

EE 503 - Homework #3

Department of Electrical and Electronics Engineering,
Middle East Technical University,
Ankara, Turkey.

Due: Dec. 23, 2015

Computer Assignment: This assignment aims to explore the AR processes. The assignment has two parts. In the first part, the auto-correlation and power spectral density of narrowband and wideband AR processes are examined. In the second part, the coefficients of the filter generating the AR process is estimated with different methods and a whitening filter is implemented.

The following table presents the numerator coefficient (b_0) and the poles of filters generating AR(4) processes labeled as P_A , P_B and P_C in this assignment. The transfer functions of the generating filter is in the form $H(z) = b_0/\prod_{k=1}^4(1 - p_k z^{-1})$. The generating filter is driven with zero mean, unit variance white Gaussian noise.

	b_0	p_1	p_2	p_3	p_4
P_A	1	$0.95 \exp(j\pi 50/180)$	$0.95 \exp(-j\pi 50/180)$	$0.97 \exp(j\pi 110/180)$	$0.97 \exp(-j\pi 110/180)$
P_B	1	$0.80 \exp(j\pi 50/180)$	$0.80 \exp(-j\pi 50/180)$	$0.85 \exp(j\pi 110/180)$	$0.85 \exp(-j\pi 110/180)$
P_C	1	$0.70 \exp(j\pi 50/180)$	$0.70 \exp(-j\pi 50/180)$	$0.75 \exp(j\pi 110/180)$	$0.75 \exp(-j\pi 110/180)$

Part 1: A process is called a narrowband process if a significant portion of its power lies in a rather narrow band of the spectrum. Wideband process is defined similarly.

- Plot the power spectral density of the processes P_A , P_B and P_C using the relation $P(e^{j\omega}) = |H(e^{j\omega})|^2$, by evaluating $P(e^{j\omega})$ function at 256 points uniformly spaced in $[0, 2\pi)$. Which one of these processes is most “narrowband”? Also use the pole-zero diagram of $H(z)$ to strengthen your answer.
- Numerically calculate the power of the process, $r_x[0] = E\{x[n]^2\}$, by approximating $r_x[0] = \frac{1}{2\pi} \int_0^{2\pi} P(e^{j\omega}) d\omega$ with $r_x[0] \approx \frac{1}{2\pi} \sum_{k=0}^{255} P(e^{j2\pi/256k}) \frac{2\pi}{256}$.
- Calculate the impulse response of the filters generating the process with the “filter” command of Matlab. Which one of the impulse responses decay at the slowest rate? Truncate the impulse response of each filter to an interval $[0, X]$ where the samples outside of this interval is negligibly small valued, say smaller than 10^{-5} in magnitude. Considering the length of truncated impulse response, give a rough number for the transient time of each filter.
- Calculate the total energy of the truncated impulse response and compare your result with $r_x[0]$ calculated in part b). Explain why these two results should match.
- Generate 3 realizations for the processes P_A , P_B and P_C by filtering unit variance white Gaussian noise with the process generating filters. For a successful process generation,

discuss why we need to only consider the samples in the “steady-state” part of the response (not transient). Let $N_{win} = 30$ and take N_{win} consecutive samples from the “steady-state” part of the output and calculate its biased auto-correlation estimate via $\hat{r}_x[k] = \frac{1}{N_{win}} \sum_{n=0}^{N_{win}-1-k} x[n]x[n+k]$. Plot the 256 point FFT of $\hat{r}_x[k]$ for each realization and also the true $P(e^{j\omega})$ found in part a) on the same figure. Also discuss whether you can get same result for the power spectral density estimate by calculating the magnitude spectrum square of $x[n]$ where $n = \{0, 1, \dots, N_{win} - 1\}$ via FFT, i.e. without calculating the biased auto-correlation estimate.

- f) Repeat part e) for $N_{win} = 100$ and $N_{win} = 200$.
- g) Repeat part e) for $N_{win} = 100$ for 50 realizations. Do not plot any realizations, but plot the average of 50 realizations and the true $P(e^{j\omega})$ on the same figure. Comment on the results given by parts f) and g).
- h) Using the impulse response found in part c), numerically calculate the auto-correlation sequence by $r_x[k] = h[n] * h[-n]$. Check whether Yule-Walker equations are satisfied or not for the generated $r_x[k]$ sequence.
- i) Calculate the 256 point inverse FFT of true power spectral densities found in part a). By using “fftshift” command, generate $r_x[k]$. Explain why we need “fftshift” operation. Check whether Yule-Walker equations are satisfied or not for the generated $r_x[k]$ sequence. If Yule-Walker equations are not satisfied, present an explanation considering the nature of DFT. Compare your auto-correlations estimates found in parts h) and i).

