2021 ASAS Homework 3: Linear Prediction and Time-*variant* Filtering

Prof. Yi-Wen Liu

Due Sunday March 28, 2021.

Assigned reading material: B. S. Atal and Suzanne L. Hanauer, "Speech analysis and synthesis by linear prediction of the speech wave," J Acoust Soc Am 50, 637 (1971)

In this homework, you will learn to represent a short speech signals as the result of an excitation signal filtered by the vocal tract. The key technique is linear prediction.

We often assume that a human voice signal is *quasi*-stationary, meaning that though it is not stationary, we can regard it as stationary if we only look at aa short duration every time. To do that, the first step is to chop the speech signal into frames by applying a rectangular window:

```
win = ones(L,1); % Rectangular window of length L.
for kk = 1:numFrames
     ind = (kk-1)*L+1:kk*L;
     ywin = y_emph(ind).*win;
end
```

The number of frames per second is called the *frame rate*. You can choose to overlap adjacent frames or not.

In the above, **`y_emph(:)`** is your voice signal filtered with a two-tap FIR that has the following z-transform expression: $H(z) = 1 - \alpha z^{-1}$, where α can be any number from 0.9 to 0.99, empirically. This step is called "emphasis" and it effectively compensates the high frequency loss due to lip radiation.

```
y_emph = filter([1 -0.95],1,y);
```

Remarks: Remember in HW2, we did "differencing". Emphasis is similar.

Then, for each frame, we need to first estimate the vocal tract filter, and then *inverse-filter* the signal to (try to) recover the *glottal* source.

The estimation of vocal tract filter can be achieved via **linear prediction**. Basically, we assume that the emphasized voice $Y_{em}(z)$ has the following property:

$Y_{em}(z) = E(z)\frac{1}{A(z)}$, where $1/A(z)$ represents an *all-pole* filter, and $\left|E(e^{j\omega})\right|$ is maximally flat with respect to ω; in other words, $e[n]$'s samples are least mutually correlated.

Estimation of $A(z) \triangleq 1 - \sum_{k=1}^{P} a_k z^{-k}$ involves solving a least-square problem. Basically, we want to find the coefficients $\{a_k\}_{k=1:p}$ such that

$$y_{\text{em}}[n] \approx \sum_{k=1}^{p} a_k y_{\text{em}}[n-k] \triangleq y_{\text{pred}}[n].$$

Depending on how you define the goodness of fitting, $\{a_k\}_{k=1:p}$ can be found by minimization/optimization techniques. For this homework, we simply minimize $\sum_{n=1}^{L} |y_{\text{em}}[n] - y_{\text{pred}}[n]|^2$ for every frame of length L. The MATLAB function to use is **`lpc()`**. Let us denote the corresponding optimal filter coefficients as $\{\hat{a}_k\}_{k=1:p}$, and the corresponding filter's z-transform notation is $\hat{A}(z) = 1 - \sum_{k=1}^{P} \hat{a}_k z^{-k}$.

Notes:

- Each frame will give different answer of $\{\hat{a}_k\}_{k=1:p}$.
- Please be careful about the plus or minus sign for this homework. If in doubt, check with us regarding the notation. Chances are we may have typos.

Now, to recover the glottal source signal, we simply apply $\hat{A}(z)$ to $Y_{\text{em}}(z)$; that is,

$$e[n] = y_{\text{em}}[n] - \sum_{k=1}^{p} \hat{a}_k y_{\text{em}}[n-k].$$

Since each frame has a different $\hat{A}(z)$, this is time-*variant* filtering. From each input frame you will create an output frame. The tricky part is this final part: you will need to put the output frames together by concatenation, *overlap-add* or any other way you can think of to produce a long output signal of the same length as the original voice signal.

Activities:

1. If things are implemented correctly, the result should not contain "frame-rate artifacts", which refers to any unwanted sound due to discontinuity at frame boundaries. If it occurs, you will hear a buzzing or cracking sound at the pitch of the frame rate (say 50 Hz).
2. If you can be sure that your output has no frame-rate artifacts, adjust the linear prediction order p and the frame rate to see if you can make the speech *un*-intelligible. The general principle is to look at the spectrum of $e[n]$. If the spectral envelope becomes sufficiently flat, you should no longer be able to tell what is the vowels and consonants.
3. **Bonus**: by replacing $e[n]$ with something else, denoted as $e'[n]$ and preferably a broadband sound, and filter $e'[n]$ with $1/\hat{A}(z)$ (which will be a time-*variant* IIR filter), you may create hybrid sound that has both speech and non-speech qualities.