COM-303 - Signal Processing for Communications

Solutions for Homework #8

Solution 1. Autocorrelation function of a random process

(a) The autocorrelation $r_x[n, n-k]$ is

$$
\begin{aligned}
r_x[n, n-k] &= \mathrm{E}[x[n]x^*[n-k]] \\
&= \mathrm{E}\left[A^2 \cos(\omega_0 n)\cos(\omega_0(n-k))\right] + \mathrm{E}[w[n]w[n-k]] + \\
&\qquad \mathrm{E}[A\cos(\omega_0 n)w[n-k]] + \mathrm{E}[A\cos(\omega_0(n-k))w[n]] \\
&= \mathrm{E}\left[A^2\right]\cos(\omega_0 n)\cos(\omega_0(n-k)) + \mathrm{E}[w[n]w[n-k]] + \\
&\qquad \mathrm{E}[A]\,\mathrm{E}[w[n-k]]\cos(\omega_0 n) + \mathrm{E}[A]\,\mathrm{E}[w[n]]\cos(\omega_0(n-k)) \\
&= \sigma_A^2 \cos(\omega_0 n)\cos(\omega_0(n-k)) + \sigma_w^2 \delta[k]
\end{aligned}
$$

where we have used $\mathrm{E}[A] = 0$.

(b) Since the autocorrelation does *not* depend only on k (i.e. the lag), the process is not wide-sense stationary process. Therefore, the power spectral density function can not be defined.

(c) Let's introduce a random phase term $\theta \in \mathscr{U}[-\pi, \pi]$ in the signal, obtaining $x[n] = A\cos(\omega_0 n + \theta) + w[n]$; the random phase term causes the expectation of the cosine to be zero independently of the frequency (since the average value of a sinusoid is zero):

$$
\mathrm{E}[\cos(\omega_0 n + \theta)] = 0.
$$

The autocorrelation now becomes

$$
\begin{aligned}
r_x[n, n-k] &= \mathrm{E}[x[n]x^*[n-k]] \\
&= \mathrm{E}\left[A^2 \cos(\omega_0 n + \theta)\cos(\omega_0(n-k) + \theta)\right] + \mathrm{E}[w[n]w[n-k]] + \\
&\qquad \mathrm{E}[A\cos(\omega_0 n + \theta)w[n-k]] + \mathrm{E}[A\cos(\omega_0(n-k) + \theta)w[n]] \\
&= \sigma_A^2 \mathrm{E}[\cos(\omega_0 n + \theta)\cos(\omega_0(n-k) + \theta)] + \sigma_w^2 \delta[k] \\
&= (\sigma_A^2/2)\mathrm{E}[\cos(\omega_0 k) + \cos(\omega_0(2n-k) + 2\theta)] + \sigma_w^2 \delta[k] \\
&= (\sigma_A^2/2)\cos(\omega_0 k) + \sigma_w^2 \delta[k]
\end{aligned}
$$

where we have used the trigonometric identity $\cos(\alpha+\beta) = (1/2)(\cos(\alpha-\beta)+\cos(\alpha+\beta))$. Since the autocorrelation $r_x[n, n-k]$ now only depends on the lag k, the process is wide-sense stationary and the power spectral density function is:

$$P_x(e^{j\omega}) = \frac{\sigma_A^2}{4}[\delta(\omega-\omega_0)+\delta(\omega+\omega_0)]+\sigma_w^2.$$

Solution 2. More white noise

(a) First of all:

$$\begin{aligned}
r_x[k] &= \mathrm{E}[x[n]x^*[n-k]] \\
&= \mathrm{E}[(s[n]+w_0[n])(s[n-k]+w_0[n-k])] \\
&= r_s[k]+r_{w_0}[k]
\end{aligned}$$

where the cross-products disappear because the two processes are independent and zero-mean. To find the autocorrelation of $s[n]$ we exploit its recursivity:

$$\begin{aligned}
r_s[k] &= \mathrm{E}[s[n]s^*[n-k]] \\
&= \mathrm{E}[(a\,s[n-1]+w_1[n])s[n-k]] \\
&= a\,r_s[k-1].
\end{aligned}$$

We also have, for $k=0$,

$$\begin{aligned}
r_s[0] &= \mathrm{E}\left[(a\,s[n-1]+w_1[n])^2\right] \\
&= a^2\mathrm{E}\left[s^2[n-1]\right]+1 \\
&= a^2\,r_s[0]+1
\end{aligned}$$

so that

$$r_s[0] = \frac{1}{1-a^2}.$$

Then, by induction

$$r_s[1] = a\,r_s[0] = \frac{a}{1-a^2}$$

$$r_s[2] = a\,r_s[1] = \frac{a^2}{1-a^2}$$

$$\ldots$$

$$r_s[k] = \frac{a^k}{1-a^2}$$

and, similarly, for negative values of the index we have

$$r_s[k] = \frac{a^{|k|}}{1-a^2}.$$

In the end

$$r_x[k] = \frac{a^{|k|}}{1-a^2} + \delta[k].$$

(b)

$$P_x(e^{j\omega}) = \frac{1}{1+a^2-2a\cos\omega} + 1$$

Solution 3. Filtering a sequence of independent random variables in Python

Below is the code that computes realizations of the output and estimates the PSD. For a comparison with the theoretical value, we need to compute the exact PSD of the output process. Call $s[n]$ the signal coming out of the filter $H(z)$; then:

