

Signal Processing and Speech Communication Lab.
Graz University of Technology

AS

Adaptive Systems—Homework Assignment 2

Name(s)

Matr.No(s).

The analytical part of your homework (your calculation sheets) **as well as** the MATLAB simulation protocol have to be delivered as **hard copy** to our mailbox at Inffeldgasse 16c, ground floor, no later than **2017/1/9**. Use a printed version of **this entire document** as the title pages and fill in your **name(s) and matriculation number(s)**. Submitting your homework as a \LaTeX document can earn you **up to 2 points!**

Your MATLAB programs (*.m files) and the simulation protocol (in pdf format!) have to be submitted via **e-mail** to the address `hw2.spsc@tugraz.at` no later than **2017/1/9**. The subject of the e-mail consists of the assignment number and your matriculation number(s) “**Assignment2, MatrNo1, MatrNo2**”. You have to zip (or tar) all your homework files to one single file with the name `Assignment2_MatrNo1_MatrNo2.zip`, e.g., `Assignment2_9833280_9933281.zip`, which has to be attached to the e-mail.

A justification of all your answers is mandatory to achieve all points!

Name(s)

Matr.No(s).

Analytical Problem 2.1 (12 Points)—Linear Prediction and Spectral Estimation

Linear Prediction is a widely used mathematical operation where the goal is to estimate future values of a discrete-time signal based on previous samples. It is widely used in speech processing where it is also known as linear predictive coding. A second importance is in the field of spectral estimation, where one uses autoregressive, moving-average, or mixed models to estimate the power spectral density of a process. Here we can make use of the Yule-Walker equations (or modified Yule-Walker) to compute the autoregressive coefficients and thus, estimating the power spectral density. In the following tasks we want to investigate linear prediction in more detail, which is one of the essential parts in digital signal processing.

(a) Assume that we observe a real process given by

$$y[n] = x[n] + \nu[n]$$

where $x[n]$ is a zero-mean real WSS random process with power spectral density $S_{xx}(e^{j\omega})$ and $\nu[n]$ is a zero-mean real white noise process with variance σ_ν^2 which is assumed to be uncorrelated with $x[n]$. From Wiener theory we know the estimate to be given by

$$\hat{x}[n] = \sum_{k=-\infty}^{\infty} h[k]y[n-k]$$

where the impulse response is chosen to minimize the MSE

$$\text{MSE} = \mathbb{E}((x[n] - \hat{x}[n])^2)$$

(i) Show that the impulse response must satisfy

$$r_{xy}[m] = \sum_{k=-\infty}^{\infty} h[k]r_{yy}[m-k]$$

(ii) Use the result from (i) and derive the frequency response $H(e^{j\omega})$ of the Wiener filter. Discuss on the use of a Wiener filter for a noise reduction scenario in AR spectral estimation.

(b) Explain and show mathematically the whitening property of the prediction error filter. Therefore, assume an AR(p) model with a white noise excitation. Sketch the system block diagram of the prediction error filter. What condition must the system fulfill? Think of poles and zeros.

(c) A student at our lab was researching on linear prediction and therefore tried to simulate different applications. Once he tried to calculate the coefficients a mistake happened and he forgot to save his session. He then obtained several pole-zero plots of the **prediction error filter** and wasn't able to reproduce his results on his own. Hopefully you can help him with this problem. In the figure below there are four pole-zero plots which the student gave us. Additionally he can remember two of the 4 coefficients

$$a_{opt} = \begin{bmatrix} 1 \\ -0.98538 \\ a_2 \\ a_3 \end{bmatrix}$$

and two poles of the AR-model located at $p_1 = 0.7198 + j0.323$ and $p_2 = p_1^*$. Which of the plots below can be the correct one?

