Name: Instructor's solutions RIN: ____________

Rensselaer Polytechnic Institute
Department of Electrical, Computer, and Systems Engineering
ECSE 4530: Digital Signal Processing, Fall 2019

Final. Closed book, closed notes.
December 17, 2019, 3:00-6:00 PM

Show all work for full credit.

- Calculators are allowed. Other electronic devices are not permitted.
- Reduce expressions involving exponentials, sines, and cosines as much as possible.
- If the sinc function is needed, use the definition $\mathrm{sinc}(x) = \frac{\sin x}{x}$.
- Geometric series formula: $\sum_{n=0}^{\infty} a^n = \frac{1}{1-a}, \quad |a| < 1.$
- Finite sum formula: $\sum_{n=M}^{N-1} a^n = \frac{a^M - a^N}{1-a}, \quad a \neq 1.$

When in doubt, show your work. Good luck!

1		30
2		25
3		30
4		25
5		25
6		35
7		30
Total		200

1. (30 points.) True/false and fill in the blank questions. You do not need to explain your reasoning. Read each question carefully.

   1. F The computational complexities of computing FFT and inverse FFT are not the same.
   2. F DFT matrix is real valued.
   3. T DFT is also known as DTFS (discrete time Fourier series).
   4. T The least squares (LS) and the recursive least squares (RLS) algorithms are deterministic approximations to the Wiener filter (if the processes are wide sense stationary).
   5. T Using linear phase filters, we can design low pass, band pass or high pass filters.
   6. F In practice, sinc interpolation is very easy to implement.
   7. T Region of convergence (ROC) for which the z-transform converges cannot contain any poles.
   8. T We can apply the bilinear transformation or the impulse invariance method to convert an analog filter to a digital filter.
   9. F Gradient descent algorithm always converges to the global minimum.
   10. Wiener filters are optimal linear discrete-time filters that are used to produce the minimum mean-square error (MMSE) of stationary signal and noise processes.
   11. Rounding and analog-to-digital conversion are examples of quantization.
   12. The least mean square (LMS) algorithm is a stochastic gradient descent to find filter taps that minimizes an error function.
   13. As long as $x[n] = x_c(nT)$ was obtained by sampling without aliasing, we can upsample by any integer factor.
   14. Moving average model can be regarded as smoothing the data.
   15. When we design a digital IIR filter using the impulse invariance approach, aliasing can be a big problem if the sampled signal contains high frequencies.

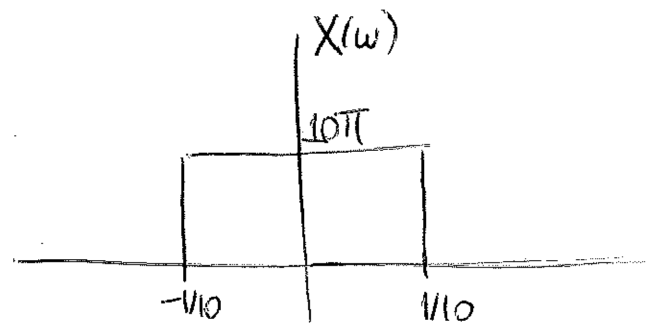
2. (25 points.) Sampling. We wish to modulate a signal $x(t) = \text{sinc}\left(\frac{t}{10}\right)$ where t is in seconds. We use a new type of modulator which produces a transmit signal $s(t) = x(t) \cdot [\cos(\omega_0 t) + \cos(2\omega_0 t)]$.

   (a) (3 pts) Plot the Fourier Transform $X(\omega)$.

   Hint:

$$\frac{\sin(At)}{\pi t} \overset{FT}{\longleftrightarrow} \begin{cases} 1, & |\omega| \leq A \\ 0, & |\omega| > A. \end{cases}$$

