Datorlab 2

Slumptal, slumpsignaler och filtrering.

In this computer exercise you will learn how to generate independent random numbers having a specified probability distributions. The random numbers will then be used to simulate stationary random functions. The random functions will be generated in the spectral domain and have normal (Gaussian) distribution: A stationary normally distributed random function is defined by its expected value and the expected periodogram, describing the average distribution of power over frequencies in the signal.
You need a copy of Matlab that includes the Statistical Toolbox. Most versions of Matlab (including our in the lab) have this toolbox included. On some occasion you will need access to data files including samples that you will be asked to analyze. All necessary files are downloadable from the course home page
http://www.math.chalmers.se/Stat/Grundutb/CTH/ess011/1415/files/labfiles.zip.
Please download the labfiles.zip file and uncompress it at the directory you plan to use for the computer exercises.

1 Preparatory exercises

1. Give a 0.95-quantile of a standard normal distribution.
2. Write down definition of average power of a signal and of "periodogram" of a signal.
3. Read Chapter 2 in "Additional material for ESS011" compendium, click . Find in Section 2.2.2 answers to the following questions;
   (a) How are phases distributed in the normal random functions?
   (b) Are the phases dependent?

2 How to generate random numbers

Let Y be a uniformly distributed random variable (between 0 and 1), and let F be a distribution function. Then a random variable X is said to be a random variable distributed according to F if $F(X) = Y$, i.e. if[1]

$$X = F^{-1}(Y).$$

[1] Note that $F^{-1}(y)$ is the *inverse* function of F (at instant y), *not* the *reciprocal* value $1/F(y)$ of $F(y)$.

If, for instance, F is

$$\begin{aligned}
\text{Rayleigh:} \quad & Y = F(X) = 1 - \mathrm{e}^{-\left(\frac{X}{2}\right)^2} && \Leftrightarrow && X = (-\ln(1-Y))^{1/2} \\
\text{normal:} \quad & Y = F(X) = \Phi\left(\frac{X-m}{\sigma}\right) && \Leftrightarrow && X = m + \sigma\Phi^{-1}(Y) \\
\text{Gumbel:} \quad & Y = F(X) = \exp\left(-\mathrm{e}^{-\frac{X-b}{a}}\right) && \Leftrightarrow && X = b - a\ln(-\ln Y),
\end{aligned}$$

then X is a Rayleigh, normally, and Gumbel distributed random variable, respectively. In Matlab uniformly distributed random variables ("random numbers") are generated by means of the command `rand`. We will use it here to produce 2000 normally-distributed random numbers:

```
m=10; sigma=3;
x=m+sigma*norminv(rand(2000,1));
plot(x,'.'), grid on
```

Here we *really* encourage you to use the command `randn` instead, i.e

```
x=m+sigma*randn(2000,1);
```

Problem 1: Simulate 1000 standard Rayleigh random numbers. Since Rayleigh distribution is a special case of the Weibull distribution use the Weibull probability plot (see Datorlab 1) to check whether the simulated random numbers seam to be Rayleigh distributed. Comment on the result.

To generate random numbers in Matlab, one can also make use of the commands `wblrnd` (Weibull), `normrnd` (normal), and `raylrnd` (Rayleigh) from the commercial Statistics Toolbox.

3 Random functions

3.1 Square wave - signal

A square wave x with period $T = 2$ can be defined using Fourier series as follows

$$x(t) = \sum_{k=1,3,\ldots} 2|a_k|\cos(\omega_k\, t + 3\pi/2), \quad \omega_k = 2\pi\, k/T, \quad |a_k| = 2/\omega_k. \tag{1}$$

Note that the average value of x is zero and hence $a_0 = 0$.

In practice the infinite sum has to be truncated, i.e. one uses only finite number of terms in the sum (1), viz

$$x_K(t) = \sum_{k=1,3,\ldots,K} \frac{4}{\omega_k}\cos(\omega_k\, t + 3\pi/2). \tag{2}$$

The following Matlab script can be used to evaluate $x_K(t)$. In the code `dt` and `N` define the sampling rate and duration of the signal $x_K(t)$, respectively. Furter `K` defines the truncation threshold

```
T=2;  N=2^8-1;
dt=T/N;  t=(0:N-1)*dt;
K=15;  w=2*(1:2:K)*pi/T;
```

```
a=2./w;  phi=3*pi/2;
xK = zeros(1,N);
for k=1:length(w);
   xK=xK+2*a(k)*cos(w(k)*t+phi);
end
```

Problem 2: Use the code to compare $x(t)$ defined in (1) with the approximation (2). Illustrate the comparison by a suitable plot.

