ECE551 - Homework 6

Khoi-Nguyen Mac

October 30, 2016

1 DTFT of Auto-correlation and Cross-correlation

$$\begin{aligned}
C_{x,y}(\omega) &= \sum_n c_{x,y}[n]e^{-jn\omega} \\
&= \sum_n \mathbb{E}\left[x[n]y[n]\right]e^{-jn\omega} \\
&= \sum_n \mathbb{E}\left[x[n](x[n]+w[n])\right]e^{-jn\omega} \\
&= \sum_n \mathbb{E}\left[x[n]x[n]\right]e^{-jn\omega} + \sum_n \mathbb{E}\left[x[n]w[n]\right]e^{-jn\omega} \\
&= \sum_n a_x[n]e^{-jn\omega} \qquad (\because x[n], w[n] \text{ are uncorrelated}) \\
&= A_x(\omega)
\end{aligned}$$

$$\begin{aligned}
A_y(\omega) &= \sum_n a_y[n]e^{-jn\omega} \\
&= \sum_n \mathbb{E}\left[y[n]y[n]\right]e^{-jn\omega} \\
&= \sum_n \mathbb{E}\left[(x[n]+w[n])(x[n]+w[n])\right]e^{-jn\omega} \\
&= \sum_n \mathbb{E}\left[x[n]x[n]\right]e^{-jn\omega} + \sum_n \mathbb{E}\left[w[n]w[n]\right]e^{-jn\omega} + \sum_n 2\mathbb{E}\left[x[n]w[n]\right]e^{-jn\omega} \\
&= \sum_n \mathbb{E}\left[x[n]x[n]\right]e^{-jn\omega} + \sum_n \mathbb{E}\left[w[n]w[n]\right]e^{-jn\omega} \\
&= A_x(\omega) + A_w(\omega)
\end{aligned}$$

2 Higly Correlated Random Processes

(a)

$$x_1[n] = \begin{cases} A & \text{even } n \\ B & \text{odd } n \end{cases}$$

Half of the sequence is A and the other half is B, so $\mathbb{E}\left[x_1[n]\right] = \mathbb{E}\left[\frac{A+B}{2}\right] = 0$ is a constant.

$$a_{x_1}[n_1, n_2] = \mathbb{E}\left[x_1[n_1]x_1[n_2]\right] = \begin{cases} \mathbb{E}\left[A^2\right] = 1 & & n_1, n_2 \text{ even} \\ \mathbb{E}\left[B^2\right] = 1 & & n_1, n_2 \text{ odd} \\ \mathbb{E}\left[AB\right] = 0 & (A, B \text{ uncorrelated}) & \text{else} \end{cases}$$

We have $x_1[0] = A$, so

$$a_{x_1}[0, n1 - n_2] = \mathbb{E}\left[x_1[0]x_1[n_1 - n_2]\right] = \begin{cases} \mathbb{E}\left[A^2\right] = 1 & \text{both odd or even} \\ \mathbb{E}\left[AB\right] = 0 & \text{one odd, one even} \end{cases}$$

$a_{x_1}[n_1, n_2] = a_{x_1}[0, n_1 - n_2]$, so $x_1[n]$ is WSS. Since its values keep alternating between A and B, it is periodic.

$$x_2[n] = \begin{cases} A & n \geq 0 \\ B & n < 0 \end{cases}$$

Similarly, $\mathbb{E}\left[x_2[n]\right] = \mathbb{E}\left[\frac{A+B}{2}\right] = 0$. We have

$$a_{x_2}[n_1, n_2] = \mathbb{E}\left[x_2[n_1]x_1[n_2]\right] = \begin{cases} \mathbb{E}\left[A^2\right] = 1 & n_1, n_2 \geq 0 \\ \mathbb{E}\left[B^2\right] = 1 & n_1, n_2 < 0 \\ \mathbb{E}\left[AB\right] = 0 & \text{else} \end{cases}$$

and

$$a_{x_1}[0, n1 - n_2] = \mathbb{E}\left[x_1[0]x_1[n_1 - n_2]\right] = \begin{cases} \mathbb{E}\left[A^2\right] = 1 & n_1 \geq n_2 \\ \mathbb{E}\left[AB\right] = 0 & n_1 < n_2 \end{cases}$$

$a_{x_2}[n_1, n_2] \neq a_{x_2}[0, n_1 - n_2]$, so $x_2[n]$ is not WSS. $x_2 = B$ on the negative side and A on the positive side, so it is not periodic.

