Signal Processing and Speech Communication Lab.
Graz University of Technology

AS

Adaptive Systems—Homework Assignment 1

v1.0

Name(s) Matr.No(s).

Your solutions to the problems (your calculations, the answers to each task as well as the OCTAVE/MATLAB plots) have to be uploaded to the TeachCenter as a single `*.pdf` file, no later than **2019/12/6**. Use **this page** as the title page and fill in your **name(s) and matriculation number(s)**. Submitting your homework as a LaTeX document can earn you **up to 3 bonus points!**
All scripts needed for your OCTAVE/MATLAB solutions (all `*.m` files) have to be uploaded to the TeachCenter as a single `*.zip` archive, no later than **2019/12/6**.
All filenames consist of the assignment number and your matriculation number(s) such as `Assignment1_MatrNo1_MatrNo2.*`, for example,

Problem solutions:	`Assignment1_01312345_01312346.pdf`
OCTAVE/MATLAB files:	`Assignment1_01312345_01312346.zip`

Please make sure that your approaches, procedures, and results are clearly presented. Justify your answers! A single upload of the files per group is sufficient.

Analytical Problem 1.1 (8 Points)—Theory

Linear Algebra
With the step from the continuous to the discrete domain, signal processing in general profits from many results that can be taken from linear algebra and matrix theory.

(a) (2 points) Perform the following tasks dealing with vectors and matrices in general.

(i) Assume that you have a matrix $\mathbf{A}$ whose columns $\mathbf{a}_i$ contain some signals. Compute a linear combination of the columns, i.e, the signals, using a single vector $\mathbf{x}$. Assume that $\mathbf{A}$ is a $(N \times 4)$ matrix. Carefully state the size of $\mathbf{x}$.

(ii) Find a single matrix $\mathbf{B}$ that allows performing the following tasks without changing the size of the vector $\mathbf{a}$ and the matrix $\mathbf{D}$. Be careful regarding the dimensions of $\mathbf{B}$.

- multiplication of the elements of a column vector $\mathbf{a} = [a_1, a_2, a_3]^\mathsf{T}$ by different constants b_1, b_2 and b_3.
- multiplication of the columns of an arbitrary $(N \times 3)$ matrix $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3]$ by different constants b_1, b_2 and b_3.

Hint: You can multiply matrices form the left or from the right depending on the dimensions.

(iii) Show that $\det(a\mathbf{A}) = a^N \det(\mathbf{A})$ assuming that $\mathbf{A}$ is an $N \times N$ matrix and a is some real-valued scalar constant. It is enough to show this for an exemplary matrix $\mathbf{A}$ of size (2×2).

Mean–Square Error Filters
The Wiener filter yields the optimum solution in the *mean–square error* sense, while requiring stationarity of the involved signals as well as knowledge of their correlation functions. The derivation is based on the gradient of the *mean–square error* (MSE) cost function $J_{\mathrm{MSE}}(\mathbf{c}) = \mathsf{E}\left[|e[n]|^2\right]$.

(b) (1.5 points) An alternative approach to derive the Wiener filter is by the so–called *principle of orthogonality*, which states that the error $e[n]$ is orthogonal to each filter input sample $x[n-k]$:

$$\mathsf{E}\left[e[n]\mathbf{x}[n]\right] = \mathbf{0}$$

(iv) Derive the orthogonality principle from the MSE cost function $J_{\mathrm{MSE}} = \mathsf{E}\left[|e[n]|^2\right]$

(v) Derive the Wiener–Hopf solution from (iv)

(vi) What property does the autocorrelation matrix need to fulfill to allow using the Wiener-Hopf solution?

(c) (4.5 points) Assume that the input signal $x[n]$ of your adaptive filter is of the form

$$x[n] = \alpha e^{j(\theta_0 n + \varphi)} + w[n]$$

where $\alpha \in \mathbb{R}$ is a constant amplitude, $\theta_0 \in \mathbb{R}$ a constant frequency, $\varphi \sim \mathcal{U}(-\pi, \pi]$ a uniformly distributed random phase and $w[n]$ zero-mean Gaussian noise with non-zero variance σ_w^2. Assume that $w[n]$ and φ are independent. Compute the autocorrelation sequence $r_{xx}[n,k] = \mathsf{E}\left[x[n+k]x^*[n]\right]$ for the noisy signal and perform the following tasks.

(*vii*) Show that the signal $x[n]$ is wide sense stationary (WSS).

(*viii*) Compute the autocorrelation matrix $\mathbf{R}_{xx}$. Is it always invertible? What happens for $\sigma_w^2 \to 0$?

