1 Noise reduction methods.

1.1 Random noise (chapter 6.2).

Generate a random noise signal 'rn1' of length 1000. The samples are generated from a normal probability density function N(m,σ), where m is the mean-value and σ is the standard deviation.

%generate a noise signal rn1 of 1000 samples
rn1=0.6 +sqrt(0.1)*randn(1,1000); %gaussian noise, m=0.6, σ^2=0.1
figure(1); subplot(3,1,1); plot(rn1); %show the noise-signal 'rn1'

Question 1.

Verify that the generated signal rn1 is generated from a normal probability density function $N\left(0.6,\sqrt{0.1}\right)$ *by showing its histogram and by computing its mean-value* $m=\frac{1}{N}\sum_{i=1}^{N}x_i$ *and its variance* $\sigma^2=\frac{1}{N-1}\sum_{i=1}^{N}(x_i-m)^2$.

%Compute and show the histogram in 20 bins.
[y,x]=hist(rn1,20); %Y=the number of samples in each bin, x=the bin midpoint
figure(1); subplot(3,1,2); bar(x,y); %show histogram
%Print on the screen the values of y and x and compare these values with the showed
%histogram in figure 1.
y
x
%Estimate m and σ^2 of the random signal rn1, and print the values on the screen.
%Without semicolon the result is printed on the screen.
m=mean(rn1) %mean value
v=var(rn1) %variance
s=std(rn1) %standard deviation

If the histogram should be interpreted as a probability density function its area must be normalized to one.
The area of bin k is interpreted as the probability that a sample is contained in the bin-interval $x_k \pm \Delta x$ with a probability of $\frac{y_k}{\sum_{k=1}^{N} y_k}$ where x_k is the centre of the bin-interval, Δx is the half-width of the bin, y_k is the number of samples in bin k, and N is the number of bins in the histogram.

%Normalize the histogram
ynorm=y/sum(y); %probability
figure(1);subplot(3,1,3);bar(x,ynorm); %probability function

Question 2.

From the generated normalized histogram (x,ynorm) select a bin (=x) and note the probability (=ynorm) that a sample is contained in the bin-interval .
Compute by hand the probability for the selected bin-interval by assuming a probability density function $N\left(0.6,\sqrt{0.1}\right)$ *and by using table-values for a N(0,1) probability density function (the table is included in the lab manuscript).*

Are the two probabilities equal?
Comment on your result!

1.2 Averaging.

1.2.1 Ensemble averaging (repetitive measurement, chapter 6.5.7).

Generate a signal 's' as a sum of two cosine signals with different frequency using the following commands:

```
n=0:399; %number of samples
f1=3000;
f2=5000;
fs=20*12000; %sample frequency
s1=cos(2*pi*f1*n/fs);
s2=sin(2*pi*f2*n/fs);
s=s1+s2; %signal
figure(2);subplot(4,1,1);plot(n,s);
```

Suppose we have 50 repetitive measurement of the signal 's' each added by random noise $N\left(0,\sqrt{0.5}\right)$.
These 50 measurements are generated as:

```
S=repmat(s,50,1); %50 copies of the signal s
RN=sqrt(0.5)*randn(50,400); %50 random noise signals of length 400 samples
MS=S+RN; %add random noise to each signal s
%Show two of the 50 measurements, number 17 and 32
figure(2);subplot(4,1,2);plot(n,MS(17,:));
figure(2);subplot(4,1,3);plot(n,MS(32,:));
```

Compute the average (ensemble averaging) of the first 9 measurements in MS.

```
MS9=MS(1:9,:); %select the first 9 measurements in MS
y9=mean(MS9); %averaged signal
%show the averaged signal
figure(2);subplot(4,1,4);plot(n,y9);
```

Question 3.

- Compute the ensemble- average of 49 measurements and show the averaged signal. Compare the plots of the two averaged signals y9 and y49 and comment the result.

- Theoretically how much do the signal-to-noise-ratio (SNR) increase when doing ensemble averaging using 9 respective 49 measurements?
- Verify the increase of SNR when using 49 measurements
(Hint: compute the ratio of the standard deviation of the original noise signal and the standard deviation of the ensemble-averaged noise signal).

1.2.2 Time averaging by filtering (chapter 6.5.5 and 10.4.2).

Averaging is now done along the signal (=time averaging). This can be done by a digital low-pass filter.
Use a gaussian low-pass filter which is designed as:

```
sigma=1.4; %decide the cut-off frequency of the filter
M=2*round(4*sigma)+1; %the length of the filter (odd)
x=-(M-1)/2:(M-1)/2; %sample points
g=exp((-x.*x)/(2*sigma*sigma)); %the digital filter
g=g/sum(g); %the gain=1
G=fft(g,512); %the frequency response function
%Show the filter
figure(3); subplot(3,1,1);stem(x,g); %in the time domain
%In the frequency domain
figure(3);subplot(3,1,2);stem(-256:255,fftshift(abs(G))); %Amplitud function
figure(3);subplot(3,1,3);stem(-256:255,fftshift(angle(G))); %Phase function
```

Question 4.

