Python Exercise I for SC3011TN: Academic Year 2020/2021

Python Exercise II: Audio denoising through FIR Wiener filtering

Carlas Smith, Michel Verhaegen, Dylan Kalisvaart & Jacques Noom

Version 1.0

February 17, 2021

TU Delft

Delft Center for Systems and Control

Introduction and theoretical reference

In this exercise, we consider a radio frequency (RF) communication system consisting of an emitter and a receiver. Interference originating from the RF environment and from thermal noise of the system cause signal degradation at the receiver. Since the nature of these aberrations is stochastic, Wiener filtering can be used to subtract the effects of the distortion.

Consider the signal configuration displayed in Figure 1 (see also Figure 7.6 on page 145 of the reader).

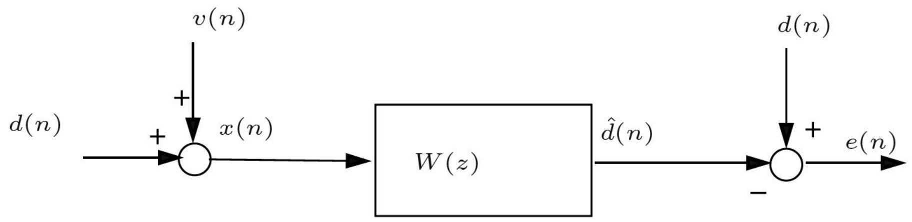

Figure 1: The problem configuration of the denoising problem.

In the configuration of Figure 1, all signals are discrete-time and real. The corrupted signal $x(n) = d(n) + v(n)$ contains both the signal of interest $d(n)$ and noise $v(n)$, which is uncorrelated to $d(n)$. In this assignment you will design a FIR filter $W(z)$, used to estimate the ground truth signal from $x(n)$ and from the statistical properties of the noise. That is, the filter $W(z)$ can be written as:

$$W(z) = w(0) + w(1)z^{-1} + \cdots + w(m-1)z^{-m+1} \tag{1}$$

such that using the measurable signals $x(n)$ and the additional information about the noise, the estimation of the ground truth signal can be obtained.

A thorough discussion of the denoising problem can be found in Subsection 7.3.3 of the reader [1].

Assignment description

Here, we will describe the assignment. As the COVID-measures prevent us from letting you record your own signals, we have done the experiment for you and provide you with the raw recordings. Therefore, we also provide you with the procedure we followed to record the signals.

To help you with reporting, you can use the provided Jupyter notebook as your report. We strongly encourage using the notebook, as hints are already provided for you. After you have answered the questions and are finished with the assignment, you can export the Jupyter notebook as a PDF (File > Download as > LaTeX or PDF).

Preparation

Make sure that the files `Audio.py`, `Communication.py` and `denoising-scheme.jpg`, as well as the recorded signals `gong_5cm.mat`, `gong_100cm.mat`, `gong_250cm.mat` and `noise.mat` are included in the same directory as the Jupyter notebook or your own code.

Install the following modules to your python using `pip`:

1. pyserial
2. sounddevice
3. scipy

For that you can use usual command prompt (in Windows use Start button $->$ ”cmd”) or if you use Anaconda, ”Anaconda prompt”. The full command for installing a python package is

```
pip install package_name
```

Obtaining the data

In this section, the transmission experiment from the introduction is described. As the COVID-measures prevent us from letting you record your own signals, we have done the experiment for you and we provide you with the raw recordings. Here, we describe the experimental setup that we used to record the signals. The experimental setup is shown in Figure 2.

For this experiment, two Arduino MK1000’s are used. These Arduino microcontrollers are able to send and receive signals through radio frequency (RF) wireless transmission. As such, the first microcontroller is configured as a sender, used to send a wav-file `gong.wav`. The second microcontroller is configured as a receiver and it is linked to the sender. By simultaneously activating both microcontrollers, the receiver will record the signal sent by the sender.