$$\begin{cases} x_3[n+1] = \frac{1}{2}x_3[n] + A \\ x_3[0] = A \end{cases}$$

We can see that

$$
\begin{aligned}
x_3[0] &= A \\
x_3[1] &= \frac{1}{2}A + A \\
x_3[2] &= \frac{1}{2}\left(\frac{1}{2}A + A\right) + A \\
x_3[3] &= \frac{1}{2}\left(\frac{1}{2}\left(\frac{1}{2}A + A\right) + A\right) + A \\
&\dots \\
\Rightarrow x_3[n] &= A\sum_{i=0}^{n}\left(\frac{1}{2}\right)^i
\end{aligned}
$$

By geometric series

$$x_3[n] = A\frac{1-\left(\frac{1}{2}\right)^{n+1}}{1-\frac{1}{2}} = 2A(1-2^{-n-1}) = A(2-2^{-n})$$

So

$$\mathbb{E}\left[x_3[n]\right] = (2-2^{-n-1})\mathbb{E}\left[A\right] = 0$$

We have

$$a_{x_3}[n_1, n_2] = \mathbb{E}\left[x_3[n_1]x_3[n_2]\right] = (2-2^{-n_1})(2-2^{-n_2})\mathbb{E}\left[A^2\right] = (2-2^{-n_1})(2-2^{-n_2})$$

and

$$a_{x_3}[0, n_1-n_2] = \mathbb{E}\left[x_3[0]x_3[n_1-n_2]\right] = (2-2^{-n_1+n_2})\mathbb{E}\left[A^2\right] = (2-2^{-n_1+n_2})$$

$a_{x_3}[n_1, n_2] \neq a_{x_3}[0, n_1-n_2]$, so $x_2[n]$ is not WSS. Since $x_3[n]$ is a geometric series, it is not periodic.

(b)

$$x_1[n] = \begin{cases} A & \text{even } n \\ B & \text{odd } n \end{cases}$$

We can see that $x_1[n+1]$ only depends on $x_1[n-1]$ as the values alternate between A and B. Therefore, $w = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ and prediction error is 0.

$$x_2[n] = \begin{cases} A & n \geq 0 \\ B & n < 0 \end{cases}$$

If $n \neq -1$ then $x_2[n+1] = x_2[n]$ and there is no prediction error. If $n = -1$ then the prediction error is $\mathbb{E}\left[x_2[0] \mid x_2[-1], x_2[-2]\right]$. Since $x_2[0] = A$, $x_2[-1] = x_2[-2] = B$, and A and B are independent, $\mathbb{E}\left[x_2[0] \mid x_2[-1], x_2[-2]\right] = \mathbb{E}\left[A\right] = 0$. Hence, $w = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and prediction error is 0.

$$\begin{cases} x_3[n+1] = \frac{1}{2}x_3[n] + A \\ x_3[0] = A \end{cases}$$

We have

$$\begin{aligned} x_3[n+1] - x_3[n] &= \frac{1}{2}x_3[n] - \frac{1}{2}x_3[n-1] \\ \Leftrightarrow x_3[n+1] &= \frac{3}{2}x_3[n] - \frac{1}{2}x_3[n-1] \end{aligned}$$

Hence, $w = \begin{bmatrix} \frac{3}{2} \\ -\frac{1}{2} \end{bmatrix}$ and prediction error is 0.

3 Adaptive Filter and LMS

(a) We are given the model $\mathbb{E}\left[x[0]x[m]\right] = 2^{-|m|} + 4^{-|m|} = a_x[m]$, therefore we can use probabilistic cost function for this problem.

(b) In general

$$\begin{aligned} R_x &= \mathbb{E}\left[X[n]X[n]^\top\right] \\ &= \begin{bmatrix} a_x[0] & a_x[1] & a_x[2] & \cdots & a_x[L-1] \\ a_x[1] & a_x[0] & a_x[1] & \cdots & a_x[L-2] \\ a_x[2] & a_x[1] & a_x[0] & \cdots & a_x[L-3] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_x[L-1] & a_x[L-2] & a_x[L-3] & \cdots & a_x[0] \end{bmatrix} \end{aligned}$$

and the cost function

$$C(w) = \mathbb{E}\left[|e[n]|^2\right]$$

where $e[n] = y[n] - d[n]$ is the prediction error, $y[n]$ is the prediction, and $d[n]$ is the reference.

Let $X[n] = \begin{bmatrix} x[n] & x[n-1] & x[n-2] & x[n-3] & \cdots & x[n-L+1] \end{bmatrix}^\top$.

