Applied Signal Processing Home Assignment #1: Adaptive Filters

1 Background

The adaptive filtering problem elaborated in class can be briefly described as follows. We assume that we sample a random signal

$$Y_n = S_n + Z_n \tag{1}$$

where S_n is the desired signal and Z_n is the (undesired) additive noise, and capitalization is used to signify that the signals are random. We would like to devise a noise predictor $\hat{Z}_n$ and subtract it from the noise signal, hence obtaining

$$e_n = Z_n - \hat{Z}_n \tag{2}$$

where e_n is the residual estimation noise (the notation is not capitalized, in order to avoid confusion with the expectation operator).

The predictor $\hat{Z}_n$ is derived using a causal linear filter

$$\hat{Z}_n = \sum_{\ell=1}^{L} w_{n,\ell} Z_{n-\ell}. \tag{3}$$

Note that the time index ℓ starts at 1, implying that it is causal, and ends at $L < \infty$ implying it is finite in time. The filter coefficients (or *taps*) can vary in time, i.e. $w_{n,\ell}$ are functions of both n, and ℓ, as done in the RLS, LMS adaptive filters. Alternatively, the coefficients can be time invariant, $w_{n,\ell} = w_\ell$, implying that the prediction is a linear-time invariant (LTI) filtering operation.

A more generalized notation denoting the desired signal by $\{D_n\}$ and the process from which it is estimated by $\{U_n\}$ is given in the lecture notes. For simplicity we also assume that all the random signals have zero mean.

1.1 A Synthetic Wide-Sense Stationary Process

The synthetic WSS process to be used in this assignment is a first order auto-regressive process, AR(1), sampled with additive noise. Namely, the noise process Z_n corresponds to:

$$Z_n = X_n + N_n \tag{4}$$

where $\{N_n\}$ is a white Gaussian noise with $\mathbb{E}N_n = 0$, and $\mathbb{E}N_n^2 = \sigma_N^2$ (and clearly $\mathbb{E}N_nN_{n-l} = 0$ for every $l \neq 0$), statistically independent of $\{X_n\}$. X_n is an AR(1) process generated by the following recursion formula:

$$X_0 \sim N\left(0, \frac{1}{1-\alpha^2}\right), \tag{5}$$

$$X_n = \alpha X_{n-1} + G_n \ \forall n \geq 1, \tag{6}$$

where $\alpha < 1$ is the parameter of the process and $\{G_n\}$ (the *innovation noise*) is a white Gaussian noise process with zero mean and unit variance (namely, $\mathbb{E}G_n = 0$, $\mathbb{E}G_n^2 = 1$). It is also assumed that $\{N_n\}$, $\{G_n\}$ and X_0 are independent.

As shown in a previous course, the auto-correlation function of the AR(1) process is

$$R_{X,\ell} = \frac{\alpha^{|\ell|}}{1-\alpha^2}\mathbb{E}G_n^2 = \frac{\alpha^{|\ell|}}{1-\alpha^2} \tag{7}$$

since $\mathbb{E}G_n^2$ is chosen here to be set to one. The auto-correlation function of Z_n is therefore

$$R_{Z,\ell} = \begin{cases} \frac{1}{1-\alpha^2} + \sigma_N^2 & \text{for } \ell = 0 \\ \frac{\alpha^{|\ell|}}{1-\alpha^2} & \text{otherwise} \end{cases} \tag{8}$$

1.2 Processing Audio in Matlab

Most of this assignment is dedicated to signal processing, and in particular audio processing in a computer environment. We encourage you to use Matlab as the programming platform. Python programming language, and in particular the NumPy environment, can also be used in a similar way. The following notes could be found helpful in your implementation:

1. Sampling frequency. Use 48KHz as the default.

2. Audio playback. In Matlab, use the following command

   ```
   sound(y,Fs);
   ```

   where 'y' is the vector to be played and 'Fs' is the sampling frequency in Hz.

   Note: The vector y should contain values between -1 and 1. Smaller values will lead to low volume and values larger than 1 will lead to clipping noises. Pay special attention to the scaling of the signals when playing the estimation residual error, since trivially, wrong scaling might lead you to the wrong conclusions.