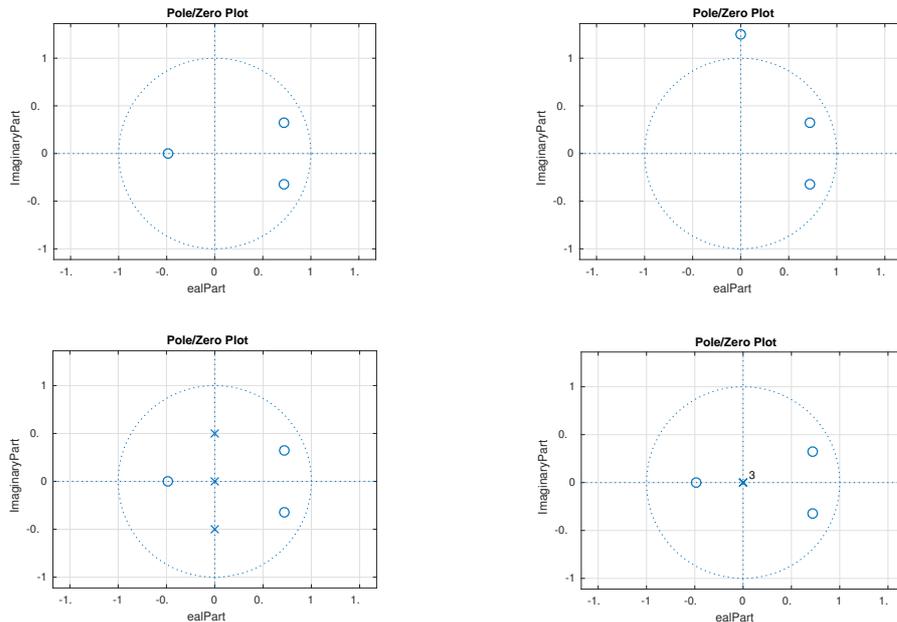
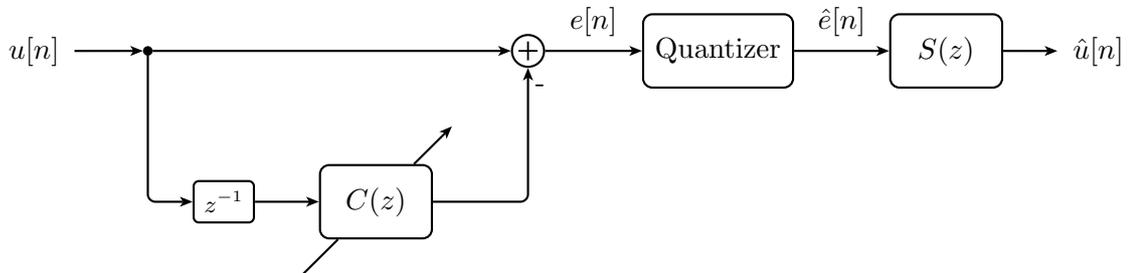


Figure 1: Different pole-zero plots of the error prediction filter.

(d) Consider the following system depicted in the figure below.



(i) Let $u[n]$ be samples of an AR process with the causal generator difference equation

$$u[n] = \nu[n] + 1.5u[n-1] - 0.5625u[n-2].$$

$\nu[n]$ are samples of white, zero-mean, Gaussian noise. The variance of the AR process $u[n]$ is known as $\sigma_u^2 = 1$. Derive the first three samples of the autocorrelation sequence $r_{uu}[k]$, $k = 0, 1, 2$. Also, compute the variance of the white-noise input, σ_ν^2 , and the noise gain of the given recursive process generator filter

$$G_G = \sigma_u^2 / \sigma_\nu^2$$

(ii) For the above AR process, compute the MSE-optimal linear predictor of second order

$$C(z) = c_0 + c_1 z^{-1}$$

Also, compute the so-called *prediction gain*

$$G_P = \sigma_u^2 / \sigma_e^2$$

(iii) For the obtained MSE-optimal predictor, specify the transfer function of the synthesis filter $S(z)$, and calculate its noise gain G_S .