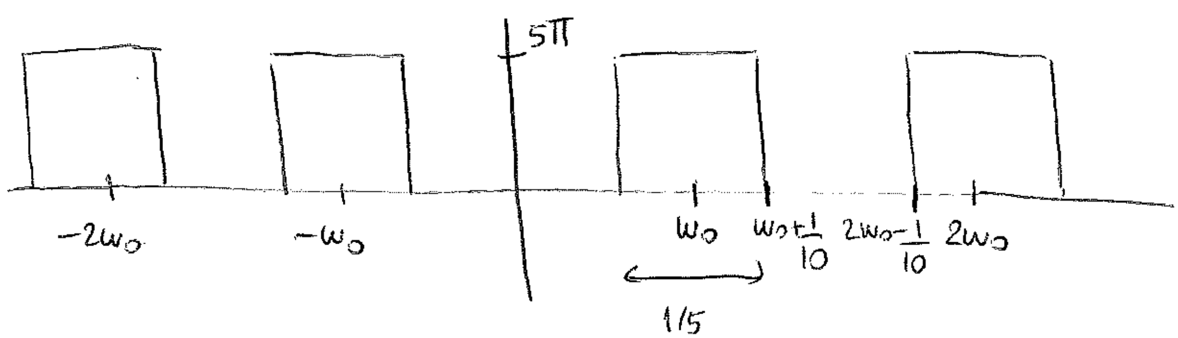
$$x(t) = \frac{\sin\left(\frac{t}{10}\right)}{\frac{t}{10}} = \frac{10\pi \sin\left(\frac{t}{10}\right)}{\pi t} \overset{FT}{\longleftrightarrow} \begin{cases} 10\pi, & |\omega| \leq \frac{1}{10} \\ 0, & |\omega| > \frac{1}{10} \end{cases}$$



   (b) (5 pts) Plot $S(\omega)$ for the output of the modulator. What is the requirement on the value ω_0 for $x(t)$ to be recoverable from $s(t)$?

$$S(\omega) = \frac{1}{2\pi} \sum_{k=1}^{2} \left[\pi X(\omega - k\omega_0) + \pi X(\omega + k\omega_0)\right]$$

$$\cos\omega_0 t \overset{FT}{\longleftrightarrow} \pi\,\delta(\omega - \omega_0) + \pi\,\delta(\omega + \omega_0)$$



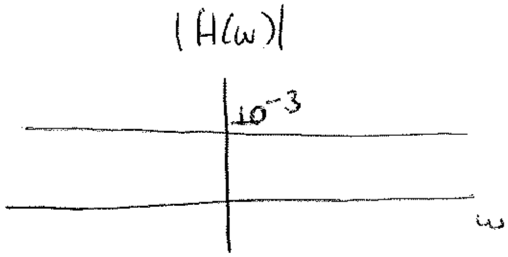
$$\omega_0 + 1/10 < 2\omega_0 - 1/10$$

$$\boxed{1/5 < \omega_0}$$

(c) (6 pts) The signal $s(t)$ is sent through a channel that attenuates it by 60 dB and delays it by 0.1 seconds.

i. What is the impulse response $h(t)$ of this channel?

ii. What is the output $y(t)$ in terms of $x(t)$?



60dB $\rightarrow 10^6$ power gain

$\Rightarrow h(t) = 10^{-3}\delta(t-0.1)$

delay

$y(t) = s(t) * h(t)$

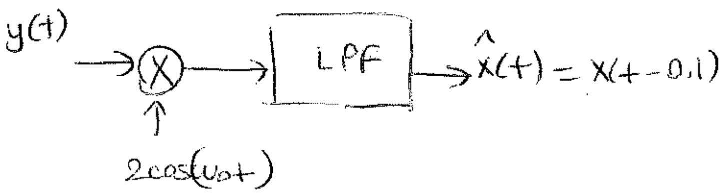
$= 10^{-3} s(t-0.1) = 10^{-3} x(t-0.1)\left[\cos(\omega_0(t-0.1)) + \cos(2\omega_0(t-0.1))\right]$

(d) (6 pts) Design a filter to recover $x(t)$ perfectly (some delay is acceptable) from $y(t)$, assuming the condition you found in part (b) is met.

$\omega_0 > 1/5$ from (b).

One option is $y(t) \cdot 2\cos(\omega_0 t) \overset{FT}{\Longleftrightarrow} Y(\omega-\omega_0) + Y(\omega+\omega_0)$

