M-ARY PREDICTIVE CODING: A NONLINEAR MODEL FOR SPEECH

M.Vandana

Final Year B.E(ECE)
PSG College of Technology
Coimbatore-641004.
vandana_mohan@indiatimes.com

ABSTRACT
Speech Coding is pivotal in the ability of networks to support multimedia services. The technique currently used for speech coding is Linear Prediction. It models the throat as an all-pole filter i.e. using a linear difference equation. However, the physical nature of the throat is itself a clue to its nonlinear nature. Developing a nonlinear model is difficult as in the solution of nonlinear equations and the verification of nonlinear schemes.

In this paper, two nonlinear models– Quadratic Predictive Coding and M-ary Predictive Coding– have been proposed. The equations for the models were developed and the coefficients solved for. MATLAB was used for implementation and testing. This paper gives the equations and preliminary results. The proposed enhancements to MPC are also discussed in brief.

1. AIM

The objective of this endeavour was to
i. Investigate the two nonlinear models proposed by the author for more effective speech coding.
ii. Development of one of the two models – MPC (M-ary Predictive Coding) – as a complete coding scheme.

2. INTRODUCTION

Speech is a very special signal for a variety of reasons. The most preliminary of these is the fact that speech is a non-stationary signal. This makes speech a rather tough signal to analyze and model.

The second reason, is that factors like the intelligibility, the coherence and other such ‘human’ characteristics play a vital role in speech analysis as against statistical parameters like Mean Square Error.

The third reason is from a communications point of view. The number of discrete values required to describe one second of speech amounts to 8000 (at the minimum). Due to concerns of bandwidth, compression is desirable.

2.1. Linear Predictive Coding

In the technique of linear prediction, the vocal tract is modeled as an all-pole filter[1]. That is, the transfer function of the vocal throat is written as

\[ H(z) = \frac{1}{\sum_{i=1}^{p} a_i z^{-i}} \]

In terms of a difference equation,

\[ y[n] = \sum_{i=1}^{p} a_i y[n-i] \]
This means that each sample of the sequence describing speech is assumed to be a linear weighted combination of a finite number of previous samples of the sequence.

By the method of minimization of Mean Square Error, the coefficients for the linear predictive filter \(a_i\) can be determined[1]. This is the method currently used for speech coding (in the form of various implementations like CELP, RELP etc.).

A look at the shape of the human throat reveals that the throat (vocal tract) is not linear. It is curved. The obvious question now is why the throat has been modeled as an all-pole filter (linear) when in reality is not linear. This is because the assumption of linearity in the throat, highly simplifies the required calculations. The final system that is required to be solved is simply a set of linear, simultaneous equations, various methods to solve which (like the Levinson Durbin algorithm) are widely known.

However, a simple implementation of LPC yields errors in the speech output at the Rx. Hence a nonlinear model was studied.

With the objectives of error minimization, intelligibility retention and computational simplicity in mind, in this project, the attempt to develop two distinct nonlinear models for speech was carried out. Both these models were investigated. Both the models were given mathematical expressions, and equations were developed for the respective parameters. MATLAB was then used to code each of these models and test them for the above-mentioned parameters. Various deductions were then drawn from the models and the results. These form the remainder of the paper.

3. NONLINEAR MODELS PROPOSED BY THE AUTHOR

3.1. Quadratic Model for speech

In this model, as against each sample being expressed as a linear weighted sum of a finite number \(p\) values preceding it, it was attempted to express each sample as a quadratic weighted sum of the same.

Preliminary equation for quadratic model:

\[
p \sum_{i=0}^{p} (a_i y[n-i] + b_i y^2[n-i])
\]

3.2. M-ary Model for speech

In this model, the current sample of the sequence is modeled as being a weighted sum of ‘m’ previous sample values, with the power of each sample value increasing with the delay of current sample from it. This attempts to model the physical fact that the longer the time elapsed since the occurrence of a sample value, the lesser its effect, i.e. the decay of each sample value with progressive increase of time.