For $L \geq 3$,

$$\begin{aligned} y[n] &= w^\top X[n] \\ d[n] &= \alpha_1 x[n-1] + \alpha_2 x[n-2] = \begin{bmatrix} 0 & \alpha_1 & \alpha_2 & 0 & \cdots & 0 \end{bmatrix} X[n] = A^\top X[n] \\ \Rightarrow e[n] &= y[n] - d[n] = (w-A)^\top X[n] \end{aligned}$$

The cost function is

$$C(w) = \mathbb{E}\left[|e[n]|^2\right] = (w-A)^\top R_x (w-A)$$

Therefore, $\min C(w) = 0$ for $w = A$. Hence, $w_{opt} = \begin{bmatrix} 0 & \alpha_1 & \alpha_2 & 0 & \cdots & 0 \end{bmatrix}$

For $L = 2$

$$R_x = \begin{bmatrix} a_x[0] & a_x[1] \\ a_x[1] & a_x[0] \end{bmatrix} = \begin{bmatrix} 2 & \frac{3}{4} \\ \frac{3}{4} & 2 \end{bmatrix}$$

$$\begin{aligned}
R_{xd} &= \mathbb{E}\left[X[n]d[n]\right] \\
&= \mathbb{E}\left[\begin{bmatrix} x[n] \\ x[n-1] \end{bmatrix} (\alpha_1 x[n-1] + \alpha_2 x[n-2])\right] \\
&= \begin{bmatrix} \alpha_1 \mathbb{E}\left[x[n]x[n-1]\right] + \alpha_2 \mathbb{E}\left[x[n]x[n-2]\right] \\ \alpha_1 \mathbb{E}\left[x[n-1]x[n-1]\right] + \alpha_2 \mathbb{E}\left[x[n-1]x[n-2]\right] \end{bmatrix} \\
&= \begin{bmatrix} \alpha_1 a_x[1] + \alpha_2 a_x[2] \\ \alpha_1 a_x[0] + \alpha_2 a_x[1] \end{bmatrix} \\
&= \begin{bmatrix} \frac{3}{4}\alpha_1 + \frac{5}{16}\alpha_2 \\ 2\alpha_1 + \frac{3}{4}\alpha_2 \end{bmatrix}
\end{aligned}$$

$$\begin{aligned}
\gamma_d &= \mathbb{E}\left[d[n]^2\right] \\
&= \alpha_1^2 \mathbb{E}\left[x[n-1]x[n-1]\right] + \alpha_2^2 \mathbb{E}\left[x[n-2]x[n-2]\right] + 2\alpha_1\alpha_2 \mathbb{E}\left[x[n-1]x[n-2]\right] \\
&= \alpha_1^2 a_x[0] + \alpha_2^2 a_x[0] + 2\alpha_1\alpha_2 a_x[1] \\
&= 2(\alpha_1^2 + \alpha_2^2) + \frac{3}{2}\alpha_1\alpha_2
\end{aligned}$$

$$C(w) = \gamma_d - 2w^\top R_{xd} + w^\top R_x w$$

$$\begin{aligned}
w_{opt} = R_x^{-1} R_{xd} &= \frac{4}{55}\begin{bmatrix} 8 & -3 \\ -3 & 8 \end{bmatrix} \begin{bmatrix} \frac{3}{4}\alpha_1 + \frac{5}{16}\alpha_2 \\ 2\alpha_1 + \frac{3}{4}\alpha_2 \end{bmatrix} \\
&= \frac{4}{55}\begin{bmatrix} \frac{1}{4}\alpha_2 \\ \frac{55}{4}\alpha_1 + \frac{81}{16}\alpha_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{55}\alpha_1 \\ \alpha_1 + \frac{81}{220}\alpha_2 \end{bmatrix}
\end{aligned}$$

For $L = 1$

$$
\begin{aligned}
R_x &= a_x[0] = 2 \\
R_{xd} &= \alpha_1 a_x[1] + \alpha_2 a_x[2] = \frac{3}{4}\alpha_1 + \frac{5}{16}\alpha_2 \\
\gamma_d &= 2(\alpha_1^2 + \alpha_2^2) + \frac{3}{2}\alpha_1\alpha_2 \\
C(w) &= \gamma_d - 2w^\top R_{xd} + w^\top R_x w \\
&= 2(\alpha_1^2 + \alpha_2^2) + \frac{3}{2}\alpha_1\alpha_2 - \left(\frac{3}{2}\alpha_1 + \frac{5}{8}\alpha_2\right) w + 2w^2 \\
w_{opt} &= R_x^{-1} R_{xd} = \frac{3}{8}\alpha_1 + \frac{5}{32}\alpha_2
\end{aligned}
$$