then use a LPF with cutoff $\frac{1}{10}$ and gain 10^3



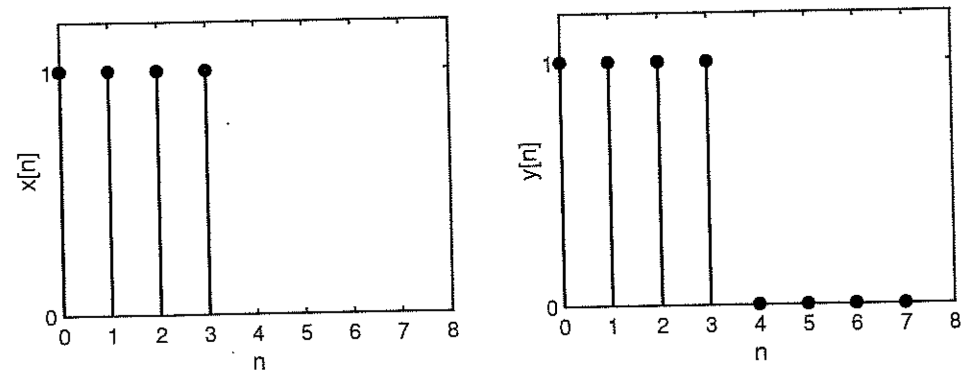
(e) (5 pts) If we were to instead sample the output $y(t)$ to get a digital signal $y[n]$ for processing (rather than using the above filter), what would be the minimum required sampling frequency according to the sampling theorem?

Maximum frequency in $Y(\omega)$ is $2\omega_0 + \frac{1}{10}$.

$$\omega_s > \underbrace{2\left(2\omega_0 + \frac{1}{10}\right)}_{\text{The Nyquist rate}}$$

3. (30 points.) Circular convolution.

Consider the two discrete-time signals, $x[n]$ and $y[n]$. The first signal $x[n]$ is a rectangular pulse of length 4. The second signal $y[n]$ is the sequence $x[n]$ zero-padded to length 8. They are shown below.



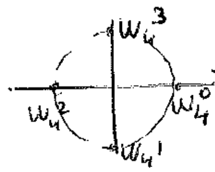
In the following questions, you can either compute or plot. Make sure your plots are labeled correctly.

(a) (5 pts) What is the magnitude of the length 4 DFT of $x[n]$, i.e. $|X[k]|$?

$$X[k] = \sum_{n=0}^{3} x[n] W_4^{nk} = \sum_{n=0}^{3} W_4^{nk}$$

$$X[0] = 4$$

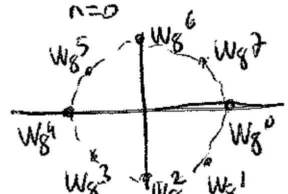
$$X[1] = X[2] = X[3] = 0$$



$$\sum_{n=0}^{3} W_N^{n} = 0$$

(b) (5 pts) What is the magnitude of the length 8 DFT of $y[n]$, i.e. $|Y[k]|$?

$$Y[k] = \sum_{n=0}^{7} y[n] W_8^{nk} = \sum_{n=0}^{3} W_8^{nk}$$



$$Y[0] = 4$$

$$Y[2] = Y[4] = Y[6] = 0 = X[1] = X[2] = X[3]$$

$$Y[1] = \sum_{n=0}^{3} W_8^{n} \qquad Y[5] = \sum_{n=0}^{3} W_8^{5n}$$

$$Y[3] = \sum_{n=0}^{3} W_8^{3n} \qquad Y[7] = \sum_{n=0}^{3} W_8^{7n}$$

$$\Downarrow$$

$$|Y[1]| = |Y[7]|$$

$$|Y[3]| = |Y[5]|$$

(c) (5 pts) What is the length 8 circular convolution of $x[n]$ with itself?

$$DFT\{x[n] \circledast x[n]\} = X[k]X[k]$$

$$\begin{matrix} \parallel \\ x_c[n] \end{matrix}$$

$$x_c[n] = \frac{1}{N} \sum_{k=0}^{N-1} X^2[k] W_N^{-nk} = \frac{1}{N} \cdot 16 = 2, \quad n = 0, \ldots, 7$$

(d) (5 pts) What is the length 8 circular convolution of $y[n]$ with itself?

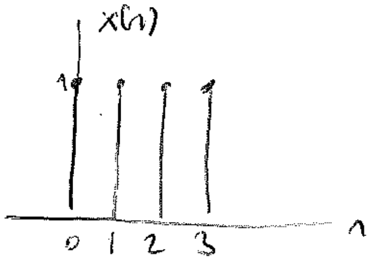
$$\mathrm{DFT}\{\underbrace{y[n] \circledast y[n]}_{y_c[n]}\} = Y[k]\,Y[k]$$

unfortunately this is not easy to compute.

$$y_c[n] = \frac{1}{N}\sum_{k=0}^{N-1} Y[k]^2 W_N^{-nk} = \frac{1}{8}\left(Y[1]^2 W_8^{-n} + Y[3]^2 W_8^{-3n} + Y[5]^2 W_8^{-5n} + Y[7]^2 W_8^{-7n}\right)$$