Preliminary equation for M-ary model:

\[
m \sum_{i=0}^{m} (a_i y[n-i])
\]
4. QUADRATIC MODEL

4.1. Development of Equations

The preliminary equation as given above is

\[ p_y[n] = \sum_{i=1}^{p} (a_i * y[n-i] + b_i * y^2[n-i]) \]

This is the equation of the model suggested. However this model is to be made to predict the speech signal. Hence, it would be more accurate to denote the LHS of the above equation as \( y_p[n] \) (the predicted value of \( y[n] \)) rather than \( y[n] \) itself. Hence,

\[ p_{y_p[n]} = \sum_{i=1}^{p} (a_i * y[n-i] + b_i * y^2[n-i]) \]  \hspace{1cm} (4.1)

The prediction error is simply the difference between the actual speech signal and its predicted value.

Therefore, the prediction error \( e[n] = y[n] - y_p[n] \)

The method of minimization of the MSE was employed to compute the parameters.

The mean square error is

\[ N-1 \]
\[ E(e^2[n]) = \sum_{n=0}^{N-1} e^2[n] \]
\[ = \sum_{n=0}^{N-1} (y[n] - y_p[n])^2 \]
\[ = \sum_{n=0}^{N-1} \{y[n] - \sum_{i=1}^{p} (a_i * y[n-i] + b_i * y^2[n-i])\}^2 \]  \hspace{1cm} (4.2)

Applying Equations 4.3 and 4.4 to 4.2,

\[ \sum_{n=0}^{N-1} \{y[n] - \sum_{i=1}^{p} (a_i * y[n-i] + b_i * y^2[n-i])\} * y[n-j] = 0 \]
\[ \sum_{n=0}^{N-1} \{y[n] - \sum_{i=1}^{p} (a_i * y[n-i] + b_i * y^2[n-i])\} * y^2[n-j] = 0 \]

for \( j = 1 \) to \( p \). \hspace{1cm} (4.5)

Simplifying and reversing the order of summation,

\[ \sum_{n=0}^{N-1} \text{\( Y1[l,m]=\sum_{n=0}^{N-1} y[n]y[n-m]\) } \]
\[ \sum_{n=0}^{N-1} \text{\( Y2[l,m]=\sum_{n=0}^{N-1} y[n]y^2[n-m]\) } \]
\[ \sum_{n=0}^{N-1} \text{\( Y3[l,m]=\sum_{n=0}^{N-1} y^2[n]y[n-m]\) } \]
\[ \sum_{n=0}^{N-1} \text{\( Y4[l,m]=\sum_{n=0}^{N-1} y^2[n]y^2[n-m]\) } \]

Solving and using matrix notation,

\[ A = [Y2^{-1}Y1 - Y4^{-1}Y3]^{-1} \times [Y2^{-1}Y_{j,1} - Y4^{-1}Y_{j,2}] \]  \hspace{1cm} (4.7)
\[ B = [Y1^{-1}Y2 - Y3^{-1}Y4]^{-1} \times [Y1^{-1}Y_{j,1} - Y3^{-1}Y_{j,2}] \]  \hspace{1cm} (4.8)

\( A, B \) are the vectors representing the coefficients \( a_i, b_i \) respectively for ‘i’ from 1 to \( p \).

\( Y1, Y2, Y3, Y4 \) are square matrices such that

\[ Y1[l,m]=\sum_{n=0}^{N-1} y[n]y[n-m] \]  \hspace{1cm} (4.9)
\[ Y2[l,m]=\sum_{n=0}^{N-1} y[n]y^2[n-m] \]  \hspace{1cm} (4.10)
\[ Y3[l,m]=\sum_{n=0}^{N-1} y^2[n]y[n-m] \]  \hspace{1cm} (4.11)
\[ Y4[l,m]=\sum_{n=0}^{N-1} y^2[n]y^2[n-m] \]  \hspace{1cm} (4.12)

(l and m take values from 1 to ‘p’.)

\[ a_i, b_i \]
$Y_{j,1}$ and $Y_{j,2}$ are column vectors such that

\[ Y_{j,1}(j) = \sum_{n=0}^{N-1} y[n]y[n-j] \quad (4.13) \]

\[ Y_{j,2}(j) = \sum_{n=0}^{N-1} y[n]y[j-n] \quad (4.14) \]

$j$ takes values from 1 to ‘p’.