3. Reading a WAV file. In Matlab use:

   ```
   [y,Fs] = audioread(filename);
   ```

   where 'filename' is the file name.

4. Writing to a WAV file. In Matlab use:

```
audiowrite(filename,y,Fs);
```

5. Filtering. For linear time invariant filtering use in Matlab:

   ```
   y = filter(b,a,x)
   ```

   which implements the following difference equation:

   ```
   a(1)*y(n) = b(1)*x(n) + b(2)*x(n-1) + ... + b(nb+1)*x(n-nb)
   - a(2)*y(n-1) - ... - a(na+1)*y(n-na),
   ```

   where 'na' is the number of feedback coefficients (related to the vector 'a') and 'nb' is the number of feedforward coefficients (related to the vector 'b').

   Pay attention to the minus signs at the right hand side of the equation!

The following audio files are attached to the assignment. Each of them contains one minute of audio, sampled at 48KHz:

1. 'airplane.wav': contains an airplane cabin noise.
2. 'cafe.wav': contains background noise recorded in a cafe.
3. 'city.wav': city noise
4. 'vacuumcleaner.wav': vacuum cleaner noise.

2 Questions

Question 1 - Optimal Linear Estimation

1. Refer to the noise process defined in Subsection 1.1 and appropriately use the notation shown in the lecture notes. Find the desired signal $\{D_n\}$, the measurements signal $\{U_n\}$. Write the matrix $\mathbf{R}$ and vector $\mathbf{p}$ in terms of the process parameters, α and σ_N^2.

2. Write the equation for the optimal linear prediction filter.

3. Discuss the following two special cases. Explicitly calculate the optimal linear filter for each case, and give intuition to the solution:

   (a) $0 < \alpha < 1$ and $\sigma_N^2 = 0$. Prove that in this case, the optimal filter has the form:

   $$\mathbf{w}^* = \begin{bmatrix} w_0 \\ 0 \\ \vdots \\ 0, \end{bmatrix} \tag{9}$$

   and then find w_0. <u>Hint:</u> there is no need to invert $\mathbf{R}$. Give intuition to the solution.

   (b) $\alpha = 0$ and $\sigma_N^2 > 0$ What is the number of non-zero filter coefficients? Give intuition to the solution.

4. Write a Matlab code that generates the WSS signal. Set the number of samples of the process to correspond to 10 seconds of audio in 48KHz sampling frequency. You can create the signal by drawing white Gaussian process using 'randn' command, and passing it through a filter for which $b = [1]$ and $a = [1, -\alpha]$ (make sure you understand why α should have a minus sign!).

   (a) For $\alpha = 0.5$ and $\sigma_N^2 = 1$, calculate the empirical mean:

   $$\frac{1}{N}\sum_{\ell=1}^{N} Z_\ell, \tag{10}$$

   and the empirical second moment:

   $$\frac{1}{N}\sum_{\ell=1}^{N} Z_\ell^2, \tag{11}$$

   where N is the number of samples in the process. Compare the values you calculated and compare them to the expected theoretical values.

   (b) Find the appropriate scaling coefficient β, so that $\mathbb{E}(\beta Z_n)^2 = \frac{1}{2}$.

   (c) Play the scaled signal using Matlab sound command. Explain why the scaling is necessary.

5. For $\alpha = 0.9$ and $\sigma_N^2 = 0.5$ do the following:

   (a) Calculate the optimal filters coefficients of orders $L = 1$ to $L = 5$. Hint: You can use Matlab 'toeplitz' command for the calculation of $\mathbf{R}$. Write the filter coefficients in your solution.

   (b) Implement the filters in Matlab and calculate $\hat{Z}_n$. Hint: If you use Matlab 'filter' command, and set the first coefficient of the b vector to zero. Make sure you understand why this is necessary.

   Calculate the prediction error signal $e_n = Z_n - \hat{Z}_n$ for all values of L. Calculate the appropriate scaling coefficient β, such that $\mathbb{E}(\beta Z_n)^2 = \frac{1}{2}$. Use Matlab 'sound' command to listen to the scaled original noise process βZ_n and to the scaled residual prediction error βe_n. Does the prediction error sound lower?