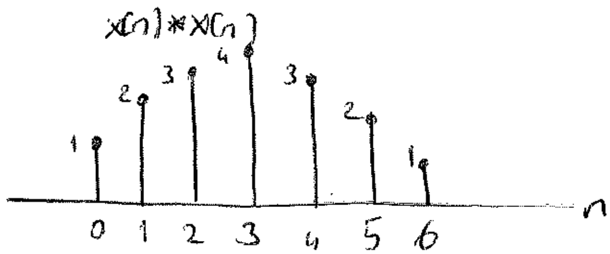
$$= \sum_{k=0}^{7} y[k]\,y[((n-k))_8] = \sum_{k=0}^{3} y[((n-k))_8] \Rightarrow \begin{cases} y_c[0] = y[0] = 1 \\ y_c[1] = y[1] + y[0] = 2 \\ y_c[2] = y[2] + y[1] + y[0] = 3 \\ y_c[3] = y[3] + y[2] + y[1] + y[0] = 4 \\ y_c[4] = \cancel{y[4]} + y[3] + y[2] + y[1] = 3 \\ y_c[5] = \cancel{y[5] + y[4]} + y[3] + y[2] = 2 \\ y_c[6] = \cancel{y[6] + y[5] + y[4]} + y[3] = 1 \end{cases}$$

(e) (5 pts) What is the linear convolution of $x[n]$ with itself?



$x[n]$ is length 4

$x[n] * x[n]$ is length $4 + 4 - 1 = 7$



(f) (5 pts) Which convolutions for the discrete-time sequences you obtained in parts (c)-(e) are related and why?

We observe that the circular convolution in part d is the same as the linear convolution of part e.

Since $y[n]$ is zero-padded to length 8, which is greater than 7, the wrapped around signal will not have aliasing. Therefore, it will be equivalent to linear convolution.

4. (25 points.) Inverse z-transform.

Each part of this problem can be considered separately.

(a) (10 pts) The z-transform of a signal with ROC $|z| > 2$ is given as

$$X(z) = \frac{2z+1}{z^2 - 3z + 2}.$$

Determine $x[n]$.

$$X(z) = \frac{A}{z-2} + \frac{B}{z-1} = \frac{Az - A + Bz - 2B}{z^2 - 3z + 2}$$

$A + B = 2$

$-A - 2B = 1$

$B = -3$

$A = 5$

$x[n] = A2^{n-1} u[n-1] + B u[n-1]$ since ROC $|z| > 2$

$= (5 \cdot 2^{n-1} - 3) u[n-1]$

(b) (3 pts) Why do we need to define a region-of-convergence (ROC) for the z-transform?

z transform might not converge everywhere.

For example, when $x[n] = u[n] \longleftrightarrow X(z) = \frac{1}{1 - z^{-1}}, \; |z| > 1$

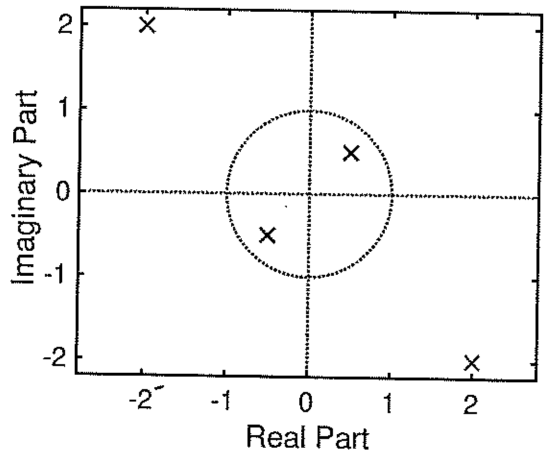
Why?

$$X(z) = \sum_{n=-\infty}^{+\infty} x[n] z^{-n} = \sum_{n=0}^{\infty} z^{-n} = \begin{cases} \frac{1}{1 - z^{-1}}, & |z| > 1 \\ \infty, & |z| < 1 \end{cases}$$

Other accepted answers:

We might have the same z-transform for two different time domain signals, if we do not define the ROC, it is not possible to distinguish them.

(c) (12 pts) Assume that the right sided signal $y[n]$ has a z-transform has 4 poles at $2-2j$, $-2+2j$, $0.5+0.5j$ and $-0.5-0.5j$. The corresponding z-plane is shown below. Determine the locations of the poles for the z-transform of $y[-n]$. Plot the corresponding z-plane. Your plot should also indicate the ROC.