Part 2: In this part, we construct the whitening filters for the processes P_A, P_B and P_C . The whitening filter is the inverse of the generating filter, $1/H(z)$; hence it is a 4 tap FIR filter for the processes examined in this assignment. Remember that $H(z)$ is the generating filter of an AR(4) process and therefore, it is in the form $H(z) = \frac{b_0}{1+a_1z^{-1}+a_2z^{-2}+a_3z^{-3}+a_4z^{-4}}$.

- a) **Auto-correlation method for AR spectrum estimation:** Generate $N_{win} = 50$ samples for each process. Calculate the biased auto-correlation estimates as in Part 1e). Replace true $r_x[k]$ with the estimated ones in the Yule-Walker equations and solve for b_0 and $[a_1 a_2 a_3 a_4]$. (Before attempting this part, please read Section 8.5.1 (p.441) of Hayes.)
- b) This part aims to redo the auto-correlation method in part a) through a different approach. Generate matrix \mathbf{X}_p of equation (4.124) of Hayes (p.181) by taking $p = 4$ and $N = N_{win} = 50$ and solve for $\mathbf{X}_p \hat{\mathbf{a}}_p = -\mathbf{x}_1$ in the least squares sense (\mathbf{x}_1 is given in equation (4.125)). Compare your results with part a). What is the main idea of this approach? Find b_0 as in the auto-correlation method.
- c) **Covariance method for AR spectrum estimation:** Find the least square solution for the middle portion of equation $\mathbf{X}_p \hat{\mathbf{a}}_p = -\mathbf{x}_1$, where middle portion is the part in between

dashed lines of the matrix in equation (4.124) and it is explicitly given in equation (4.130). Find b_0 as in the auto-correlation method.

- d) In the earlier parts, we have implemented auto-correlation and covariance methods for the estimation of the process generating filter. We may denote the estimated generator filter as $\hat{H}(z) = \frac{\hat{b}_0}{1 + \hat{a}_1 z^{-1} + \hat{a}_2 z^{-2} + \hat{a}_3 z^{-3} + \hat{a}_4 z^{-4}}$. An approximation to the true whitening filter can then be $1/\hat{H}(z)$.

We consider the application of the estimated whitening filter on the process $x[n]$ as shown in Figure 1. In this figure $w[n]$ is the white noise with unit variance, $H(z)$ is the process generating filter and $1/\hat{H}(z)$ is the estimated whitening filter. If the estimation for the whitening filter is perfect, then we have $y[n] = w[n]$, i.e. white noise output.

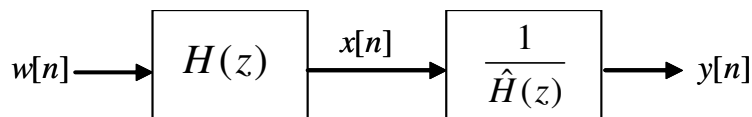


Figure 1: The (estimated) whitening filter applied to $x[n]$

Consider the overall transfer function for the cascade, $H_c(z) = H(z)/\hat{H}(z)$, and find the power spectral density of $y[n]$ at 256 points uniformly spaced in $[0, 2\pi)$ as in part 1a) and also calculate the variance of $y[n]$ as in part 1b). To find the power spectral density of $y[n]$ and variance of $y[n]$, take $N_{win} = 50$ samples as before and apply both auto-correlation method and covariance method using N_{win} samples. Is there a significant difference between the methods for the processes P_A, P_B and P_C ?

Using $N_{win} = 50$ samples, present the power spectral density plot of $y[n]$ for 3 different realizations for each process (making a total of 9 plots). Also present the variance of $y[n]$ for 3 different realizations.

- e) Repeat part d) for $N_{win} = \{100, 200\}$.
- f) Repeat part d) for $N_{win} = \{50, 100, 200\}$ for 500 realizations. Do not present any plots; but present the average of $y[n]$ variance over 500 realizations for the stated N_{win} values and for the stated methods (auto-correlation and covariance methods). Keeping in mind that the perfect whitening filter results in an unit variance $y[n]$ process, compare the AR spectrum estimation methods for different data record size (N_{win}) and for different process types (narrowband, wideband).

Note: Also submit your Matlab code along with your plots and comments. You may consider using the “publish” feature of Matlab.