   (c) Calculate the average estimation error, $\frac{1}{N}\sum_{\ell=1}^{N} e_\ell^2$, for all values of L. Explain why this value cannot be smaller than $1 + \sigma_N^2$.

   (d) For all values of L, calculate the noise reduction (in dB scale) given in the following formula:

$$\mathrm{NR_{dB}} \triangleq 10\log_{10}\frac{\frac{1}{N}\sum_{\ell=1}^{N} Z_\ell^2}{\frac{1}{N}\sum_{\ell=1}^{N} e_\ell^2}. \tag{12}$$

   Explain the operative meaning of $\mathrm{NR_{dB}}$.

Question 2 - Steepest Descent Algorithm for the Stationary Case

For the process with parameters $\alpha = 0.9$ and $\sigma_N^2 = 0.5$, calculate the filter for $L = 4$ using the steepest descent algorithm as follows:

1. Calculate the eigenvalues of $\mathbf{R}$ (you can use Matlab 'eigs' function). What is the largest eigenvalue?

2. For every $\mu \in \{0.001, 0.01, 0.1, 0.2\}$ calculate the weight vector $\mathbf{w}_n$ over 100 iterations of the steepest descent algorithm. For every iteration (and every value of μ) calculate the error norm:

$$||\mathbf{c}_n||^2 = ||\mathbf{w}_n - \mathbf{w}^*||^2. \tag{13}$$

   Plot a figure whose X axis is the iteration (starting from 0, in which $\mathbf{w}$ is a zero vector) and whose Y axis is the relative normalized weight error in dB scale. Namely:

$$10\log_{10}\frac{||\mathbf{c}_n||^2}{||\mathbf{w}^*||^2} \tag{14}$$

   Plot all the graphs, corresponding to all values of μ on a single figure (using the 'hold' command). Set the maximal value in the Y axis to 20dB.

3. Explain the results. Relate the maximal μ value for which the error decreases to the bound seen in class ($\frac{2}{\lambda\mathrm{max}}$).

Question 3 - Least Mean Squares (LMS) Algorithm

In this question we shall design filters for the process with parameters $\alpha = 0.9$ and $\sigma_N^2 = 0.5$, using the LMS algorithm as follows:

1. Implement the LMS algorithm for the filter calculation for all the combinations of $L \in \{1, 2, 4\}$ and $\mu \in \{0.01, 0.001, 0.0001\}$. For every setting of L and μ, plot the error in the filter coefficients (14) in dB scale as a function of the iterations, also. Note that this time, the number of iterations is the number of samples in the process. For every pair (L, μ), calculate the relative prediction error as in (12) and write it in the title of the corresponding figure.
2. Explain the trade-off between μ and L. What do you think is a reasonable setting for μ and L? Justify your answer.

   Hint: Check what happens to the largest eigenvalue of $\mathbf{R}$ when L increases.

Question 4 - Recursive Least Squares (RLS) Algorithm

In this question we shall design filters for the process with parameters $\alpha = 0.9$ and $\sigma_N^2 = 0.5$, using the RLS algorithm as follows:

1. Implement the RLS algorithm as presented in the lecture notes for $L = 2$, $\lambda = 0.99$ and various values of δ . Plot the figure of the relative coefficient error (14). Calculate the relative prediction error as in (12) and write it in the title of the figure.
2. Suggest a good setting or δ and explain your choice.
3. Explain why λ should be set to be close to one.

Question 5 - Real Life Signals

In this question we will use both the LMS and RLS algorithms on the four audio files described in Subsection 1.2. For every file, use both RLS and LMS, and several settings of the algorithm parameters (L and either μ or λ). For any set of parameters you choose, do the following (no need to submit the outcome of Sections (a)-(d). The submission does need to include the answers Sections (1)-(2)):

(a) As a trivial benchmark, use a filter with $L = 1$, and $\mathbf{w} = [1]$. Namely, the current sample is used as the predictor for the next sample. Make sure you always outperform this trivial benchmark.

(b) Listen to the original files and to the residual prediction error using Matlab 'sound' command. Can you hear that the noise power is reduced? Can you hear the 'adaptation time' of the filter?