(c) The gradient of selected cost

$$
\begin{aligned}
\nabla_w C(\hat{w}) &= \nabla_w \mathbb{E}\left[|e[n]|^2\right] \\
&= \mathbb{E}\left[2e[n]\nabla_w e[n]\right] && \text{(chain rule)} \\
&= -2\mathbb{E}\left[e[n]X[n]\right] && (\nabla_w e[n] = -X[n])
\end{aligned}
$$

The gradient descent update equation (with μ as the learning rate)

$$
\hat{w}[n+1] = \hat{w}[n] - \frac{1}{2}\mu\nabla_w C(\hat{w}[n]) = \hat{w}[n] + \mu\mathbb{E}\left[e[n]X[n]\right]
$$

converges to a local minimum if $C(w)$ is strictly convex (R_x is invertible) and differentiable. Indeed, if $\hat{w}[n] \to \hat{w}$ converges then $\hat{w}[n+1]$ to the same limit, the gradient equation becomes

$$
\hat{w} = \hat{w} - \frac{1}{2}\mu\nabla_w C(\hat{w}) \Rightarrow \nabla_W C(\hat{w}) = 0
$$

which is a characterization of a local minimum of $C(w)$.

(d) In LMD, we assume that $\mathbb{E}\left[e[n]X[n]\right] \approx e[n]X[n]$. Therefore, the LMS update equations are

$$
\begin{aligned}
e[n] &= d[n] - \hat{w}[n]^\top X[n] \\
\hat{w}[n+1] &= \hat{w}[n] + \mu X[n]e[n]
\end{aligned}
$$

We have

$$
\nabla_w C(\hat{w}) = -2\mathbb{E}\left[X[n](d[n] - \hat{X}[n]^\top \hat{w})\right] = -w(R_{xd} - R_x\hat{w})
$$

so that the ideal iterations are

$$\hat{w}[n+1] = (I - \mu R_x)\hat{w}[n] + \mu R_{xd}$$

This is a linear difference equation in the vector $\hat{w}[n]$. Such difference equation has a convergent solution iff the eigenvalues of $I - \mu R_x$ are contained in the unit circle. The eigenvalues of $I - \mu R_x$ are given by

$$\lambda_k = 1 - \mu\psi_k, \qquad k = 1, \cdots, L$$

where $\psi_1 < \psi_2 \leq \cdots \leq \psi_L$ are the eigenvalues of R_x, sorted by increasing order. We want

$$\begin{aligned} & -1 < \lambda_k < 1 \\ \Leftrightarrow & -1 < 1 - \mu\psi_k < 1 \\ \Leftrightarrow & 1 > \mu\psi_k - 1 > -1 \\ \Leftrightarrow & 0 < \mu\psi_k < 2 \\ \Rightarrow & 0 < \mu < \frac{2}{\psi_L} \end{aligned}$$

where ψ_L is the largest eigenvalue of R_x. Since $tr(R_x) = \sum_{k=1}^{L} \psi_k \geq \psi_L$

$$0 < \mu < \frac{2}{tr(R_x)}$$

This does not guarantee convergence of $\hat{w}[n]$ because R_x is assumed to be invertible.

4 Regularized Wiener Filter and Leaky LMS

(a) We want to solve w for

$$R_x w = R_{xd}$$

If R_x is singular, it is not invertible and therefore LMS will diverge.

(b) To avoid singularity, we can add a regularization term to the cost function, i.e.

$$\begin{aligned} C(w) &= \mathbb{E}\left[|e[n]|^2\right] + \lambda \|w\|^2 \\ &= w^\top R_x w - 2w^\top R_{xd} + \gamma_d + \lambda w^\top w \\ &= w^\top (R_x + \lambda I)w - 2w^\top R_{xd} + \alpha \end{aligned}$$

Therefore, the gradient is

$$\nabla C(w) = 2\left((R_x + \lambda I)w - R_{xd}\right)$$

and

$$(R_x + \lambda I)w = R_{xd} \Rightarrow w_{opt} = (R_x + \lambda I)^{-1} R_{xd}$$

(c) If R_x is singular, its eigenvalues are zero. By adding λ, we can shift the eigenvalues to λ to have it invertible, where the inverse is unique.