Thus, coefficients $a_i$ and $b_i$ were determined to develop a quadratic model for the given speech signal.

4.2. Deductions from Mathematical Analysis, used in implementation

From equation set 4 above, matrices $Y_1$, $Y_2$, $Y_3$, $Y_4$ are square matrices. $Y_1$ and $Y_4$ are symmetric matrices i.e. $Y_1(l,m) = Y_1(m,l)$ and the same holds true for $Y_4$ as well. Hence with $2^*p$ coefficients to be estimated in all (i.e. ‘p’ previous values being considered at current sample determination), instead of $p^*p$ elements to be computed for each of these matrices, it is sufficient to compute simply $\{p(p+1)/2\}$ elements for each. Similarly, coefficient matrices $Y_2$ and $Y_3$ are but the transpose of each other. Hence, to compute one of the two is sufficient.

The net result of these deductions is that from a total of $4^*p^*p$ computations that originally needed to have been performed, the actual number required is simply

\[ N(C) = 2^*\{p(p+1)/2\} + \{p^*p\} \]

\[ = p(2p+1) \quad (4.15) \]

i.e. a saving of $p(2p-1)$ terms in computation.

This quadratic model was implemented using MATLAB.

However, this algorithm by itself didn’t yield results as expected. Also, this model doesn’t have a physical significance as MPC can be shown to have (section 5). Hence, work on QPC was suspended after investigation of preliminary results and attention was turned to M-ary Prediction for the reasons explained there under.

5. M-ARY PREDICTION

5.1. Basic Model and reasons for choice

The preliminary equation for the M-ary model is

\[ y[n] = \sum_{i=1}^{m} (a_i*y[i-n]) \quad (5.1) \]

An assumption that $|y[x]|<=1$ for all x was made. (This assumption will later be supported in the implementation hence its consequences at this point will correctly serve as the justification of the choice of this model.)

Observation of the above model equation reveals that the power to which an element (a sample) is raised, increases as its delay with respect to the current sample (i.e. the sample to which its contribution is to be estimated) increases.

In the process of production of speech as for any physical process, the decay of energy occurs with the passage of time. Similarly, the effect of a sample at time $t$ will decay to a much smaller value at time $(t+dt)$. As $dt$ increases, this reduction of effect will increase progressively. It is this decaying that is modeled in the above equation. As it has been assumed that $|y(x)|<=1$ for all x under consideration, as the delay ‘i’ in the above equation increases, the contribution of a sample to future samples keeps decreasing which
models the decay phenomenon in actual speech production. Additionally, it permits the setting of a value of ‘m’ that represents the maximum number of previous values that will affect the current sample value. This can be interpreted as a measure of the Reverberation Time. This in turn will help to control the intelligibility of the reproduced speech. Now that the reasons for the choice of this model and the constraints have been discussed, let’s move on to a brief look at the equations and the relevant results, which have been utilized in the implementation.

5.2. Development of Equations

The Preliminary equation is

\[ y[n] = \sum_{i=1}^{m} (a_i \cdot y[n-i]) \]  

(5.1)

Again, since this is an approximate model and attempts to predict the signal, \( y[n] \) can be substituted with \( y_p[n] \).

The instantaneous value of the error is

\[ e[n] = y[n] - y_p[n] \]

The Mean Square Error is

\[ E(e^2[n]) = \sum_{n=0}^{N-1} e^2[n] \]

\[ = \sum_{n=0}^{N-1} (y[n] - y_p[n])^2 \]

\[ = \sum_{n=0}^{N-1} \sum_{i=1}^{m} \{y[n] - \Sigma (a_i \cdot y[n-i])\}^2 \]  

(5.2)

By the method of minimization of Mean Square Error,

\[ \hat{\partial E(e^2[n])} = 0 \text{ for } i=1 \text{ to } m \]  

(5.3)

Applying equation 5.3 to 5.2,

\[ \sum_{n=0}^{N-1} \sum_{i=1}^{m} \{y[n] - \Sigma (a_i \cdot y[n-i])\}^2 = 0 \]  

(5.4)

for \( j=1 \) to \( m \).

