Fall 2020-21

EE 503

Homework #2

(Due: Dec. 30, 2020, 17:30)

Problem: Solve Computer Exercises C4.3 (page 212) of Hayes (also given in the next page).

Notes:

1. Read Section 4.4.5 (page 166) before attempting the problem.
2. The MATLAB file spike.m is given in page 172 of Hayes.
3. In part f), generate noisy $\hat{g}[n]$ and design the least square (LS) equalization filter $h_N(n)$ with the knowledge $\hat{g}[n]$ (instead of $g[n]$) using spike.m as in part b. Then, use the equalization filter $h_N(n)$ designed with the noisy data on the input $y[n] = g[n] * x[n]$ (generated with the noise-free $g[n]$) and check whether the mismatch of $g[n]$ and $\hat{g}[n]$ is important or not.
4. BONUS: In part f), study the total least squares (TLS) method from internet sources and redo part f) with the TLS estimate. Comment whether LS- and TLS-based solutions make a difference in this problem. (You may change $w[n]$ which is distributed uniformly in $[-0.005, 0.005]$ to a wider or smaller range to see the difference the effect on LS- and TLS-based solutions.)

C4.3. In this problem, we look briefly at the problem of deconvolution using FIR least squares inverse filters. Suppose that we have recorded a signal, $y(n)$, that is known to have been blurred by a filter having a unit sample response

$$g(n) = \begin{cases} \cos\big(0.2[n-25]\big)\exp\big\{-0.01[n-25]^2\big\} & ; \quad 0 \le n \le 50 \\ 0 & ; \quad \text{otherwise} \end{cases}$$

The signal that is to be recovered from $y(n)$ is a sequence of impulses,

$$x(n) = \sum_{k=1}^{10} x(k)\delta(n - n_k)$$

where the values of $x(k)$ and n_k are as listed in the following table.

n_k	25	40	55	65	85	95	110	130	140	155
$x(k)$	1	0.8	0.7	0.5	0.7	0.2	0.9	0.5	0.6	0.3

(a) Make a plot the observed signal $y(n) = g(n) * x(n)$ and determine how accurately the amplitudes and locations of the impulses in $x(n)$ may be estimated by simply looking at the peaks of $y(n)$.

(b) Using the m-file `spike.m`, design the least squares inverse filter $h_N(n)$ of length $N = 50$ that has the optimum spiking delay.

(c) Filter $y(n)$ with your optimum spiking filter and plot the output of the filter $\hat{x}(n) = h_N(n) * y(n)$. What are your estimated values for the amplitudes and locations of the impulses in $x(n)$?

(d) Your results in part (c) assume noise-free observations of $y(n) = g(n) * x(n)$. Suppose these measurements are noisy,

$$\tilde{y}(n) = g(n) * x(n) + v(n)$$

where $v(n)$ is white Gaussian noise with variance σ_v^2. Repeat part (c) using $\tilde{y}(n)$ with $\sigma_v^2 = .0001$ and $\sigma_v^2 = .001$ and comment on the accuracy of your estimates of $x(k)$ and n_k.

(e) As discussed in Problem 4.19, the effect of measurement noise may be reduced by incorporating a *prewhitening parameter* α in the design of the least squares inverse filter. Write a MATLAB m-file or modify `spike.m` to allow for noise reduction in the least squares inverse filter design. Using this m-file, repeat your experiments in part (d) using different values for the prewhitening parameter. Comment of the effectiveness of α in reducing the noise. What values for α seem to work the best?

(f) Your results in parts (b) and (c) assume perfect knowledge of $g(n)$. Repeat the design of your least squares inverse filter assuming that $g(n)$ has been measured in the presence of noise, i.e., you are given

$$\hat{g}(n) = g(n) + w(n)$$

where $w(n)$ is white noise that is uniformly distributed between $[-.005, .005]$. Filter $y(n)$ with your optimum spiking filter and plot the output of the filter $y\hat{x}(n) = h_N(n) * y(n)$. How accurate are your estimates of the amplitudes and locations of the impulses in $x(n)$?