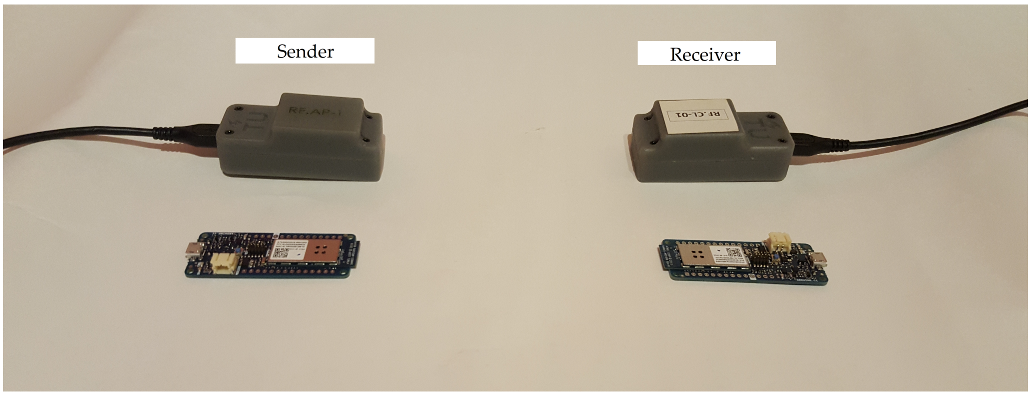

Figure 2: Experimental setup to send and receive sounds. The Arduino MK1000 microcontrollers are contained in the grey boxes.

However, the signal transmission is affected by RF noise. Examples of sources of this noise are disturbances in the RF band, distance-dependent noise caused by the distance between the microcontrollers and thermal distortion of the signal. These noise sources act additively on the sent signal `gong.wav`. For distances of 5 cm, 100 cm and 250 cm between the microcontrollers, we recorded `gong_5cm.mat`, `gong_100cm.mat` and `gong_250cm.mat` respectively, which you can listen to later in this assignment.

To filter the noise from the signal using denoising, we also need a recording of the noise for every distance of 5 cm, 100 cm and 250 cm between the microcontrollers. The total noise has a complicated spectrum, so to save you some mathematical work we solely focus on ambient, distance-independent noise. This noise is assumed to be zero-mean and white, and a measurement of this noise is contained in `noise.mat`.

NB. The recorded values in `gong_5cm.mat`,`gong_100cm.mat`,`gong_250cm.mat` and `noise.mat` are expressed in decibels relative to full scale (dB-FS), which is a unit used to expressed signal amplitudes in digital systems.

With these signals, you have enough information to calculate the weights of a FIR Wiener filter. This assignment will guide you through this process. A snippet of code is provided to you in the Jupyter notebook, which will help you with importing the signals.

Exercises

1. Formulate the denoising problem as a Minimum Variance FIR Wiener Problem, using the symbols $x(n)$, $v(n)$, $d(n)$, $\hat{d}(n)$, $e(n)$ as in Figure 1. In particular:

   - Explain for each variable what signal they represent
   - Write the cost function $J(W(z))$ and the optimization problem that should be solved to find the Minimum Variance FIR Wiener Filter.

   Note that no numerical values are required, but you need to specify the meaning of all the symbols and their relation to the sent and received data.

2 points

2. Plot the received audio signals and the received noise signal. You can plot all the audio signals in the same figure, but use a separate plot for the noise signal. Do not forget to label the axes and to provide a legend.

   NB. The recorded values in `gong_5cm.mat`, `gong_100cm.mat`, `gong_250cm.mat` and `noise.mat` are expressed in decibels relative to full scale (dB-FS).

1 point

3. Determine the Wiener-Hopf equation for the denoising case with filter order m. Clearly specify the size and content of all quantities involved.

   What assumptions are we making for the (true and noised) audio signal and the noise in order to obtain the weights of the filter w?

   Derive how can we calculate the right hand side of the Wiener-Hopf equation, assuming that we know $r_x(k),\ \forall k$ and the noise variance σ_v^2.

2 points

4. Estimate the mean and variance (σ_v^2) of the measured noise. Print these estimates.

1 point

5. (a) Assuming the audio signals are ergodic, calculate the auto-correlation function $r_x(k)$ of every received audio signal for lags $0, \ldots, 9$. Print these values.