Separating and reversing the order of summation,

\[ \sum_{i=1}^{m} \sum_{n=0}^{N-1} a_i \cdot y[n-i] \cdot y_j[n-j] = \sum_{n=0}^{N-1} y[n] \cdot y_j[n-i] \]  

(5.5)

for \( j=1 \) to \( m \).

Writing in matrix notation,

\[ A_c \cdot A = Y_{0j} \]  

(5.6)

Where,

\( A \) is the M-ary Prediction coefficients’ vector;
\( A_c \) = the constant coefficient vector of the equation;
And \( Y_{0j} \) = the constant vector.

\[ A(i) = a_i, \text{ the } i\text{th } M\text{-ary Prediction coefficient;} \]

\[ A_c(i,j) = \sum_{n=0}^{N-1} y[n-i] \cdot y[n-j] \]  

(5.7)

\[ Y_{0j}(j) = \sum_{n=0}^{N-1} y[n] \cdot y[j[n-j]] \]  

(5.8)

In all the above equations, \( i \) and \( j \) range from 1 to \( m \).
5.3. Deductions from Mathematical Analysis, used in implementation

From the above equation Eq.5.7, it can be observed that Ac(i,j)=Ac(j,i). This symmetry was used to reduce the number of computations required. From a total of m*m (summation term) computations, the required number was reduced to m(m+1)/2.

Thus, the total number of computations (summations over <N>) required for the determination of the M-ary predictive coefficients is

\[ N(C) = m + m(m+1)/2 \]  \hspace{1cm} (5.9)

(Note: Comparing with Eq.4.15 that gives \( N(C) \) for QPC, it can be observed that MPC requires less number of computations.)

5.4. Implementation

One key feature to be remembered during the analysis of the implementation is the fact that the decay of a speech sample that occurs physically can only be modeled using the above equation when, in mathematical terms, \( |y(x)| \leq 1 \).

The speech data acquisition was done in MATLAB using the simple wavread command. The speech sample sequence returned by this command is already normalized and lies in the range [-1 1]. However, from a numerical methods point of view, when the condition of \( |y(x)| < 1 \) occurs, higher and higher powers of this sample tend towards zeros i.e. they make zero (negligible) contribution. To handle the speech data read in as it is would thus result in loss of fine details. This was handled in the implementation by adaptively scaling the data as and when each frame is encountered so that the maximum possible use of the amplitude range [-1 1] is made thus preventing any loss of fine details due to the relative smallness in magnitude of low-valued samples in a frame because of the limitations of resolution (here MATLAB’s and in a real-world implementation, the limitations of the hardware).

The final stage in the implementation was a median filter with a window size of 5, to reduce the noise in the reproduced speech signal.

5.5. Parameters Identified after preliminary work

After the preliminary work, the identified parameters were:
1. The frame size (N in the above equations for the models)
2. The parameter ‘m’, representative of the Reverberation time
3. The gradient of the signal within the observed frame
4. The number of overlapping samples

5.6. Add-ons for MPC

![Figure 1. First frame of 1000](image1.png)

![Figure 2. Second frame of 1000](image2.png)
Table 1: Comparison of Gradient (maximum and mean) and MSE for two 1000-sample frames of speech under conditions of no upsampling and upsampling at a factor of 2.

<table>
<thead>
<tr>
<th></th>
<th>FRAME 1</th>
<th>FRAME 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No Upsampled</td>
<td>Upsampled @ 2</td>
</tr>
<tr>
<td>Gradient (max)</td>
<td>0.0079</td>
<td>0.0879</td>
</tr>
<tr>
<td>MSE</td>
<td>4.998</td>
<td>1.1306</td>
</tr>
</tbody>
</table>

Upsampling is the add-on applied for all the MPC implementations whose statistics are tabulated in this paper. Upsampling is the resampling of the original data at a higher rate. This is found to reduce the first order gradient of the signal and thus improve the quality of the speech output.

As can be seen from the above figures and table, upsampling greatly improves the tracking of the signal trend by the MP coefficients. However, as can also be observed from these, the factor by which upsampling is performed is very crucial. A large value would lead to excellent signal reproduction but may end up compromising on the compression. A lower value on the other hand would increase errors but facilitate excellent compression. Hence, a balanced choice of upsampling needs to be done. This led to the use of adaptive upsampling for the coding of the speech signal.