For any Fourier series $x(t) = \sum_{k>0} 2|a_k|\cos(\omega_k\, t + \phi_k)$ the function

$$s(\omega_k) = 2|a_k|^2, \qquad k \geq 1, \tag{3}$$

called periodogram, describes how the average power of the signal is distributed over angular frequencies ω_k. The average power V of the signal x is given by

$$V = \frac{1}{T}\int_0^T x(t)^2\, dt = \sum_{k=1}^{+\infty} s(\omega_k),$$

since one has assumed that $\int_0^T x(t)\, dt = 0$ implying that $a_0 = 0$. The truncated signal $x_K(t)$, see (2), was defined by setting $|a_k| = 0$ for $k > K$ and hence its average power $V = \sum_{k=1}^{K} s(\omega_k)$.

Problem 3: Compute the average power of the square wave x and the average power of the approximation x_K, $K = 15$. How much energy (in %) is missing in x_K?

Problem 4: Compute and plot periodogram $s(\omega_k)$ of x_K, $K = 15$, for $k = 1, 2, \ldots, 25$. (Hint: a script can be found in the appendix to this lab.)

3.2 Finding periodogram using `ftt`

Consider a periodic signal $x(t)$, with period T, represented in form of a time series $x(t_i)$, $t_i = i \cdot dt$ and $T = N \cdot dt$. Neglecting the so called "aliasing", that may occure for signals having unbounded periodogram, then the first $(N-1)/2$ values of the periodogram can be found using the `fft` (Fast Fourier Transform) function. This will be illustrated using signal `xK` created in Section 3.1. The signal has, for all $k > K$, $s(\omega_k) = 0$, see (3). Hence one can recover all $s(\omega_k)$ using the `fft` algorithm. This will be done using the following Matlab script

```
N2 = (N-1)/2;
FxK=fft(xK);
S = fftshift(abs(FxK).^2)/N^2;
S = 2*S(N2+2:N);
dw = 2*pi/T;                                          % T=N*dt
w = (1:N2)*dw;
semilogy(w, S,'*')
axis([0 2*pi*25/T 1e-35 1])
```

Problem 5: Is the periodogram computed using `ftt` good approximation of the periodogram of x_K?

3.2.1 Example: periodogram of measured sea waves

In the numerical examples above, we used artificial data (the square wave) to illustrate some concept from signal processing. We will now consider *real* measurements from the North Atlantic Ocean. The data set contains sea level elevation (in meters) at some location on NAO. The data is seen as a sequence of sea waves. Command `help sea` gives some important informations about the waves; that the sea is build up by the wind waves and swell; the average wave period is 4 seconds while swell has about 10 seconds period

Now, load the data set `sea.dat` and read about the measurements; then plot it:

```
xs=load('sea.dat');
help sea
size(xs)
plot(xs(:,1),xs(:,2),'')
N=length(xs); dt=0.25; T=N*dt;
```

Problem 6: Estimate the periodogram of the sea level data using `fft`. *Hint:* replace `xK` by `xs` in the first script presented in Section 3.2.

Command `help sea` gave information that lower peak period in the signal is `Tp2 = 11.5s` while higher peak period is `Tp1=5.6s`. Note that the square wave has period 2.

Problem 6a: Zoom the signal. Can you see the two main periods?

The angular frequency for which a periodogram attains its maximum ω_p is called the peak frequency. Similarly $T_p = 2\pi/\omega_p$ is called peak period. The significant wave height H is defined as $H = 4\sqrt{V}$, where V is the signals average energy. Recall that the average energy is equal of the sum of periodogram.

Problem 7: Estimate the peak peak frequency ω_p , the peak period T_p and the significant wave height H of the sea level data `sea`. (Hint: a script can be found in the appendix to this lab.)

The measured signal seams to vary in the unpredictable way. The computed periodogram of the signal is very noisy which we attribute to randomness of the signal. In oceanography one is often using a theoretical power spectrum to describe the distribution of energy between different frequencies. The power spectrum $S(\omega)$ is defined for any $\omega \geq 0$ while periodogram only for ω_k. Roughly, periodogram $s(\omega_k) \approx S(\omega_k)\, d\omega$.

There are many proposed spectra in the literature. Popular Pierson-Moskowitz spectrum, often used to model fully developed sea, will be used here. The spectrum depends on two parameters H and ω_p and is given by

$$S(\omega) = c\omega^{-5} \exp(-1.25\omega_p^4/\omega^4), \qquad c = 0.3125\, H^2\omega_p^4. \tag{4}$$

Problem 8: Use the estimated values of ω_p and H to evaluate the "theoretical" periodogram. (Hint: a script can be found in the appendix to this lab.) Does the theoretical periodogram describe well the observed periodogram?)

