COM-303 - Signal Processing for Communications

Homework #8

Exercise 1. Autocorrelation function of a random process

Consider the following real-valued random process

$$x[n] = A\cos(\omega_0 n) + w[n]$$

where A is a Gaussian random variable with zero mean and variance σ_A^2 and where $w[n]$ is a zero-mean white noise process, independent of A, with variance σ_w^2.

(a) What is the autocorrelation of $x[n]$?

(b) Can we define the power spectral density of the process?

(c) Repeat (a) and (b) in the case when the cosine starts with a random phase offset, uniformly distributed over $[-\pi, \pi]$ and independent of all the other random variables.

Exercise 2. More white noise

A real-valued random process is defined by

$$x[n] = s[n] + w_0[n]$$

where $w_0[n]$ is a unit-variance, zero-mean white noise process and where $s[n]$ is defined as

$$s[n] = a\,s[n-1] + w_1[n]$$

with $a \in \mathbb{R}$ and $w_1[n]$ a unit variance, zero-mean white noise process independent of $w_0[n]$.

(a) Determine the autocorrelation $r_x[k]$.

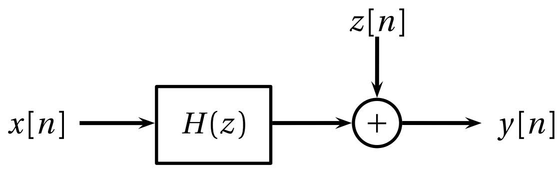
(b) Determine the power spectral density function $P_x(e^{j\omega})$.

Exercise 3. Filtering a Sequence of Independent Random Variables in Python

Let $x[n]$ be a real-valued Gaussian random process, with zero mean and variance $\sigma_x^2 = 3$. We filter the process with the FIR filter $h[n]$ where

$$h[1] = 1/2, \quad h[2] = 1/4, \quad h[3] = 1/4, \quad h[n] = 0 \ \forall \ n \neq 1, 2, 3$$

Moreover, at the output of the filter, we add white Gaussian noise $z[n]$ with unit variance. The system is shown in the following diagram:



(a) Write a routine in Python to generate N samples of the input process, N samples of the additive Gaussian noise and compute the output of the system.

(b) write a routine to estimate the power spectral density of the output

(c) compare the numerical estimation of the PSD with its theoretical value

Exercise 4. Analytic Signals & Modulation.

In this exercise we will explore a modulation-demodulation scheme commonly used in data transmission systems. Consider two real sequences $x[n]$ and $y[n]$, which represent *two* data streams we want to transmit. Assume that their spectrum is of lowpass type, i.e. $X(e^{j\omega}) = Y(e^{j\omega}) = 0$ for $|\omega| > \omega_c$. Consider further the following derived signal:

$$c[n] = x[n] + j y[n]$$

and the modulated signal:

$$r[n] = c[n] e^{j\omega_0 n}, \quad \omega_c < \omega_0 < \pi - \omega_c$$

(a) Set $\omega_c = \pi/6$, $\omega_0 = \pi/2$ and sketch $|R(e^{j\omega})|$ for whatever shapes you choose for $X(e^{j\omega})$, $Y(e^{j\omega})$. Verify from your plot that $r[n]$ is an analytic signal.

The signal $r[n]$ is called a *complex passband signal.* Of course it cannot be transmitted as such, since it is complex. The transmitted signal is, instead,

$$s[n] = \Re\{r[n]\}.$$

This modulated signal is an example of Quadrature Amplitude Modulation (QAM).

(b) Write out the expression for $s[n]$ in terms of $x[n]$, $y[n]$. Now you can see the reason behind the term QAM, since we are modulating with two carriers in quadrature (i.e. out of phase by 90 degrees).

Now we want to recover $x[n]$ and $y[n]$ from $s[n]$. To do so, follow these steps:

(c) Show that $s[n] + j(h[n] * s[n]) = r[n]$, where $h[n]$ is the Hilbert filter. In other words, we have recovered the analytic signal $r[n]$ from its real part only.

(d) Once you have $r[n]$, show how to extract $x[n]$ and $y[n]$.