The next issue is the upsampling factor for a frame. As is deducible from the above table, the gradient of the frame plays a pivotal role in the accuracy with which it can be coded and recovered. Hence, the gradient of the signal was studied and divided into zones, depending on which the upsampling factor was assigned.

The second add-on being considered for MPC is the application of silence suppression schemes like Voice Activity Detector or Comfort Noise Generation algorithms[3]. These are currently under study.

5.7. Quantization

The quantization applied was of two kinds: Companding (logarithmic quantization[4]) and Adaptive Companding. The former was applied with a A-law compander (A=87.6) in MATLAB but found unsatisfactory as it caused a deterioration of speech quality, unaccompanied by any significant improvement with respect to bit rate. Hence, adaptive companding was investigated wherein the number of bits per sample in the current analysis frame was decided by considering the distribution of the gradients within the frame. Mu-law companding was implemented here with the standard parameter of Mu=255. (Results with and without adaptive companding are provided in Table 2 under Section 5.8).

5.8. Results

So far, MPC with fixed upsampling has been studied for male speech. Sample waveforms are given below. Performance statistics for different MPC implementations are tabulated below and compared with the corresponding parameters for some standard speech coding schemes[3][5]. Since there is no scientific objective method to test speech quality[3], three measures of Mean Opinion Score(MOS), Segmented SNR(SSNR) and Mean Square Error(MSE) have been selected for comparison.
TABLE 2: Comparison of results (using fixed upsampling for male speech) with speech coding standards

<table>
<thead>
<tr>
<th></th>
<th>Bitrate (KBps)</th>
<th>MOS</th>
<th>SEGSNR (dB)</th>
<th>MSE *10^{-3}</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPC75</td>
<td>4.88</td>
<td>3.4</td>
<td>25.67</td>
<td>13.0</td>
</tr>
<tr>
<td>MPC75.5</td>
<td>6.83</td>
<td>3.9</td>
<td>30.75</td>
<td>5.3</td>
</tr>
<tr>
<td>MPC75-AC</td>
<td>4.28</td>
<td>3.0</td>
<td>11.63</td>
<td>34.3</td>
</tr>
<tr>
<td>MPC87.5</td>
<td>5.72</td>
<td>3.6</td>
<td>13.25</td>
<td>23.0</td>
</tr>
<tr>
<td>USFS-1016-AC</td>
<td>0.6</td>
<td>3.2</td>
<td>___</td>
<td>7.6</td>
</tr>
<tr>
<td>G.721 AD PCM</td>
<td>4.0</td>
<td>4.1</td>
<td>___</td>
<td>19.38</td>
</tr>
<tr>
<td>GSM EFR</td>
<td>1.53</td>
<td>4.0</td>
<td>___</td>
<td>___</td>
</tr>
<tr>
<td>GSM</td>
<td>1.625</td>
<td>3.5</td>
<td>___</td>
<td>___</td>
</tr>
<tr>
<td>LPC-10²</td>
<td>0.3</td>
<td>2.3</td>
<td>___</td>
<td>___</td>
</tr>
</tbody>
</table>

- data unavailable

1 - data from [3] and [5]
2 - data from [6]

Segmented SNR has been recorded since this provides a better indication of the perceptual quality[4]. For the specified standards though, since perceptual quality is of utmost concern, the SNR is not as meaningful[5].

The given standards for comparison do not include nonlinear coding techniques. One such speech coding technique is sine wave synthesis of speech, which attempts to prove that speech can be adequately described by the changing pattern of vocal resonances alone disregarding the acoustic elements of speech[2]. Though implementation details are unavailable, the quality of the speech output is comparable to MPC75-AC.

6. CONCLUSION

The results for implementations of MPC so far have been obtained with fixed upsampling only. The application of adaptive upsampling is to be investigated.

The frequency domain significance of the model is also being studied. Since conventional transform techniques cannot be applied, statistical means are being investigated to study the frequency response of the M-ary Predictive Model.

7. REFERENCES