(d) For leaky LMS, we simply add the regularization term to the cost function, i.e.

$$\begin{aligned} C_{reg}(w) &= C(w) + \lambda \|w\|^2 \\ \Rightarrow \nabla C_{reg}(w) &= \nabla C(w) + \lambda \|w\|^2 \\ &\approx -2X[n]e[n] + \lambda \|w\|^2 \end{aligned}$$

Therefore, the update equation is

$$\hat{w}[n+1] = \hat{w}[n] - \frac{1}{2}\mu \nabla C_{reg}(w) = \hat{w}[n] + \mu \left(X[n]e[n] - \frac{\lambda}{2} \|w\|^2 \right)$$

(e) We have

$$\begin{aligned} a_x[k] &= \frac{3}{4} + \frac{1}{4}(-1)^k \\ \Rightarrow a_x[k] &= \begin{cases} 1 & k \text{ even} \\ \frac{1}{2} & k \text{ odd} \end{cases} \end{aligned}$$

For $L = 3$

$$R_x = \begin{bmatrix} a_x[0] & a_x[1] & a_x[2] \\ a_x[1] & a_x[0] & a_x[1] \\ a_x[2] & a_x[1] & a_x[0] \end{bmatrix} = \begin{bmatrix} 1 & \frac{1}{2} & 1 \\ \frac{1}{2} & 1 & \frac{1}{2} \\ 1 & \frac{1}{2} & 1 \end{bmatrix}$$

For one step prediction, $d[n] = x[n+1]$, therefore

$$R_{xd} = a_x[1]a_x\vec{[}2]a_x[3] = \begin{bmatrix} \frac{1}{2} \\ 1 \\ \frac{1}{2} \end{bmatrix}$$

Wiener filter of x is $w_{opt} = R_x^{-1} R_{xd}$. However, R_x is singular. We can use the pseudo-inverse of R_x instead, i.e.

$$R_x^{\dagger} = \begin{bmatrix} \frac{1}{3} & -\frac{1}{3} & \frac{1}{3} \\ -\frac{1}{3} & \frac{4}{3} & -\frac{1}{3} \\ \frac{1}{3} & -\frac{1}{3} & \frac{1}{3} \end{bmatrix}$$

$$w_p = R_x^{\dagger} R_{xd} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

With $\lambda = 0.1$

$$R_x + \lambda I = \begin{bmatrix} 1.1 & 0.5 & 1 \\ 0.5 & 1.1 & 0.5 \\ 1 & 0.5 & 1.1 \end{bmatrix}$$

$$w_l = (R_x + \lambda I)^{-1} R_{xd} \approx \begin{bmatrix} 0.0276 \\ 0.8840 \\ 0.0276 \end{bmatrix}$$

5 Python Problem - Wiener's LMS

(a) Figure 1 illustrates the prediction problem as an adaptive filter diagram, where the input is $x[n]$, the reference $d[n] = x[n+1] = \alpha x[n] + s[n] - 0.5s[n-1]$, and the cost function $C(w) = \mathbb{E}\left[|e[n]|^2\right] = \mathbb{E}\left[|d[n] - y[n]|^2\right] = \mathbb{E}\left[\left|d[n] - w^\top X[n]\right|^2\right]$.

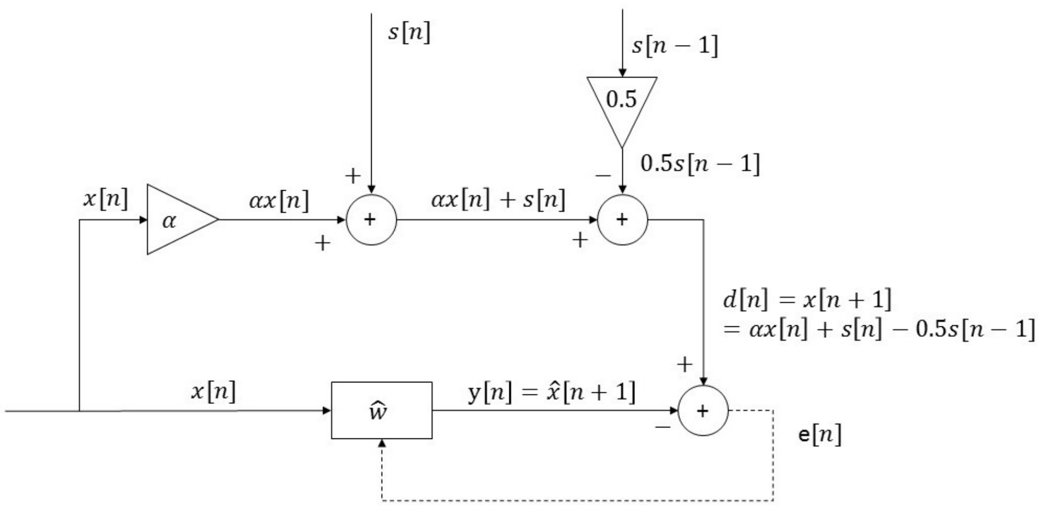

Figure 1: Adaptive filter diagram

(b) We have

$$x[n+1] = \alpha x[n] + s[n] - 0.5s[n-1]$$

Therefore, its Z-transform is

$$
\begin{aligned}
X(z)z &= \alpha X(z) + S(z) - 0.5S(z)z^{-1} \\
\Leftrightarrow X(z)(z-\alpha) &= S(z)\frac{1-0.5z^{-1}}{z-\alpha} \\
\Rightarrow H(z) &= \frac{1-0.5z^{-1}}{z-\alpha}
\end{aligned}
$$