Time reversal $x[n] \longleftrightarrow X(z^{-1})$ where ROC is inverted

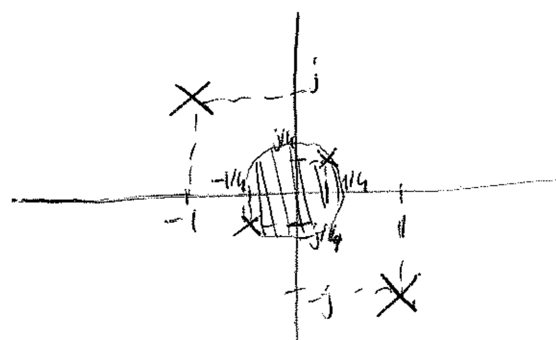
ROC of $X(z)$: $|z| > 2\sqrt{2}$

poles of $X(z^{-1})$: $\frac{1}{2-2j} = \frac{1}{4} + \frac{j}{4}$

$\frac{1}{-2+2j} = -\frac{1}{4} - \frac{j}{4}$

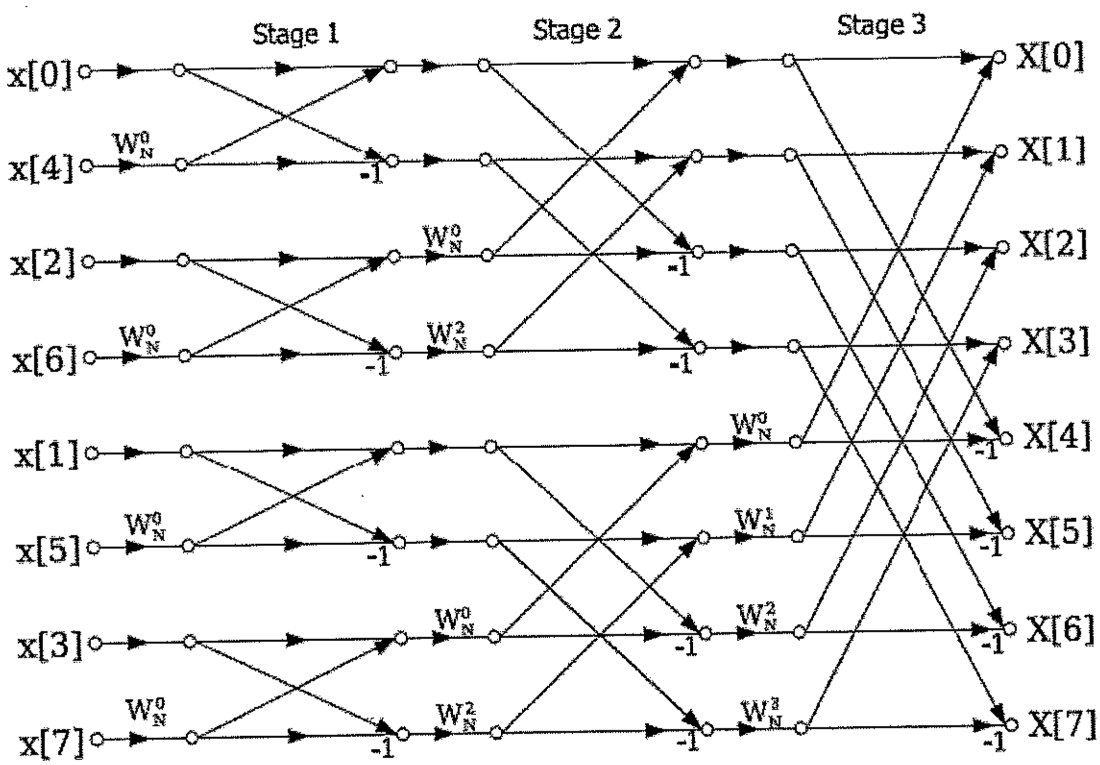
$\frac{1}{0.5+0.5j} = 1-j$

$\frac{1}{-0.5-0.5j} = -1+j$



ROC: $|z| < \frac{1}{2\sqrt{2}}$ z-plane

5. (25 points.) FFT Algorithm. Suppose we compute an $N = 8$-point DFT of $x[n]$, $n = 0, \ldots, N-1$ using the eight-point decimation-in-time FFT algorithm with the twiddle factors as shown below.



(a) (5 pts) For the above algorithm, how many multiplications you need at each stage? Explain.

