# ECE 792-41 Statistical Methods for Signal Analytics

## Project 1: Linear Prediction and Synthesis

The grade of the project will be based on the completeness, performance, and novelty of your design as well as the quality of your report. You should prepare a README file (in txt or html) to describe your files for source codes, audio results, and report. Details can be found in the submission guideline.

You may use MATLAB or other programming languages such as Python, R, and C++ to do your project. You are allowed to use only primitive functions such as convolution, FFT, etc. You should implement your own functions that are closely related to the course materials such as autocorrelation, Levinson-Durbin recursion, etc. You may use more sophisticated functions only for verification purposes. You are not allowed to use or modify upon any speech encoder/decoder built by other people (such as those in the Internet and textbook CDs). Any resources used in your work should be cited at the end of your report.

This is an individual project. You should sign the **Honor Pledge** at the beginning of the report: "*I pledge in my honor that I have not given or received any unauthorized assistance on this report.*"

---

1. **Simulating wide-sense stationary AR signals:** Implement a function capable of generating a simulated AR($p$) signal $\{u(n)\}$. Let $p = 5$. Choose a set of parameters so that the AR process will be stable, and fix the parameters throughout this problem.

(1) In Problem 4 of Homework 2, we have shown that the mean and variance of a simulated random variable $u(n)$ converges to the mean and variance of the true AR process. Use this theoretical result to design a test that allows you determine how many elements need to be thrown away for your chosen AR parameters so that the remaining simulated values form a wide-sense stationary AR process. Modify your AR signal generation function accordingly for future use. (Hint: You may want to repeatedly generate AR signals using independent innovations. Calculate the sample variance at every time index.)

(2) Calculate the value of the theoretical autocorrelation sequence $\{r(k), k = 0, \ldots, 5\}$ for your chosen parameters. (Hint: Express the Yule-Walker equations in the matrix-vector form about the coefficient vector $\mathbf{a} = [a_1, \ldots, a_5]^T$, and then reformulate it into a linear system about an unknown vector $\mathbf{r} = [r(0), \ldots, r(5)]^T$.)

(3) Use the correlation ergodicity property to estimate the *unbiased* autocorrelation $\hat{r}(k)$. Plot its mean squared error as a function of the sequence length.

(4) Use the correlation ergodicity property to estimate an *biased* autocorrelation $\tilde{r}(k)$. Plot in one figure its mean squared error, bias squared, and sample variance as a function of the sequence length.

**2. Linear Prediction and Levinson-Durbin Recursion:**

(1) Derive from scratch the normal equation (N.E.) and augmented N.E. for the order-5 forward linear prediction.

(2) Derive from scratch the Levinson-Durbin Recursion for the augmented N.E.

(3) Replace the theoretical quantities in the N.E. by their *biased* autocorrelation estimates, and solve the forward prediction coefficient vector by direct inverse. Repeat such procedure with different random seeds 1000 times, and obtain 1000 estimated forward prediction coefficient vectors. What are the sample mean and the sample

variance-covariance matrix for these 1000 vectors? What do the diagonal elements and off-diagonal elements of the sample variance-covariance matrix measure, respectively?

(4) Repeat (3) using *unbiased* autocorrelation estimates. Compare the results with those of (3). Explain your observations.

(5) Replace the matrix inverse operation in (3) by Levinson-Durbin Recursion. Compare the results with those of (3). Explain your observations.

3. **Analyzing Speech Signals**: This task lets you explore the basic analysis and synthesis of speech signal based on the concept of linear prediction. Some test clips are provided in course webpage, encoded at 8 kHz and 8 bits per sample.

(1) Design an automated procedure to determine the average duration of the speech signal within which the signal is approximately wide-sense stationary. Draw a block diagram or write pseudocode in the report for the designed procedure. We shall call the best average duration the "frame length." You may partition the speech signal into frames of this length in the rest of this problem.

(2) Build a $10^{th}$-order linear predictive model and implement a function that can efficiently find the optimal prediction coefficients for a given frame of the speech signal. (You may reuse the functions written for Problem 1.) Show through simulation how much is the difference between the true speech signal and the predicted one from your model. Note that you can use different coefficients for different frames, and you should examine the difference both "objectively" (using some suitable quantitative measures) and "subjectively" (listen to the signals).

(3) Examine variations on the above speech analysis: through simulations of higher and lower order than 10, discuss how the selection of order affects the performance of the prediction.

4. **(Optional) Linear Prediction Codec**: So far you have built tools to determine the model coefficients and the prediction error ("residue"). As there are just a few coefficients for each frame of a speech signal and the dynamic range of the residue is small, you can use fewer bits to represent and transmit the speech signal than keeping the raw samples at a fixed rate of 8 bits/sample. Develop a coding scheme, which should determine when you need to update and transmit the model coefficients, how the transmitter can inform the receiver of this model update, how you encode the parameters and the residues, and other things you find necessary. Write the output of your encoder as a bitstream to a Mat file, which will be read by a decoder to reconstruct the speech. Justify the choices you have made in your design. Determine the compression ratio of the best result your codec on test clips can have in terms of some good performance index (which may take account of sound quality, waveform distortion, etc.).

```
Note: You should conduct your exploration and implementation in a systematic way:
start with the simple choices (such as a simple quantizer and encoding, etc.), devise
approaches to validate your design as well as each implementation step/module, and
gradually refine your work.
```

### Resources and suggestions on technical and write-up for your project

- Make sure you explain and analyze your results: What do they tell us? What conclusion can we draw? Critical evaluations and discussions are keys to a good technical report. Don't just put figure/table and move on. Make sure you label the axes of your figures, and provide self-explanatory legend for multiple curves.

- When you print out figures, try to avoid using dark background when possible – it won't show up well and it consumes lots of ink. Also please use double-sided printing for your report when possible.