   2 points

   (b) Calculate the cross-correlation function $r_{dx}(k)$ of every received audio signal for lags $0, \ldots, 9$. Print these values.

   1 point

   (c) Using Eq. (2), compute and plot the auto-correlation of the noise $v(n)$ for lags $0, \ldots, 9$. Is the noise ZMWN? Explain your answer.

   1 point

   Note: if the signal $x(n)$ is auto-correlation ergodic, the auto-correlation function $r_x(k)$ can be estimated as (if we have only values $x_1, x_2, \ldots, x_N$):

$$r_x(k) = \frac{1}{N-k} \sum_{i=k+1}^{N} x(i)x^*(i-k) \tag{2}$$

6. Using a 10-th order FIR Wiener filter, calculate the optimal filter coefficients for every received audio signal. Print these values.

   Hint: the `scipy.linalg.toeplitz` function may be useful.

2 points

7. Using the computed filter coefficients, compute an estimate of the denoised audio signals from the received ones. Plot together the denoised signal with the originally received one. Create one plot for every audio signal.

   Use the *play_signal()* function to play the denoised signals - do you hear the difference between the originally received and the denoised one?

   Hint: the *scipy.signal.lfilter()* function can be useful when using the Wiener filter. You can compare your solution to the *scipy.signal.wiener()* function, which uses its own estimate of the noise instead of the noise measurement. Do you notice any differences? 3 points

8. (a) To investigate the effect of the filter order on the denoising, vary the filter order between $m = 2$ and $m = 20$. For what distance and for what filter order do you have the best denoised audio signal? Justify your answer. 1 point

   (b) Denote the best denoised signal from item "a." as $d(n)$, and calculate the cost function J of the Wiener problem assuming this signal is the ground truth. Print best filter order and the its respective cost function value for every received denoised audio signal. 1 point

   (c) Plot the filter order (from $m = 2$ to $m = 20$) versus cost function value for all received signals. 1 point

9. This is an **optional** bonus question, in which you can earn bonus points to make up for any points you lost earlier during this exercise. Note that if you already have the maximum amount of points for this exercise (i.e. grade 10), any additional bonus points will be ignored (that is, they do not carry over to other exercises or the exam).

   In this bonus question, we will characterize the signal over distance dependency by computing the signal-to-noise-ratio (SNR). For that:

   (a) Calculate the periodogram P_d using the denoised audio signals only for the highest filter order from Question 8. Calculate the periodogram P_x using the noised audio signals (see section "*5.2.8 WSS Processes in the Frequency Domain*" in the lecture notes). Plot the **denoised** estimated power spectra. **Hint:** be careful with the indexing. You can use *numpy.fft.fft()* and *numpy.fft.fftfreq()* functions. 2 bonus points

   (b) SNR is the ratio between the **ground truth** audio signal energy and the noise energy. The energy can be calculated as $E = \int_{-\infty}^{+\infty} P d\omega$. Approximating the integral by finite sum, calculate

      i. the SNR of the denoised audio signals;

      ii. the SNR of the noised audio signals. This can be done by relating the power spectrum of the noised signal with the power spectrum of the ground truth signal.

   In a single figure, plot these SNR values versus distance. 2 bonus points

   (c) Are the graphs different? Explain why. 1 bonus point

Deliverables

You are required to hand-in a report, showing your results and answering the questions about this exercise. Python scripts should be included as text in your report.

To help you with reporting, we provided a Jupyter notebook on Brightspace, which you can use as your report. We strongly encourage using the notebook, as hints and code snippets are already provided for you. After you have answered the questions and are finished with the assignment, you can export the Jupyter notebook as a PDF (instructions on how to do so are included in the notebook).

The report or exported Jupyter notebook should be dated and handed in before the deadline (see Course Schedule on Brightspace). The report should be handed in digitally by only one group member, but make sure the names and student numbers of all group members are included in the report or exported Jupyter notebook.

Bibliography

[1] M. Verhaegen, *Stochastic Processes for Physics Students: An Introduction*, Feb. 2017

[2] M. Hayes, *Statistical Digital Signal Processing and Modeling*, John Wiley & Sons, New York, 1996.