Since $A_s(z) = 1$,

$$
\begin{aligned}
A_x(z) &= H(z)H(z^{-1}) \\
&= \frac{1-0.5z^{-1}}{z-\alpha}\frac{1-0.5z}{z^{-1}-\alpha} \\
&= (1.25 - 0.5z - 0.5z^{-1})\frac{1}{1-\alpha z+\alpha z^{-1}-\alpha^2} \\
&= (1.25 - 0.5z - 0.5z^{-1})\frac{1}{1-\alpha z}\frac{1}{1-\alpha z^{-1}}
\end{aligned}
$$

Let $P(z) = 1.25 - 0.5z - 0.5z^{-1}$ and $G(z) = \frac{1}{1-\alpha z}\frac{1}{1-\alpha z^{-1}}$. We notice that, by geometric series

$$
\frac{1}{1-\alpha z} = \sum_{m=0}^{\infty}(\alpha z)^n
$$

$$
\frac{1}{1-\alpha z^{-1}} = \sum_{m=0}^{\infty}\alpha^m z^{-m}
$$

Therefore

$$
G(z) = \sum_{m\geq 0, n\geq 0}\alpha^{n+m}z^{n-m}
$$

Let $g[\cdot]$ be the inverse z-transform of $G(z)$. We see that $g[0]$ is the sum subject to $n - m = 0 \Leftrightarrow n = m$, i.e.

$$
g[0] = \sum_{m\geq 0}\alpha^{n+m} = \sum_{m\geq 0}\alpha^{2m}
$$

In general

$$
g[k] = \sum_{m\geq 0}\alpha^{2m+k}
$$

Therefore

$$
a_x[k] = \sum_{m\geq 0}(1.25\alpha^{2m+k} - 0.5\alpha^{2m+k+1} - 0.5\alpha^{2m+k-1})
$$

(c) Figure 2-7 show the results of different approaches with varied choices of α and L (in Figure 3, LMS result is almost the same as original signal, due to scaling). Overall, LMS approach has the best result and probabilistic approach does not work well with $\alpha = 0$. Moreover, larger L also gives better approximation.

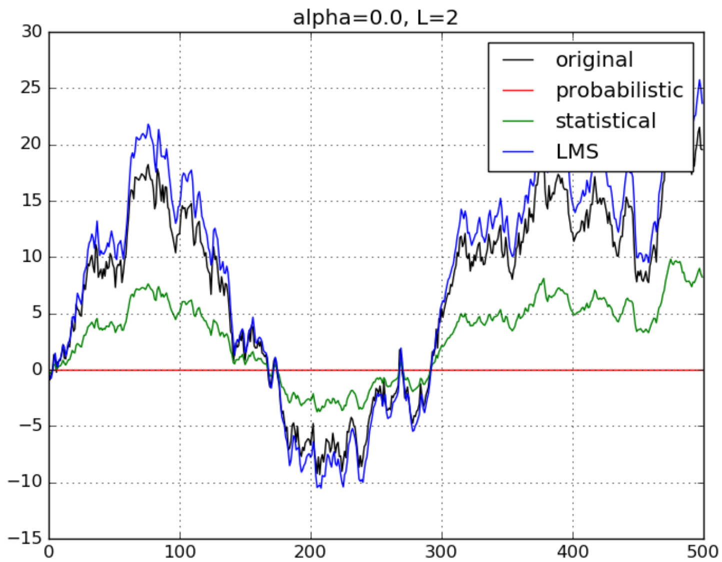

Figure 2: $\alpha = 0, L = 2$.

