Signal Processing
Lab 5 : Linear Predictive Coding

Panagiotis Koutsourakis *Ahmed Besbes* *Loic Simon*
Chaohui Wang

December 5, 2008

1 Introduction

In this Lab, you will see the Linear predictive coding (LPC), which is a popular technique for speech compression and speech analysis.

LPC assumes that the present sample of the signal is predicted by the past p samples of the signal (also known as the order of LPC) such that:

$$\widehat{y(n)} = \sum_{i=1}^{p} a_i y(n-i)$$

where $\widehat{y(n)}$ is the prediction of $y(n)$, $y(n-i)$ is the i-th previous sample and $\{a_i\}$ are called the linear prediction coefficients. The error between the actual sample and the predicted one (also known as reconstruction error) can be expressed as:

$$\epsilon(n) = y(n) - \widehat{y(n)} = y(n) - \sum_{i=1}^{p} a_i y(n-i).$$

As we saw in class, the optimal coefficients verify:

$$\begin{aligned} \mathbb{E}[e(n)y(n-m)] = 0, m \in \{1, \ldots, p\} \Rightarrow \\ \mathbb{E}\left[\left(y(n) - \sum_{k=1}^{p} a_k y(n-k)\right) y(n-m)\right] = 0, m \in \{1, \ldots, p\} \Rightarrow \\ \mathbb{E}[y(n)y(n-m)] = \sum_{k=1}^{p} a_k \mathbb{E}[y(n-k)y(n-m)], m \in \{1, \ldots, p\} \end{aligned}$$

This results in p unknowns in p equations. By using an estimation of the correlation we end up with :

$$\begin{aligned} \begin{pmatrix} r(0) & r(1) & \cdots & r(p-1) \\ r(1) & r(0) & \cdots & r(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ r(p-1) & r(p-2) & \cdots & r(0) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{pmatrix} &= \begin{pmatrix} r(1) \\ r(2) \\ \vdots \\ r(p) \end{pmatrix} \\ \mathbf{Ra} &= \mathbf{r} \\ \mathbf{a} &= \mathbf{R^{-1}r} \end{aligned}$$

where $r(k) = \sum_{n=0}^{M-1-k} y(n)y(n+k)$ and M is the number of samples.

2 LPC with Auto-correlation

Model the signal *"paint_ball.wav"* for the interval $[6000, 75000]$, where the signal is approximately stationary, using the LPC with auto-correlation.

Q1 Write a function that estimates the auto-correlation of a signal. Tips :

- The prototype of the function could be `corr=lab_autocorrelation(signal)`, where `signal` is the input signal and `corr` the auto-correlation of the input signal .
- For the core of the function use the matrix multiplication instead of for loops, the Matlab function `toeplitz` if possible. You will need to pad the signal with zeros, in order to do the computation.

Q2 Write a function that estimates the coefficients of the LPC of the signal. Tips:

- The prototype of the function could be `A = lab_estimate_coefficient(corr, p)`, where `corr` is the auto-correlation of the signal, `p` is the order of LPC and `A` is the coefficients of LPC.
- Use Matlab function `toeplitz` to compute matrix **R**.

Q3 Estimate the coefficients of the LPC for the signal filtered by a Hamming window (this estimation is called the auto-correlation method) for the order values $p = 5, 10, 50, 100$. Tips:

- Compute the hamming window using Matlab function `hamming`.

Q4 Plot the reconstruction error of LPC for $p = 5, 10, 50, 100$ and estimate its standard deviation. Tips:

- Use the Matlab function `filter` to estimate the reconstruction error.
- Store the standard deviation of each p value in a matrix , for the Q5.

Q5 Plot the standard deviation of the different values of p you estimated in the Q4.

Q6 Plot the log spectrum of the reconstruction error for different values of order $p = 5, 10, 50, 100$. Explain what happens as the order increases.

Q7 Plot the spectrum of the original signal and compare with the spectra of the parametric signals using the estimated coefficients of the LPC for different order values ($p = 5, 10, 50, 100$). Tips:

- The spectrum of the parametric signal could be computed by using $[|\frac{std(e(n))}{1-\sum_{k=1}^{p} a_k e^{-j2\pi kn/M}}|]$