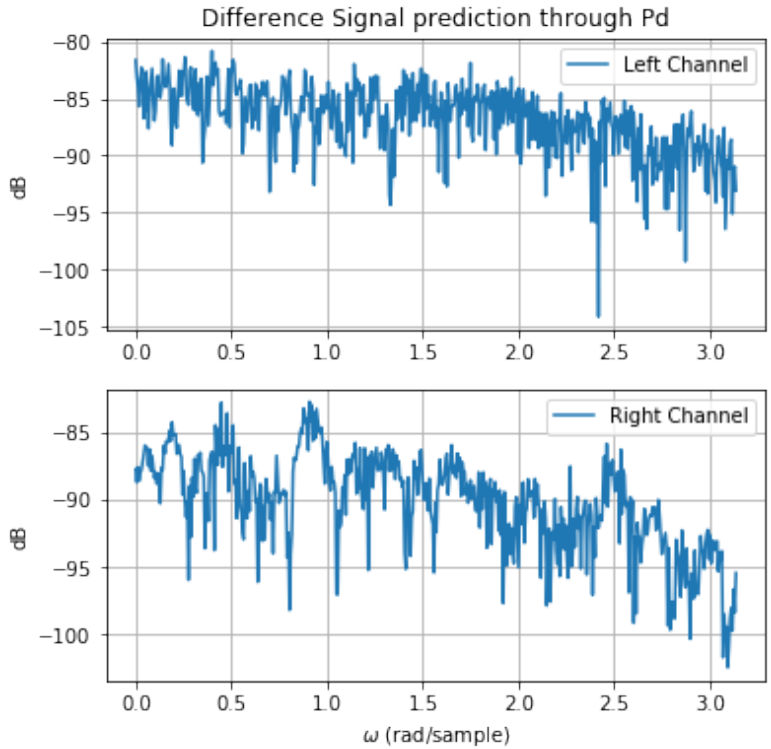
(b) Prediction

Figure: Prediction through Ps for the word "Hello"

Initial Plots



(a) Difference



(b) Prediction

Figure: Prediction through Pd for the word "Hello"

Initial Discussion

- 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. Due to this, as a large portion of the input (for a period) 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.

Performance of the Filters

- 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(averaged across left and right channels over multiple experiments)

Prediction Gains

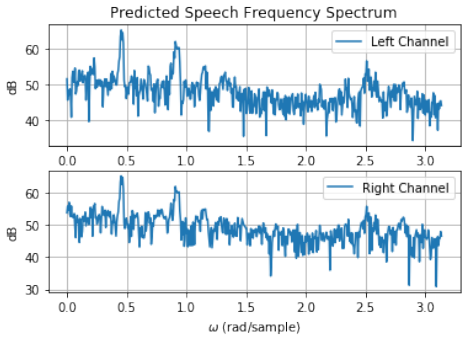
Sr.No.	Filter	Gain
1	Ps	67.86 dB
2	Pd	-26.34 dB

The prediction gain is computed as

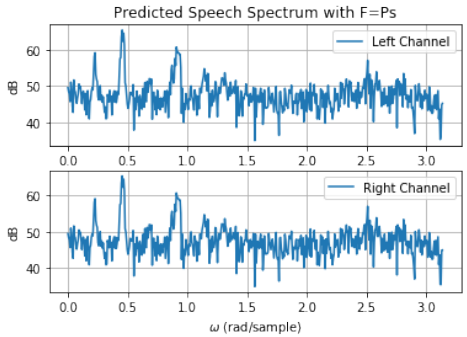
$$P.G = 10log(\frac{P_i}{P_e}) \tag{24}$$

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

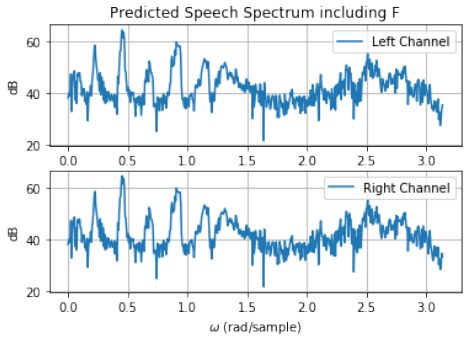
Final Plots



(a) 3 Bit PCM



(b) Without F



(c) With F

Figure: Performance of the Full Coder for the word "Hello"

Final Discussion

- In the first plot, we see some of the drawbacks in a PCM (just quantizer) 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.

Performance of the Full Set Up

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

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.

Thank You