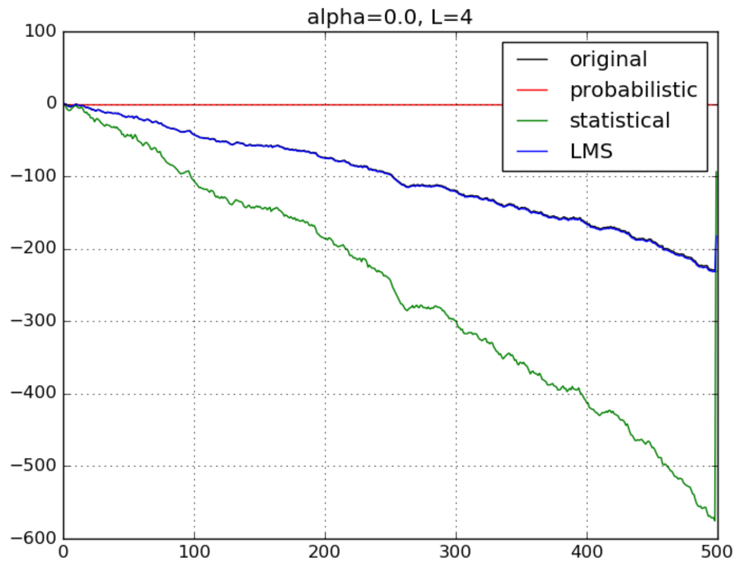

Figure 3: $\alpha = 0, L = 4$.

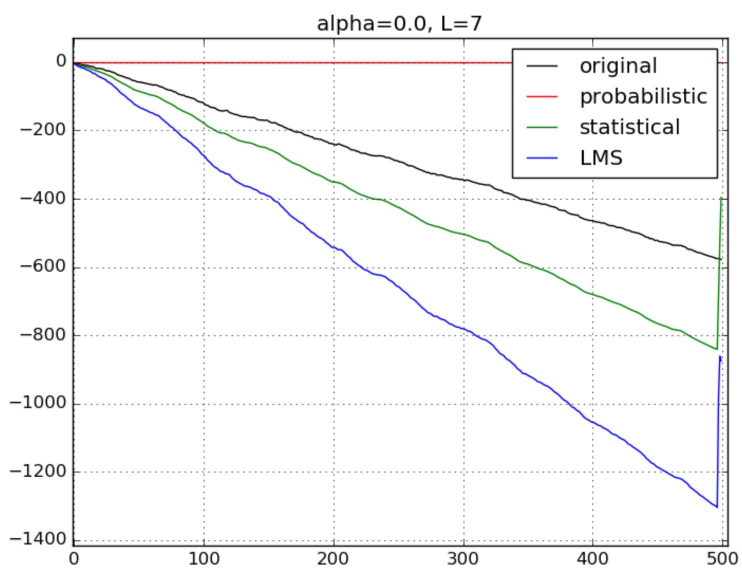

Figure 4: $\alpha = 0, L = 7$.

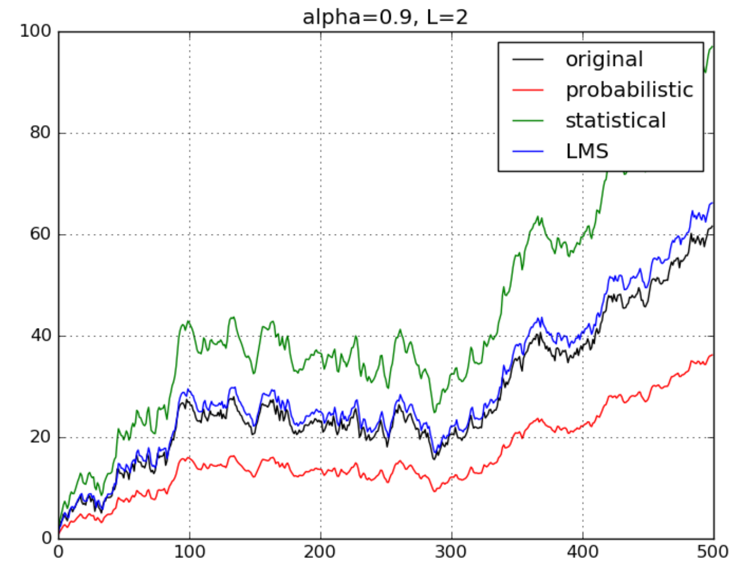

Figure 5: $\alpha = 0.9, L = 2$.