(c) Calculate the noise reduction according to (12). Take into account the prediction noise only after the adaptation time.

(d) Plot figures whose X axis the sample index and whose Y axis corresponds to the *instantaneous power*, where instantaneous power p_n of a signal x_n is defined as follows:

$$p_n \triangleq \frac{1}{M} \sum_{\ell=0}^{M-1} x_{n-\ell}^2. \tag{15}$$

Take $M = 10000$. Plot the instantaneous power of both the original signal and the prediction error. In the title of the figure, write the audio file name, the algorithm (LMS or RLS), the parameters (μ or λ) and the noise reduction $\mathrm{NR}_{\mathrm{dB}}$. An example for such a figure appears in Figure 1.

In your submission:

1. For each of the four audio files, include a figure according to the explanation in item (d) above, for at least one good setting of the RLS algorithm and one good setting of the LMS algorithm.
2. Discuss which noise processes are more predictable and which noises processes are less predictable.

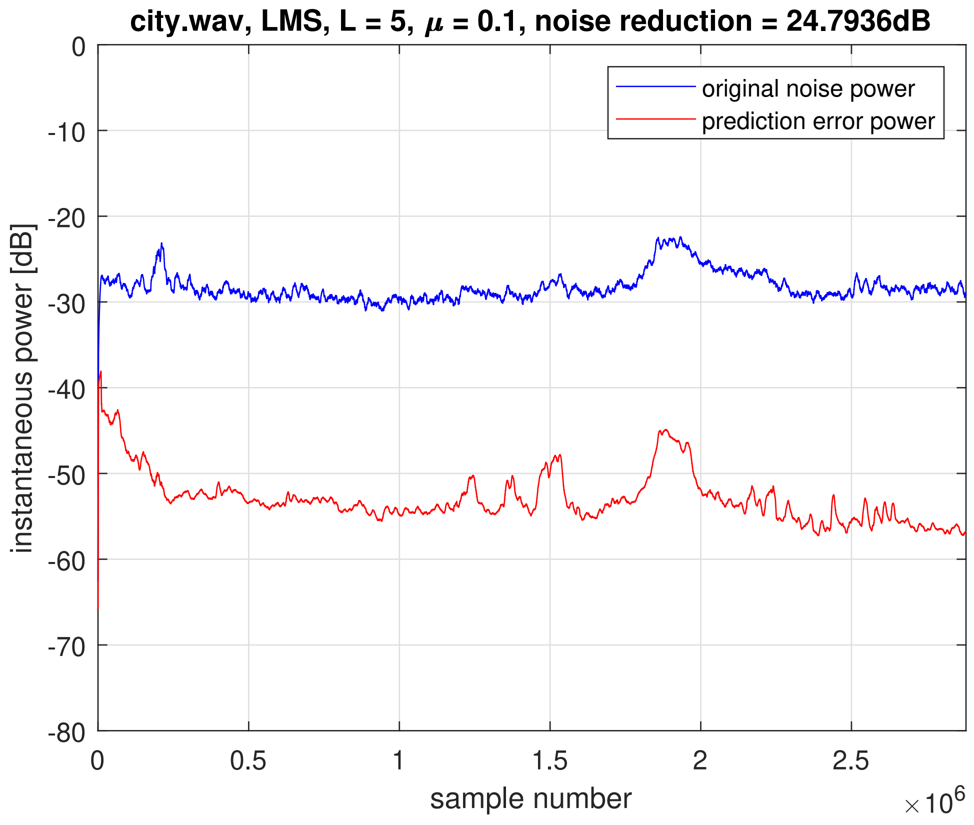

Figure 1: Example figure for Question 5, Section 1

Question 6 - Competition

Submit a Matlab file (with extension '.m') containing the function with the following header:

```
function znext = adaptivepredict(zvec)
```

The function input 'zvec' corresponds to a vector of samples. The function output 'znext' is a scalar containing the prediction of the next sample of the process (not included in the zvec vector). The prediction can be implemented by either RLS or LMS with your preferred parameter setting. Your function will be evaluated on a different audio file (not included in the assignment) by checking the average prediction error in various locations in the file.

You may submit a Python file instead of Matlab (though you are strongly encouraged to use Matlab).