FOUNDATIONS OF SIGNALS AND SYSTEMS
18.5 Homework assignment
Prof. T. Erseghe

Exercises 18.5

Solve the following MatLab problems:

1. The file 'ex18_5_1.mat' contains in vector x some ECG samples taken with spacing $T = 1/125\,\mathrm{s}$. After removing the signal average value (use the mean MatLab function), plot the signal as well as its Fourier transform in absolute value, and determine the position $\omega_0 > 0$ of the first peak. By resorting to the expression $\omega_0 = 2\pi/T_p$, identify the ECG period $T_p = 2\pi/\omega_0$. You can use the MatLab function "[maxval,pos] = max(abs(X))" for this.

2. The file 'ex18_5_2.mat' contains in vector x some ECG samples taken with spacing $T = 1/125\,\mathrm{s}$ and corrupted by a sinusoidal noise. After removing the signal average value (use the mean MatLab function), plot the signal as well as its Fourier transform in absolute value. Then, filter the signal with an high-pass filter that rejects all pulsations in the range $|\omega| < \pi$, by applying a selection in the Fourier domain and then by applying an inverse transform. The sinusoidal noise should be absent in the filtered signal.

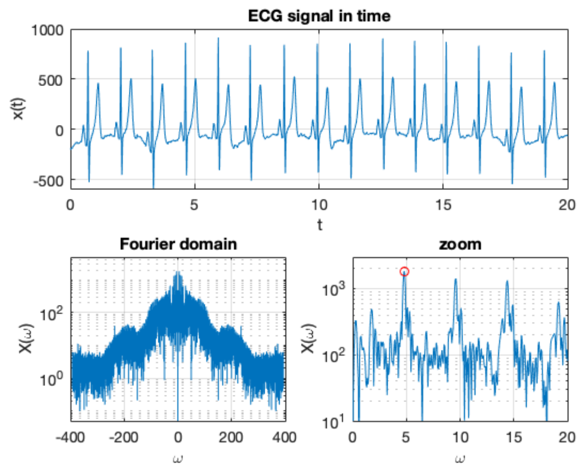
Solutions.

1. In the code we first subtract the average value, then use a trick to increase the definition in the Fourier domain (i.e., to increase the value of N), namely that of adding zero-valued contributions at the end of the signal. The search for the maximum is restricted in the range $[4, 8]$ since this is the range we can identify by looking at the plots. We also display the estimated period which turns out to be $T_p = 1.3151$.

```
load('ex18_5_1.mat') % defines t, x, T
x = x - mean(x);
x = [x, zeros(1,2*length(x))]; % trick to tighthen
     Fourier sampling
N = length(x);
t = (0:N-1)*T;
X = fftshift(T*fft(x));
om = (-round((N-1)/2):round(N/2)-1) *2*pi/(N*T);

% find max: range [4,8] set by looking at the plot
[maxval,pos] = max(abs(X).*(om>4).*(om<8));
om0 = om(pos); % estimated omega0
disp(['estimated Tp = 2 pi/omega0 = ' num2str(2*pi
   /om0)])

figure(1)
subplot(2,1,1)
plot(t,x)
grid
xlabel('t')
ylabel('x(t)')
title('ECG signal in time')
axis([0 20 ylim])
subplot(2,2,3)
semilogy(om,abs(X))
grid
xlabel('\omega')
ylabel('X(\omega)')
title('Fourier domain')
subplot(2,2,4)
semilogy(om,abs(X))
hold on
semilogy(om0,maxval,'ro')
grid
xlabel('\omega')
ylabel('X(\omega)')
axis([0 20 1e1 3e3])
title('zoom')
```



2. We apply the same tricks as in the previous exercise. The high-pass filter is implemented by multyplying entrywise the Fourier coefficients X by a selection of the pulsation values "abs(om)>pi" which is active (i.e., equal to one) only for $|\omega| > \pi$. The filtered signal is then obtained by inverse transform.

```
load('ex18_5_2.mat') % defines t, x, T
x = x - mean(x);
x = [x, zeros(1,2*length(x))]; % trick to tighthen
    Fourier sampling
N = length(x);
t = (0:N-1)*T;
X = fftshift(T*fft(x));
om = (-round((N-1)/2):round(N/2)-1) *2*pi/(N*T);
Y = X.*(abs(om)>pi); % filter signal
y = ifft(ifftshift(Y)/T); % filtered signal in
   time

figure(1)
subplot(2,2,1)
plot(t,x)
grid
xlabel('t')
ylabel('x(t)')
title('time domain - distorted')
axis([0 20 ylim])
subplot(2,2,2)
plot(t,y)
grid
xlabel('t')
```

```
ylabel('y(t)')
title('time domain - filtered')
axis([0 20 ylim])
subplot(2,2,3)
semilogy(om,abs(X))
axis([xlim 5e-2 5e3])
grid
xlabel('\omega')
ylabel('X(\omega)')
axis([0 20 1e1 1e4])
title('Fourier domain - distorted')
subplot(2,2,4)
semilogy(om,abs(Y))
grid
xlabel('\omega')
ylabel('Y(\omega)')
axis([0 20 1e1 1e4])
title('Fourier domain - filtered')
```