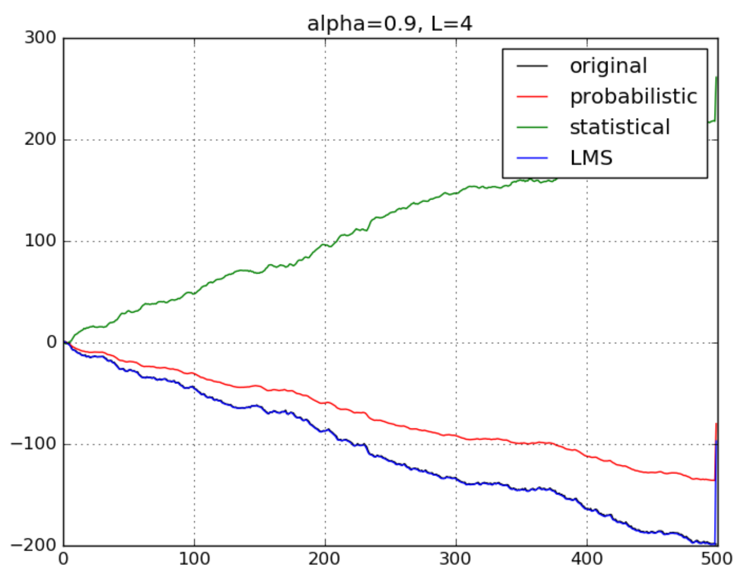

Figure 6: $\alpha = 0.9, L = 4$.

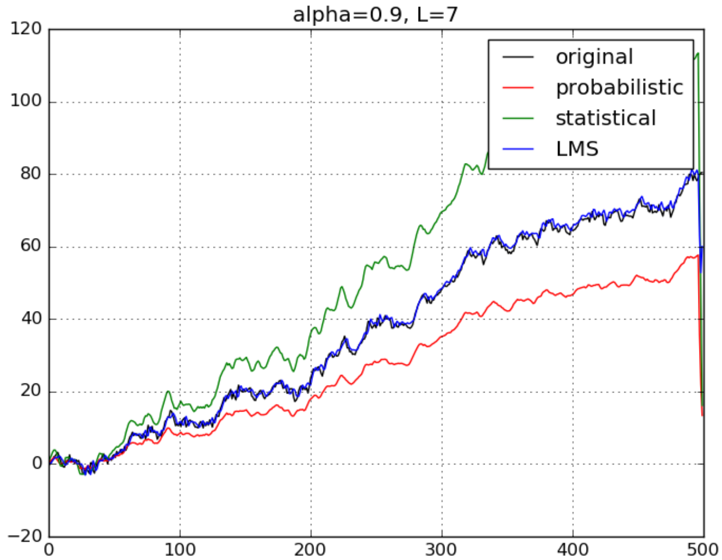

Figure 7: $\alpha = 0.9, L = 7$.

6 Python Problem - AR System Identification

The signal is approximated using LMS algorithm. To find the best L, the script runs through multiple L's and choose the one with the lowest MSE. The script also has regularization to avoid divergence, where λ is selected among $\{1, 1^{-1}, ..., 1^{-9}\}$ by the lowest MSE. The result is showed in Figure 8, where the optimal setting is $L = 1$ giving $w \approx 0.9514$ and $MSE \approx 0.0540$

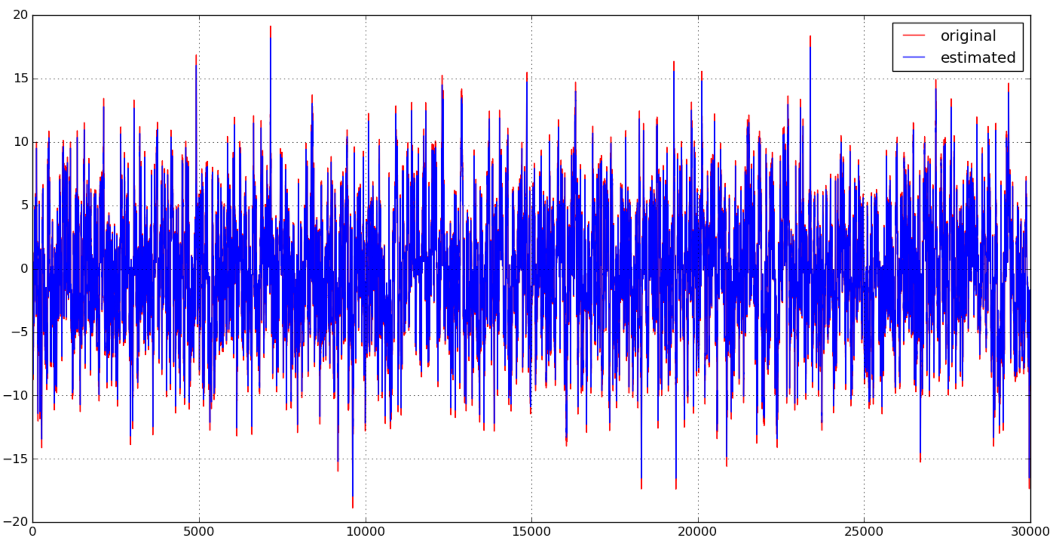

Figure 8: Best LMS result with $L = 1, w \approx 0.9514$, and $MSE \approx 0.0540$.