One can see that periodogram varies a lot around the "thoeretical" power spectrum. In addition energy contents for low frequencies, corresponding to ocean swell, seams to be missed by the "theoretical spectrum". In the following section we shall study random functions which have a random (very irregular) periodograms.

3.3 Random function

Next we will present random functions $X(t)$, say, having expectation $E[X(t)] = 0$ and expected periodogram $s(\omega_k)$, say, for example periodogram of a square wave or sea waves derived using Pierson-Moskowits power spectrum. As before we define

$$|a_k| = \sqrt{s(\omega_k)/2}, \qquad \omega_k = 2\pi k/T.$$

A periodic random function is defined as follows

$$X(t) = \sum_{k=1}^{+\infty} 2|A_k| \cos(\omega_k\, t + \Phi_k), \quad |A_k| = |a_k| R_k/\sqrt{2}, \quad \Phi_k = 2\pi\, U_k, \tag{5}$$

where R_k are independent Rayleigh distributed random numbers while U_k are independent uniformly distributed random variables. (It is not difficult to show that $E[X(t)] = 0$ and that $X(t)$ is normally distributed.) In practice the infinite series in (5) has to be truncated leading to the following approximation

$$X_K(t) = \sum_{k=1}^{K} 2|A_k| \cos(\omega_k\, t + \Phi_k). \tag{6}$$

3.3.1 Example of random function having theoretical periodogram of the sea waves

In this example we will use the periodogram defined using the Pierson-Moskowitz power spectrum of sea waves, viz

$$s(\omega_k) = S(\omega_k)\, d\omega,$$

where $S(\omega)$ is defined by (4). The following code can be used to simulate sea waves having Pearson-Moskowitz spectrum

```
 t=xs(:,1)';
 a = sqrt(SPM*dw/2);
 R = sqrt(- 2 *log(rand(1,length(w))));                     % Rayleigh random numbers
 U=rand(1,length(w));
 A=a.*R/sqrt(2);                                             % random amplitudes
 Phi=2*pi*U;                                                 % random phases
 Xs = zeros(1,N);
 for k=1:length(w);
     Xs = Xs + 2*A(k)*cos(w(k)*t + Phi(k));
 end
figure
plot(t,xs(:,2))
hold on
plot(t,Xs,'r')
```

Problem 9: Zoom the last plot. Is the variability of the measured signal xs and the simulated `Xs` similar?

Problem 9a: Estimate periodogram of the simulated random signal `Xs` and compare it with Pierson-Moskowitz spectrum `SPM*dw` evaluated in Problem 8. (Use `fft` algorithm given in Section 3.2.1 to evaluate the periodogram of `Xs`.) Are these similar?

The periodogram of `Xs` is a random signal. The expected value of the periodogram is equal to the Pierson-Moskowitz spectrum.

Problem 10: Simulate `Ns` times, for example `Ns=10`, the Random signals `Xs`. For each simulation estimate a periodogram. (You will get `Ns` periodograms.) Take the average of the periodograms and compare it with the Pierson-Moskowitz spectrum `SPM*dw` evaluated in Problem 8. Is the average periodogram closer to the theoretical one? What would happen if one has increased the number of simulations `Ns`?

4 Appendix

In the appendix some MATLAB scripts are given.

Problem 4:

```
K=15;   T=2;
wod=2*(1:2:K)*pi/T;
figure
semilogy(wod, 2*(2./wod).^2,'ro')
hold on
we = 2*pi*[2:2:16 17:25]/T;
plot(we,1E-30,'ro')
```

Problem 6:

```
N2 = (N-1)/2;
Fxs=fft(xs);
S= fftshift(abs(Fxs).^2)/N^2;
S= 2*S(N2+2:N);
dw = 2*pi/T;
w = (1:N2)*dw;
figure
plot(w,S)
hold on
```

Problem 7:

```
[Smax ind]=max(S);
wp=w(ind); Tp=2*pi/wp; H=4*sqrt(sum(S))
```

```
[wp Tp H]
```

Problem 8:

Given values of peak angular frequency ω_p and significant wave height H the "theoretical" periodogram can be evaluated using the following code

```
c=0.3125 *H^2*wp^4;
SPM= c*w.^-5.*exp(-1.25.*wp.^4./w.^4);
plot(w,SPM*dw,'r','linewidth',3)
axis([0 4 0 4E-3])
```