Professor Fearing EECS120/Python MiniLab 1 v1.01 (draft) Fall 2016

Due at 6 pm, Fri. Sep. 30 on BCourses

Up to 2 people may turn in a single iPython (Jupyter) notebook. Upload .ipynb and answers in pdf.

This exercise is worth 1% grade. (Python exercises will be 9% of grade).

Notes: Use Python 2.7, not 3.

Notebook file is: `www-inst.eecs.berkeley.edu/~ee120/fa16/hwk/PY1-drum.ipynb`

Sound file is: `www-inst.eecs.berkeley.edu/~ee120/fa16/hwk/Tight-High-Tom-lpf.wav`

In this Python exercise, you will try to model the sound of a drum as the unit sample response of a linear difference equation (all-pole model) with infinite impulse response output $y[n]$. (Note, the drum is probably non-linear, and the drum stick is not quite an impulse, so results may vary.)

The LDE is given by:

$$\sum_{k=0}^{N} a_k y[n-k] = b_0 \delta[n] \qquad \text{or} \qquad y[n] = \frac{1}{a_0}\left(-\sum_{k=1}^{N} a_k y[n-k] + b_0 \delta[n]\right) \tag{1}$$

Assume initial conditions $y[n] = 0$ for $n < 0$. Let $a_0 = 1$.

1. (10 pts) Assume $N = 4$ and given coefficients a_1, a_2, a_3, a_4. Find $y[n]$ for $n = 0, 1, 2, 3$ for unit sample input (note that $y[0] = b_0/a_0$).

2. (30 pts) In the `PY1-drum` notebook, add code to the `for` loop to calculate `z` which is $y[n]$ from eq.(1). Also, update `state[]` which should contain $[y[n]\ y[n-1]\ ...\ y[n-N]]^T$.

Given a sound file, we want to estimate the filter coefficients a_k. The unknown a_k can be found from the least square inverse of the system $V\mathbf{a} = -\mathbf{y}$ where V are the stacked outputs of the system, and $\mathbf{y}$ is the vector of outputs. For example, with 6 measurements of $y[n]$ and 4 a_k, the LDE solution can be expressed as:

$$V\mathbf{a} = \begin{bmatrix} y[0] & 0 & 0 & 0 \\ y[1] & y[0] & 0 & 0 \\ y[2] & y[1] & y[0] & 0 \\ y[3] & y[2] & y[1] & y[0] \\ y[4] & y[3] & y[2] & y[1] \\ y[5] & y[4] & y[3] & y[2] \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = - \begin{bmatrix} y[1] \\ y[2] \\ y[3] \\ y[4] \\ y[5] \\ y[6] \end{bmatrix} = -\mathbf{y}.$$

The least squares solution for $\mathbf{a}$ is given by $\mathbf{a} = -(V^T V)^{-1} V^T \mathbf{y}$

The quality of fit can be quantified by the root-mean-square error:

$$\text{r.m.s. error} = \sqrt{\frac{1}{N}\sum(\hat{y} - y)^2}$$

where $\hat{y}$ is the LDE output, and y is the measured data.

3. (30 pts) In the `PY1-drum` notebook, choose the size of the state vector `colDim` and number of data samples `rowDim` such that rms error is less than 2500. Note that data is sampled at 44 kHz, so samples are only 22 μs apart.

4. (20 pts) Play the `synthDrum.wav` file. Qualitatively describe differences between the sound of the original sampled signal and the reconstructed signal. What might be missing in the reconstructed sound?

5. (10 pts) Choose a different percussion sound file from `freewavesamples.com/sample-type/drums/kick`, choose `rowDim, colDim` to find a low rms error. Qualitatively describe whether the LDE output is a good match for the original sound. Note sound file source, and show plot comparing $y, \hat{y}$ over number of data samples.