$$\begin{aligned}
r_y[k] &= \mathrm{E}\left[(s[n]+z[n])(s[n-k)+z[n-k]\right] \\
&= \mathrm{E}\left[(s[n]s[n-k]\right]+\mathrm{E}\left[z[n]z[n-k]\right] \\
&= r_s[k]+\delta[k] \\
&= h[n]*h[-n]*r_x[k]+\delta[k] \\
&= 3\,h[n]*h[-n]+\delta[k]
\end{aligned}$$

so that the PSD is

$$P_y(e^{j\omega}) = 3\,|H(e^{j\omega})|^2 + 1$$

To compute $|H(e^{j\omega})|^2$:

$$\begin{aligned}
|H(e^{j\omega})|^2 &= \left|(1/4)(2e^{-j\omega}+e^{-j2\omega}+e^{-j3\omega})\right|^2 \\
&= (1/16)\left|e^{-2j\omega}\right|^2\left|2e^{j\omega}+1+e^{-j\omega}\right|^2 \\
&= (1/16)\left|1+3\cos\omega+j\sin\omega\right|^2 \\
&= (1/16)(6+6\cos\omega+4\cos 2\omega)
\end{aligned}$$

so that in the end:

$$P_y(e^{j\omega}) = \frac{17}{8}+\frac{9}{8}\cos\omega+\frac{3}{4}\cos 2\omega$$

```
from __future__ import division
import numpy as np
import matplotlib.pylab as plt
import scipy as sp
import scipy.signal
```

```
def compute_y(N):
    h = np.array([0, 0.5, 0.25, 0.25])
    # generate x[n]
    x = np.sqrt(3) * np.random.randn(N)
    # filter it with h[n]
    x1 = np.concatenate((x[-3:], x, x[:3]))
    y1 = sp.signal.lfilter(h, 1., x1)[3:N + 3]
    # generate z[n]
    z = np.random.randn(N)
    # generate y[n]
    y = y1 + z
    return y

def estimate_psd(N, M):
    """
    :param N: the length of the input vector
    :param M: the number of iterations
    :return:
    """
    PSD = np.zeros(N, dtype=float)

    for loop in range(M):
        PSD += np.abs(np.fft.fft(compute_y(N))) ** 2 / N
    return PSD / M


if __name__ == '__main__':
    # compare the experimental PSD to the theoretical PSD
    # for varying values of M
    for M in [50, 500, 5000]:
        PSD = estimate_psd(N, M)
        omega = np.linspace(0, 2 * np.pi, num=N)
        PSD_theo = 17 / 8. + 9 / 8. * np.cos(omega) + 3 / 4. * np.cos(2 * omega)
        plt.figure(num=count, figsize=(5, 3), dpi=90)
        plt.plot(PSD, 'r--', label='estimate')
        plt.plot(PSD_theo, 'b-', hold=True, label='theoretical')
        plt.legend()
        plt.show()
```

Solution 4. Analytic Signals & Modulation

(a) The modulation theorem tells us that $R(e^{j\omega})$ is the convolution of $C(e^{j\omega})$ with $\tilde{\delta}(\omega-\omega_0)$, i.e. $R(e^{j\omega}) = C(e^{j(\omega-\omega_0)})$. Since $C(e^{j\omega}) = X(e^{j\omega}) + jY(e^{j\omega})$ and both $X(e^{j\omega})$, $Y(e^{j\omega})$ live on the $[-\omega_c, \omega_c]$ interval, $R(e^{j\omega})$ lives on the $[\omega_0 - \omega_c, \omega_0 + \omega_c]$ interval. Since $\omega_c < \omega_0 < \pi - \omega_c$, this interval is entirely contained in the $[0, \pi]$ interval, thus $r[n]$ is analytic.

(b) Clearly

$$r[n] = (x[n] + jy[n])(\cos(\omega_0 n) + j\sin(\omega_0 n))$$

so that

$$s[n] = x[n]\cos(\omega_0 n) - y[n]\sin(\omega_0 n).$$

(c) Let $g[n] = s[n] + j(h[n] * s[n])$. We know from the derivation on page 118 in Chapter 5 that

$$G(e^{j\omega}) = \begin{cases} 2S(e^{j\omega}) & \text{for } 0 \le \omega < \pi \\ 0 & \text{for } -\pi \le \omega < 0 \end{cases}$$

so let us consider the positive-frequency part of $S(e^{j\omega})$. We can see from (**??**) that this is the sum of $X(e^{j(\omega-\omega_0)})/2$ and $jY(e^{j(\omega-\omega_0)})/2$, both of which are shifted versions of $X(e^{j\omega})$ and $Y(e^{j\omega})$ which live between $\omega_0 - \omega_c$ and $\omega_0 + \omega_c$, i.e. in the positive-frequency part of the spectrum. We can therefore write:

$$G(e^{j\omega}) = (X(e^{j\omega}) + jY(e^{j\omega})) * \tilde{\delta}(\omega - \omega_0)$$

which, in the time domain becomes

$$g[n] = (x[n] + jy[n])e^{j\omega_0 n} = r[n].$$

(d) $x[n] = \Re\{r[n]e^{-j\omega_0 n}\}$ and $y[n] = \Im\{r[n]e^{-j\omega_0 n}\}$.