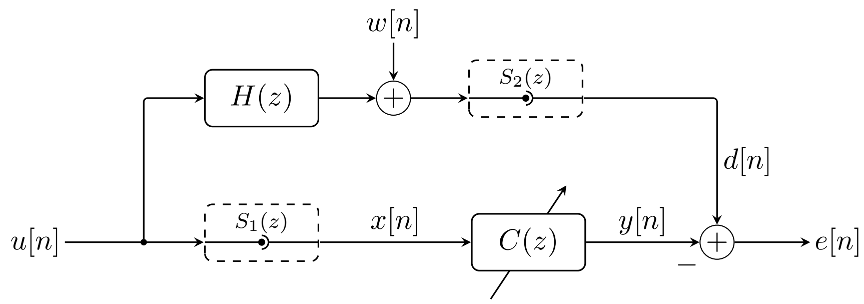
(*ix*) When dealing with stochastic signals, we can only analyze the frequency content of the signal by computing the Fourier transform (the DTFT) of the corresponding autocorrelation function, then called the power spectral density (PSD)[1] $S_{xx}(\theta)$. Compute and sketch the PSD, defined as

$$S_{xx}(\theta) = \sum_{k=-\infty}^{\infty} r_{xx}[k] e^{-j\theta k}.$$

Again show what happens for $\sigma_w^2 \to 0$. Can you use the PSD to find out whether the autocorrelation matrix is invertible?

Analytical Problem 1.2 (15 Points)—System Identification with Realistic Sensors

Consider the following system identification problem:



Assume that $u[n]$ and $w[n]$ are independent, jointly stationary, white noise processes with variances $\sigma_u^2 = 1$ and $\sigma_w^2 = \frac{1}{4}$, respectively. We want to identify the unknown LTI system $H(z) = 1 + \frac{1}{4}z^{-1}$ using a second-order adaptive filter $C(z)$ (i.e., $N = 3$ and $\mathbf{c} = [c_0, c_1, c_2]^T$). Unfortunately, as often the case in reality, we cannot assume that our adaptive filter has direct access to $x[n]$ or $d[n]$, but only to filtered versions of these. This can be due to non-ideal sensor responses $S_1(z)$ and $S_2(z)$ as indicated in the flowgraph, which are unknown to the user. This problem should help understanding the problems sensor imperfections can cause.
For all the following cases, determine

1. the values of the auto-correlation sequence $r_{xx}[k]$ and the autocorrelation matrix $\mathbf{R}_{xx}$,
2. the values of the cross-correlation vector $\mathbf{p} = \mathsf{E}\big[d[n]\mathbf{x}[n]\big]$,
3. the optimal coefficient vector $\mathbf{c}_{\text{opt}}$ in the sense of a minimum mean-squared error, i.e. $\mathbf{c}_{\text{opt}} = \operatorname{argmin}_{\mathbf{c}} J_{\text{MSE}}(\mathbf{c})$ with $J_{\text{MSE}}(\mathbf{c}) = \mathsf{E}\big[|e[n]|^2\big]$, and
4. the minimum mean-squared error $J_{\min} = J_{\text{MSE}}(\mathbf{c}_{\text{opt}})$.

---

[1]You can find additional information in Ch. 1.3 in [SM+05]

Always answer if you can identify the system correctly, and if so, why this is the case. If not, explain why system identification fails. Until stated otherwise assume that you do not know $S_1(z)$ and $S_2(z)$.
Hint: In some cases it may be helpful to redraw the signal model by taking the linearity of the systems into account.

(a) (3.5 points) Let $S_1(z) = S_2(z) = 1$.

(b) (3.5 points) Let $S_1(z) = S_2(z) = 1 + 0.5z^{-1}$.

(c) (3.5 points) Let $S_1(z) = 1$ and $S_2(z) = z^{-1}$.

(d) (3.5 points) Let $S_1(z) = 1$ and $S_2(z) = 1 + z^{-2}$.

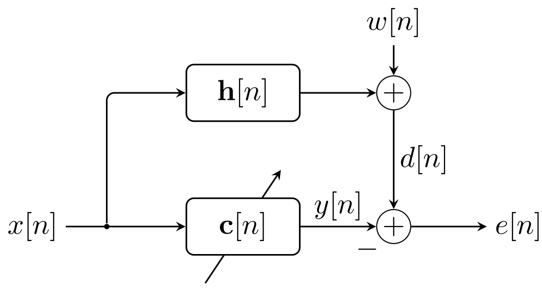
(e) (1 point) What happens when the sensor responses are known? Can the system be identified correctly? Can the cascade in **(d)** be identified correctly? You do not need to compute anything, just give a short explanation.