As long as you explain your ~~reasoning~~ reasoning well, several acceptable answers:
- each butterfly operation requires one multiplication (due to the twiddle factor)
& each stage has 4 butterflies ⇒ This yields 4 multiplications per stage
- more rigorous answers suggest that since $W_N^0 = 1$, we only need 2 multiplications in stage 2, 3 multiplications in stage 3.

(b) (5 pts) Assume that $x[1] = x[3] = x[5] = x[7] = 0$. What does it imply in terms of the $N = 8$-point DFT of $x[n]$, i.e. $X[k]$? In this case, how many multiplications do you need in total to implement the FFT?

Note that when $x[1] = x[3] = x[5] = x[7] = 0$, we have that

$$\left.\begin{aligned} X[0] &= X[4] \\ X[1] &= X[5] \\ X[2] &= X[6] \\ X[3] &= X[7] \end{aligned}\right\} \text{For the 8-point DFT The period becomes 4}$$

This operation eliminates Stage 3 ⇒ acceptable ~~answer~~ answers are 8 multiplications in total, or 2 multiplications (only in stage 2)

(c) (7 pts) Now assume that the inputs are $X[k]$, $k = 0,\ldots,7$ and you want to compute $x[n]$, $n = 0,\ldots,7$. How would you modify the above algorithm if you are only allowed to change the branch gains? You can indicate the new gains on the plot above.

Note that inverse FFT is given as

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k] W_N^{-nk}$$

This is quite similar to FFT definition: $X[k] = \sum_{n=0}^{N-1} x[n] W_N^{nk}$

If we use the same algorithm, we just need to multiply the input with $\frac{1}{N}$ and replace W_N^{kn} by W_N^{-kn}.

(d) (8 pts) Now assume that $x[n]$ is a complex-valued sequence given as

$$x[n] = x_1[n] + jx_2[n], \quad n = 0,\ldots,7$$

where $x_1[n]$ and $x_2[n]$ are two real-valued sequences of length 8. Denote by $X_1[k]$ and $X_2[k]$ the DFTs of these sequences. Use DFT properties to determine $X_1[k]$ and $X_2[k]$ from $X[k]$. Show your work.

$x_1[n]$ is real $\Rightarrow X_1[k] = X_1^*[-k]$

$x_2[n]$ is real $\Rightarrow X_2[k] = X_2^*[-k]$

$x[n] = x_1[n] + jx_2[n] \xleftrightarrow{DFT} X[k] = X_1[k] + jX_2[k]$ --- (1)

$= X_1^*[-k] + jX_2^*[-k]$

Similarly, $X[-k] = X_1[-k] + jX_2[-k]$ ----- (2)

$= X_1^*[k] + jX_2^*[k]$

Adding (1) and (2)

$$X[k] + X[-k] = \underbrace{(X_1[k] + X_1^*[k])}_{\text{real }\#} + j\underbrace{(X_2[k] + X_2^*[k])}_{\text{real }\#} \quad (3)$$

Subtracting (2) from (1)

$$X[k] - X[-k] = \underbrace{(X_1[k] - X_1^*[k])}_{\text{imaginary }\#} + j\underbrace{(X_2[k] - X_2^*[k])}_{\text{imaginary }\#} \quad (4)$$

From (3) & (4)

$\text{Real}\{X[k] + X[-k]\} + \text{Imag}\{X[k] - X[-k]\} = 2X_1[k]$

$\text{Real}\{X[k] - X[-k]\} + \text{Imag}\{X[k] + X[-k]\} = 2X_2[k]$

6. (35 points.) Adaptive filtering. Consider an autoregressive process that satisfies the difference equation

$$x[n] - x[n-1] - 0.1x[n-2] = v[n]$$

where $\{x[n]\}$ is a wide sense stationary (WSS) process, and $v[n]$ is a white noise process with variance $\sigma_v^2 = 0.64$. Let $r_x[l] = \mathbb{E}[x[n]x[n-l]]$ be the autocorrelation function of the process $\{x[n]\}$ where "$\mathbb{E}$" is the statistical expectation.

(a) (5 pts) Generate a system of equations for the above process:

$$x[n]x[n-l] + a_1 x[n-1]x[n-l] + \ldots a_M x[n-M]x[n-l] = v[n]x[n-l], \quad l = 1,\ldots,M \quad (*)$$

What are the values of M, $a_1,\ldots,a_M$?