(iv) For the lossy encoder, we use a uniform scalar quantizer followed by an ideal lossless coder. Such a lossy encoder can be modeled by a simple additive-noise model with the following properties

$$\sigma_q^2 = \frac{\Delta^2}{12}$$

$$R = 0.5 \log_2(2\pi e \sigma_e^2) - \log_2 \Delta$$

$$\mathbb{E}[q[n]q[n-k]] = 0, \forall k \neq 0$$

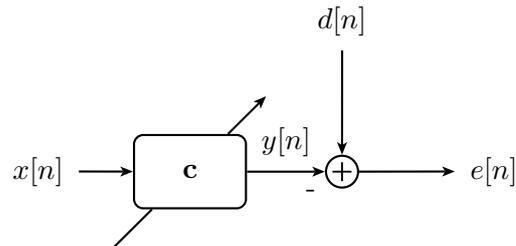
R is the average bit rate in bits/sample and Δ is the quantization cell size. Assume four different bit rates $R \in [2 \ 5 \ 10 \ 20]$ bit/sample. For the MSE-optimal predictor, calculate the cell size Δ , the quantization noise variance σ_q^2 and the reconstruction error variance σ_e^2 at the decoder output $\hat{u}[n] = u[n] + r[n]$ for every bit rate.

Name(s)

Matr.No(s).

Analytical Problem 2.2 (10 Points)—Leaky Gradient Method

Consider the following linear filtering problem as depicted in the figure below.



As an adaptive algorithm we want to use the so-called *leaky gradient search* which is given by the update equation

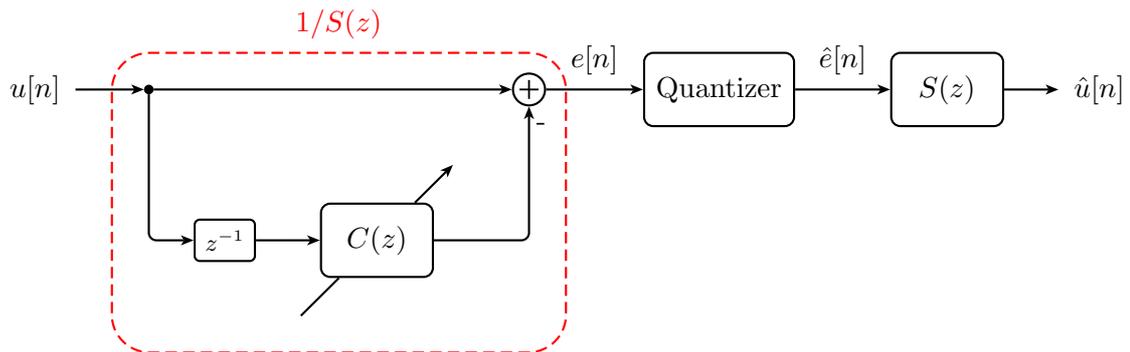
$$\mathbf{c}[n] = (1 - \mu\alpha)\mathbf{c}[n - 1] + \mu(\mathbf{p} - \mathbf{R}_{xx}\mathbf{c}[n - 1])$$

The parameter α is known as the leakage parameter with the property $0 < \alpha \ll 1$ and μ is the step size.

- (a) Let us assume convergence. Where does this algorithm converge to?
- (b) Transform the given adaption rule in a way such that it adapts the misalignment vector $\mathbf{v}[n]$.
- (c) Decouple the adaption rule into a set of scalar adaption expressions by using a unitary transform given by $\mathbf{q}[n] = \mathbf{Q}^H \mathbf{v}[n]$
- (d) Write the decoupled equation as individual exponential sequences and derive a condition on μ to ensure convergence. Therefore you can assume that α is known. Specify the range $\mu_{min} < \mu < \mu_{max}$.
- (e) Assume a fixed step size parameter μ . Compute the worst-case convergence time constant τ_{max} .

Name(s)

Matr.No(s).

MATLAB Problem 2.3 (12 Points)—Open-Loop Prediction and Information Theory


You should apply a LMS implementation to perform open-loop linear prediction. In general, open-loop prediction does not reduce the reconstruction error; here we show that it still increases the information transfer. $u[n]$ is a moving-average process of order two, described by the difference equation

$$u[n] = w[n] + 0.75w[n - 1] + 0.75w[n - 2]$$

where $w[n]$ is a white noise process with uniform distribution on $[-1, 1]$.

