ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

School of Computer and Communication Sciences

Software-Defined Radio: A Hands-On Course — Midterm, October 30, 2013

Problem 1. (8 p.)

```
% define the parameters
f1=100; f2=300;
Fs=1000;
NFFT=1024;
T=10;  %seconds

t=0:1/Fs:T; % time vector
vf=-Fs/2:Fs/NFFT:Fs/2-Fs/NFFT; %frequency vector

s=sin(2*pi*f1*t)+sin(2*pi*f2*t);

%plot the frequency spectrum
figure, plot(vf,abs(fftshift(fft(s,NFFT)))), xlabel('f[Hz]'), ylabel('|s_F(f)|')
```

Problem 2. (8 p.)

(a) The signal $s(t)$ is obtained by multiplying the sum of sinusoids by a rectangular window. The Fourier transform of a sinusoid is given by two symmetric Dirac functions. The Fourier transform of a rectangular pulse is a sinc. So, the Fourier transform of $s(t)$ will be the convolution between four Dirac functions and a sinc.

(b) The spectrum is represented from $-f_s/2$ to $f_s/2$, so $f_s = 1000$Hz. The main lobes have width $2W = 40$Hz and the secondary lobes have width $W = 20$Hz. From $W = 1/T$, we obtain $T = 50$ms.

(c) The lobes are narrower for the second figure. As their width is inverse proportional to T, we conclude that T has increased.

(d) Because the frequencies f_1 and f_2 did not change, but the location of the peaks changed, we conclude that the sampling frequency changed. The new value is $f_{s2} = 500$Hz. In this case $W = 10$Hz, so $T = 100$ms.

Problem 3. (3 p.)

```
% xcorr;
ip1=xcorr(a,b);
ip1=ip1(length(a));

% conv
ip2=conv(a,fliplr(conj(b)));
ip2=ip2(length(a));

% vector multiplication
ip3=a*b';
```

The third method is the most efficient: there are no unnecessary computations. For `xcorr` and `conv`, we only need the middle value, and discard all the others.

Problem 4. (3 p.) Let `s` be the given sequence:

`0 0 0 0 -1 0 -1 0 -2 2 2 -2 -3 0 1`.

(a) The zeroes are at the beginning of the sequence, so `s1` is longer.

(b) `length(s1)+length(s2)`=`length(s) - number of initial zeroes +1`= `15-4+1=12`;

(c) Because `s1` is the longest, `length(s)` will be `2*length(s1)-1`. Therefore, `length(s1)=8` and `length(s2)=4`.

Problem 5. (12 p.)

```
% SDR 2013 midterm - Problem 5

clear all;close all;clc;

%the number of transmitted symbols (not including the training sequence)
nSymbols=1000;

% load the training sequence, named traninig
load training;

%load the received signal containing the delay, the training symbols and
%the modulation symbols
load received

%% (a) estimate the start of the QAM symbol sequence in the received data and the
%rotation

corrResult=xcorr(received, training);
corrResult(1:length(received)-1)=[];
%start position of the training sequence
[M, pos]=max(abs(corrResult));
%start position of the QAM symbols
startPos=pos+length(training);

%estimate the rotation angle
rTraining=received(pos:pos+length(training)-1);
estimRotation=mean(rTraining./training);


%% (b) pull out the noisy symbol sequence and correct for the phase rotation

rSymbols=received(startPos:startPos+nSymbols-1);
rSymbolsCorrected=rSymbols*estimRotation';


%% (c) plot the complex plane noisy symbols
scatterplot(rSymbols), title ('before rotation correction')
scatterplot(rSymbolsCorrected), title ('after rotation correction')


%% (d) decode the bit sequence
bitsREstimated=sign(real(rSymbolsCorrected));
bitsIEstimated=sign(imag(rSymbolsCorrected));


% you can compare the result with the bit sequences stored in bitsR.mat
```

```
% and bitsI.mat

load bitsR
load bitsI

errR=nnz(bitsR-bitsREstimated)
errI=nnz(bitsI-bitsIEstimated)
```

Problem 6. (3 p.)

```
...
y_up = upsample(y, USF);
...
% the noise is complex
noise = sigma/sqrt(2) * (randn(size(z))+1j*randn(size(z)));
...
% the receiver matched filter is the time-inversed conjugate transmitter filter
y_rec = conv(r, fliplr(conj(h)));
...
```

Problem 7. (3 p.)

(a)
```
%% determining the square of each element of vector X

x2=x.^2;
```

(b)
```
%% finding the position of the maximum element of vector X

[~,pos]=max(x);
```

(c)
```
%% mapping a vector of integers between 0 and 3, denoted X,
% to QAM constellation points
% you know that qammap=[-1+1i, -1-1i, 1+1i, 1-1i]

x_qam=qammap(x+1);
```