$x[n]x[n-1] - x[n-1]x[n-1] - 0.1x[n-2]x[n-1] = v[n]x[n-1]$

$x[n]x[n-2] - x[n-1]x[n-2] - 0.1x[n-2]x[n-2] = v[n]x[n-2]$

$\Rightarrow M=2, \quad a_1 = -1, \quad a_2 = -0.1$

(b) (5 pts) Take the statistical expectation of both sides in equation (*) and write down the Yule-Walker equation with $l = 0$.

$x[n]x[n] + a_1 x[n-1]x[n] + a_2 x[n-2]x[n] = v[n]x[n]$

$\mathbb{E}[x[n]x[n]] + a_1\mathbb{E}[x[n-1]x[n]] + a_2\mathbb{E}[x[n-2]x[n]] = \mathbb{E}[v[n]x[n]] = \mathbb{E}[(v[n] + x[n-1] + 0.1x[n-2]) \cdot v[n]]$

$r_x[0] - r_x[1] - 0.1r_x[2] = \mathbb{E}[v^2[n]] = \sigma_v^2 = 0.64$

(c) (10 pts) Take the statistical expectation of both sides in equation (∗), to determine the Yule-Walker equations. Write down these equations in matrix form such that $R_x \cdot \mathbf{a} = \mathbf{r_x}$ where R_x is an $M \times M$ autocorrelation matrix, $\mathbf{a}$ is an $M \times 1$ vector that contain a_1 to a_M and $\mathbf{r}$ is an $M \times 1$ vector that contain the "-" of autocorrelations from $r_x[1]$ to $r_x[M]$. Determine $r_x[0]$ to $r_x[M]$.

From part a, when we take the expectation, we get

$$r_x[1] - r_x[0] - 0.1 r_x[1] = \mathbb{E}[v[n]x[n-1]] = 0$$

$$r_x[2] - r_x[1] - 0.1 r_x[0] = \mathbb{E}[v[n]x[n-2]] = 0$$

$$\begin{bmatrix} r_x(0) & r_x(1) \\ r_x(1) & r_x(0) \end{bmatrix} \begin{bmatrix} -1 \\ -0.1 \end{bmatrix} = \begin{bmatrix} -r_x(1) \\ -r_x(2) \end{bmatrix}$$

From part b, $r_x(0) - r_x(1) - 0.1 r_x(2) = 0.64$

Solving these, $r_x[0] = 0.9 r_x[1]$

$r_x[2] = r_x[1] + 0.1 r_x[0] = 1.09 r_x[1]$

& $0.9 r_x(1) - r_x(1) - 0.1 \times 1.09 r_x(1) = 0.64$

$r_x(1) = -3.06$

$r_x(0) = -2.756$

$r_x(2) = -3.34$

(d) (10 pts) Assume that we observe the signal $y[n]$ which is a noisy version of $x[n]$ and is given by

$$y[n] = x[n] + w[n]$$

where $w[n]$ is a white noise process with variance $\sigma_w^2 = 1$. We want to design a filter to recover an estimate of $x[n]$ from $y[n]$, i.e. the desired output $d[n] = x[n]$. Design a Wiener filter with length 3 to estimate $x[n]$. Note: Recall that the Wiener filter coefficients $\hat{h}$ satisfy $R_y\hat{h} = \mathbf{p}$ where R_y is the autocorrelation matrix of $y[n]$ and $\mathbf{p} = [\mathbb{E}[y[n]d[n]], \mathbb{E}[y[n-1]d[n]], \mathbb{E}[y[n-2]d[n]]]^\top$ is the cross correlation vector between $y[n]$'s and $d[n]$.

i. What is R_y in terms of R_x?

ii. Determine $\mathbb{E}[d[n]d[n]]$.

iii. Determine $\mathbf{p}$.

Let's use vector notation: $\underline{y}[n] = \begin{bmatrix} y[n] \\ y[n-1] \\ \vdots \end{bmatrix}$, $\underline{x}[n] = \begin{bmatrix} x[n] \\ x[n-1] \\ \vdots \end{bmatrix}$, $\underline{w}[n] = \begin{bmatrix} w[n] \\ w[n-1] \\ \vdots \end{bmatrix}$

0 since $\underline{w}[n]$ is zero mean & independent of $\underline{x}[n]$

i. $R_y = \mathbb{E}[\underline{y}[n]\underline{y}^T[n]] = \mathbb{E}[\underline{x}[n]\underline{x}^T[n]] + \mathbb{E}[\cancel{2\underline{x}[n]\underline{w}^T[n]}] + \mathbb{E}[\underline{w}[n]\underline{w}^T[n]]$