(a) Implement the LMS algorithm based on the following function header

```
function [y,e,c] = lms(x,d,N,mu,c0)
% INPUTS:
% x ..... input signal vector
% d ..... desired output signal (of same length as x)
% N ..... number of filter coefficients
% mu ,... step size parameter
% c0 ..... initial coefficient vector (optional; default all zeros)
% OUTPUTS:
% y ..... output signal vector (of same length as x)
% e ..... error signal vector (of same length as x)
% c ..... coefficient matrix (N rows, number of columns = length of x)
```

- (b) Can a finite-order filter \mathbf{c} perfectly predict the process $u[n]$? Justify your answer!
- (c) Compute (analytically) the autocorrelation sequence $r_{uu}[k]$ and (numerically) the optimal filter coefficients \mathbf{c}_{MSE} for $N = 2$, $N = 5$, and $N = 10$.
- (d) Use your implementation of the LMS algorithm to compute the coefficient vector \mathbf{c} and the error signal $e[n]$. Plot the first coefficient c_0 as a function of time n and compare it to its analytic solution. What can you observe? Again, perform this task for $N = 2$, $N = 5$, and $N = 10$. Use at least 10^5 samples of $u[n]$ for your simulation.

A measure of how much *information* about the original process $u[n]$ is available after quantization is given by the entropy rate of the quantized process $\hat{e}[n]$. This entropy rate is given by the conditional entropy of the current sample $\hat{e}[n]$ conditioned on the entire past:

$$\bar{H}(\hat{e}) = H(\hat{e}[n]|\hat{e}[n-1], \hat{e}[n-2], \dots).$$

(It tells you how many bits you do not know about the current sample given you know all previous ones.) Entropy is non-increasing by conditioning, i.e. the more samples of the past you know, the more you can guess about the current sample:

$$H(\hat{e}[n]) \geq H(\hat{e}[n]|\hat{e}[n-1]) \geq H(\hat{e}[n]|\hat{e}[n-1], \hat{e}[n-2]) \geq \dots \geq \bar{H}(\hat{e}).$$

(e) Evaluate the effect of linear prediction with $N = 0$, $N = 2$, $N = 5$, and $N = 10$ on the information that is available after quantization with $B = 4$ bits. Use the scripts from the website:

- `BGsQuantizer.m`: This function quantizes the input data with a B -bit quantizer (input parameter), where the thresholds depend on the 10%/90% quantile of the data. Take a look at the implementation!
- `CondEntropy.m`: This function estimates the entropy $H(\hat{e}[n])$ and the conditional entropies given one and two previous samples.

Plot the three output values of `CondEntropy.m` as a function of N .

(f) What can you observe? Does open-loop linear increase the information at the output of the quantizer?

Name(s)

Matr.No(s).

Analytical Problem 2.4 (6 Points)—BONUS: The ill conditioning of a matrix

The ill conditioning of a matrix R_x increases with its condition number $\chi(R_x) = \frac{\lambda_{\max}}{\lambda_{\min}}$. When R_x is a correlation matrix of a stationary process, then $\chi(R_x)$ is bounded from above by the dynamic range of the PSD $R_x(e^{j\omega})$ of the process $x(n)$. The larger the spread in eigenvalues, the wider (or less flat) the variation of the PSD function. This is also related to the dynamic range or to the data spread in $x(n)$ and is a useful measure in practice.

- (i) Consider a zero-mean stationary random process with power spectral density

$$R_x(e^{j\omega}) = \sum_{l=-\infty}^{\infty} r(l)e^{-j\omega l}$$

then show *mathematically* that the eigenvalues are bounded as follow

$$\min_{\omega} R_x(e^{j\omega}) \leq \lambda_i \leq \max_{\omega} R_x(e^{j\omega})$$

for all $i = 1, 2, \dots, M$.

- (ii) We are interested to find a criterion to check when an autocorrelation matrix becomes ill condition. This is possible by finding the bound of the condition number via inspecting the Eigenvalue spread and the spectral dynamic range. Using the property in (i), show *mathematically* that

$$\chi(R_x) = \frac{\lambda_{\max}}{\lambda_{\min}} \leq \frac{\max_{\omega} R_x(e^{j\omega})}{\min_{\omega} R_x(e^{j\omega})}$$