OCTAVE/MATLAB Problem 1.3 (12 Points)—Least–Squares Tracking of a Time–Varying System

Consider the system identification problem for a time–varying impulse response. The desired signal follows

$$d[n] = \mathbf{h}^{\mathsf{T}}[n]\mathbf{x}[n] + w[n]$$

where $\mathbf{h}[n]$ indicates the impulse response, now dependent on the current time index n.



(a) (5 points) Write an Octave/Matlab function that computes the *least–squares* optimum filter coefficients $\mathbf{c}_{\mathrm{LS}}[n] = \operatorname{argmin}_{\mathbf{c}} J_{\mathrm{LS}}(\mathbf{c}, n)$ according to the cost function

$$J_{\mathrm{LS}}(\mathbf{c}, n) = \sum_{k=0}^{n} |e[k]|^2.$$

You should hand in (at least) two Octave/Matlab file: One file that creates the signals, calls your function and preforms all required plots and one file containing your function itself. Implement your function according to the following specifications:

```
function c = ls_filter( x, d, N)
% x ... input signal vector
% d ... desired output signal vector (of same length as x)
% N ... number of filter coefficients
```

(b) (1 point) For the 'unknown' system, implement a filter with the following time–varying 3–sample impulse response:

$$\mathbf{h}[n] = \begin{bmatrix} -1 \\ 2 - 0.97^n \\ 0.3 \cdot \cos(\theta n) \end{bmatrix}$$

where $\theta = \frac{3\pi}{1000}$. Visualize the time–varying impulse response $\mathbf{h}[n]$ for $n = 0 \ldots 999$.

(c) (4 points) Assume now that the system is noise–free, i.e. $w[n] = 0$. Generate 1000 samples (for $n = 0 \ldots 999$) of an input signal $x[n]$ drawn from a stationary white noise process with zero mean and variance $\sigma_x^2 = 1$. Compute the output $d[n]$ of the system, under the condition that initially all delay elements contain zeros, i.e., $x[n] = 0$ for $n < 0$.
The adaptive filter should have 3 coefficients ($N = 3$). By calling your function `ls_filter` with length-M segments of both $x[n]$ and $d[n]$, the coefficients of the adaptive filter $\mathbf{c}[n]$ for $n = 0 \ldots 999$ can be computed. Note that we thus obtain $\mathbf{c}[n] = \text{argmin}_{\mathbf{c}}\, J(\mathbf{c}, n)$ where the cost function becomes $J(\mathbf{c}, n) = \sum_{k=n-M+1}^{n} |e[k]|^2$. For segment lengths of $M = \{20, 50\}$, create plots comparing the time evolution elements of the coefficient vector $\mathbf{c}[n]$ to the actual impulse response $\mathbf{h}[n]$. Compare and discuss your results and explain the effect of different M.
Hint: You can display the results in any (understandable) way you like, for example using `plot()`, `subplot()` or `waterfall()`. Make sure to label the axis and indicate the displayed data with appropriate legends.

(d) (2 points) Repeat task **(c)** for $w[n]$ being a zero–mean white noise process with variance $\sigma_w^2 = 0.02$ and compare the obtained results.

OCTAVE/MATLAB Problem 1.4 (5 Points)—Bonus: Weighted Least–Squares

This task again deals with least–squares linear filtering, now trying to achieve better performance for time varying systems. Consider that observations of $x[n]$ and $d[n]$ for $n = 0 \ldots M$ are given.

(a) (2 points) Derive the optimum filter coefficients $\mathbf{c}$ in the sense of a weighted least–squares, i.e., find $\mathbf{c}_{\text{wLS}}[n] = \text{argmin}_{\mathbf{c}}\, J_{\text{wLS}}(\mathbf{c}, n)$ for the weighted least–squares cost function

$$J_{\text{wLS}}(\mathbf{c}, n) = \sum_{k=n-M+1}^{n} g[n-k] \cdot |e[k]|^2.$$

(b) (1 point) Use your Octave/Matlab function `ls_filter( x, d, N)` from Problem 1.3 and extend it to implement the cost function $J_{\text{wLS}}(\mathbf{c}, n)$ for the weighted LS algorithm given above, naming the new function `ls_filter_weighted( x, d, N)`. If you want you can also add the weighting as an additional input argument.

(c) (2 points) Evaluate and compare the performance using the weighting function

$$g[m] = \lambda^{-m} \quad \text{for } m \geq 0$$

and using the time–varying system $\mathbf{h}[n]$ from Problem 1.3, but with $\theta = \frac{3\pi}{100}$ and $M = 50$. Try to find a suitable value for λ that accomplishes the tracking.

References

[SM+05] Petre Stoica, Randolph L. Moses, et al. Spectral Analysis of Signals. 2005.