$= R_x + \sigma_w^2 I$

ii. $\mathbb{E}[d[n]d[n]] = \mathbb{E}[x[n]x[n]] = r_x(0)$

iii. $$\underline{p} = \mathbb{E}\begin{bmatrix} y[n]x[n] \\ y[n-1]x[n] \\ y[n-2]x[n] \end{bmatrix} = \mathbb{E}\begin{bmatrix} x^2[n] + x[n]w[n] \\ x[n]x[n-1] + x[n]w[n-1] \\ x[n]x[n-2] + x[n]w[n-2] \end{bmatrix} = \begin{bmatrix} r_x(0) \\ r_x(1) \\ r_x(2) \end{bmatrix}$$

(e) (5 pts) Determine the expression for the minimum mean square error (MMSE) for the length 3 filter in part (d).

$\hat{h} = R_y^{-1} p$ (Wiener filter coefficients)

output of this filter: $\sum_{k=0}^{2} \hat{h}[k]\, y[n-k]$

error = MMSE $= \mathbb{E}\left[\left(d[n] - \sum_{k=0}^{2} \hat{h}[k]\, y[n-k]\right)^2\right]$

$$= \underbrace{\mathbb{E}[d^2[n]]}_{r_x(0)} - \underbrace{2\sum_{k=0}^{2}\hat{h}[k]\,\mathbb{E}[d[n]y[n-k]]}_{p^T h} + \underbrace{\sum_{k=0}^{2}\sum_{l=0}^{2}\hat{h}[k]\hat{h}[l]\,\mathbb{E}[y[n-k]y[n-l]]}_{h^T R_y h}$$

7. (30 points.) IIR filter design.

Consider an IIR system with input $x[n]$ and output $y[n]$ described by the difference equation

$$y[n] = -\sum_{k=1}^{N} a_k y[n-k] + \sum_{m=0}^{M} b_m x[n-\cancel{k}].$$

m

(a) (5 pts) Determine the transfer function $H(z)$ of the system.

$$Y(z) = -\sum_{k=1}^{N} a_k Y(z) z^{-k} + \sum_{m=0}^{M} b_m X(z) z^{-m}$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{m=0}^{M} b_m z^{-m}}{1+\sum_{k=1}^{N} a_k z^{-k}}$$

(b) (15 pts) Now consider the digital filter with $H_d(z) = \frac{B(z)}{A(z)}$ where $B(z) = b_0 + b_1 z^{-1} + \ldots b_M z^{-M}$ and $A(z) = 1 + a_1 z^{-1} + \ldots + a_N z^{-N}$. Propose a procedure to determine the impulse response $h_d[n]$.

$$H_d(z) = \sum_{n=0}^{\infty} h_d[n] z^{-n} = h_d[0] + h_d[1] z^{-1} + h_d[2] z^{-2} + \ldots$$

$$= \frac{\sum_{m=0}^{M} b_m z^{-m}}{1+\sum_{k=1}^{N} a_k z^{-k}}$$

$$\Rightarrow \left(h_d[0] + h_d[1] z^{-1} + h_d[2] z^{-2} + \ldots\right)\left(1 + \sum_{k=1}^{N} a_k z^{-k}\right) = \sum_{m=0}^{M} b_m z^{-m}$$

$$h_d[0] + \sum_{k=1}^{N} a_k h_d[0] z^{-k} + h_d[1] z^{-1} + \sum_{k=1}^{N} a_k h_d[1] z^{-k-1} + \ldots = \sum_{m=0}^{M} b_m z^{-m}$$

$h_d[0] = b_0$, $h_d[0] a_1 + h_d[1] = b_1$, $h_d[0] a_2 + h_d[1] a_1 + h_d[2] = b_2$, and so on.

(c) (10 pts) Describe a procedure that computes the discrete time frequency response $H\left(\frac{2\pi}{N}k\right)$ for $k = 0, 1, \ldots, N-1$, in terms of $A(z)$ and $B(z)$ using the FFT algorithm.

$$H_d(z) = \frac{B(z)}{A(z)} \longrightarrow H_d[k] = \frac{\text{DFT}(b[n])}{\text{DFT}(a[n])} = \frac{B[k]}{A[k]}, \quad k = 0, \ldots, N-1$$

where $b[n] = b_n$ and $a[n] = a_n$

$\downarrow$ has $M+1$ unknowns $\quad \downarrow$ has N unknowns

Blank page for extra work

Blank page for extra work

Blank page for extra work