Generate a measurement sb=s+rn2 where s is the signal you used before and rn2 is a random noise signal $N\left(0,\sqrt{0.5}\right)$. *Do averaging by filtering the measurement sb and show the filtered signal y.*

```
rn2=.....; %noise signal
sb=.....; %measurement
%Show the magnitude frequency response of the measurement and the filter.
SB=fft(sb,512); %frequency representation of the measurement
figure(4); subplot(2,1,1);stem(0:255,abs(G(1:256)); %filter
figure(4); subplot(2,1,2);stem(0:255,abs(SB(1:256)); %measurement
y=conv(sb,g); %filtering by convolution in the time domain
figure(5);subplot(3,1,1);plot(n,s); %signal
figure(5);subplot(3,1,2);plot(n,sb); %measurement
figure(5);subplot(3,1,3);plot(y); %filtered signal
```

Question 5.

-Tune the filter, i.e. select a new sigma-value that fits the measurement by changing the cut-off frequency of the gaussian filter.
Hint: for help in the tuning, study the frequency representation of the measurement and the filter for different sigma-values.

- Use the tuned filter to filter the measurement sb (reuse the old code!) and comment on the result.

1.3 The Fourier transform (chapter 10.4.2).

If the signal s has a narrow bandwidth and the noise has a wide bandwidth the SNR can be improved by doing a Fourier transform (FT) of the measurement sb.

```
SB=fft(sb,1024); %FT in 1024 points
%Show results
figure(6);subplot(4,1,1);plot(n,s); %signal
figure(6);subplot(4,1,2);plot(n,rn2); %noise
figure(6);subplot(4,1,3);plot(n,sb); %measurement
figure(6);subplot(4,1,4);plot(0:511,abs(SB(1:512))); %FT
```

Question 6.

- Identify the two signals s1 and s2 and also the noise rn2 in the Fourier-transform of the measurement sb.

- Check how much the SNR increase in the Fourier-transformed measurement SB compare to the SNR in the measurement sb.

Hint: estimate the SNR in the measurement sb and in the Fourier-transform of sb (=SB).

1.4 Auto-correlation (chapter 6.2.5 and 6.5.8).

The autocorrelation can be used to reduce random noise in periodic signals.
If s1 and rn2 are uncorrelated, the autocorrelation of the measurement sb1=s1+rn2 is the sum of the autocorrelation of s1 and rn2, i.e. $R_{sb1sb1}(\tau)=R_{s1s1}(\tau)+R_{rn2rn2}(\tau)$.

```
sb1=s1+rn2;
auto=xcorr(sb1,'unbiased'); %compute the autocorrelation
%Show results
figure(7);subplot(3,1,1);plot(n,s1);
figure(7);subplot(3,1,2);plot(n,sb1);
figure(7);subplot(3,1,3);plot(-200:200,auto(400-200:400+200)); %autocorrelation
```

Question 7.

- How do you interpret the "x-scale" in the autocorrelation signal?

- At which x-value in the autocorrelation do you see the influence of the noise?

Explain by computing according to the formula: $R_{sb1sb1}(\tau)=R_{s1s1}(\tau)+R_{rn2rn2}(\tau)$ and show all three autocorrelations in one figure.

Question 8.

- Combine the two noise reduction methods autocorrelation and Fourier-transform. Check it by computing the Fourier-transform of the autocorrelation of the measurement sb and show your result.

- Did you get a higher SNR-value compare to only doing Fourier-transform?

---

2. Detection using cross-correlation (matched filter technique).

Cross-correlation can be used to detect a "known signal" in a measurement. The known signal is often shorter in length (a transient signal) compare to the measurement. The result of the detection should be: the known signal is present in the measurement and at which position.

The detection can be implemented by a technique called "matched filter", where the matched filter is a folded version of the known signal, and the computation is done by convolution (=filtering). It can be interpreted as a "local match" of the known signal and the measurement for each position.

2.1 A simple example.

Create a measurement as a square signal, the known signal as one period of the measurement, and do the detection using the matched filter technique.

```
%Generate the measurement and the known signal.
%The measurement is a square signal of 400 samples.
%The known signal is one period of the measurement, 20 sample in length.
transient=ones(1,20);
transient(11:20)=-ones(1,10);
square=repmat(transient,1,20);
%Show signals
exptr=[transient, zeros(1,400-length(transient))];
figure(8);suplot(3,1,1);stem(0:399,exptr);
figure(8);subplot(3,1,2);stem(0:399,square);

%Do the detection by the matched filter technique.
mfilter=fliplr(transient); %folded transient, mfilter(-n)
y_square=conv(square,mfilter); %crosscorrelation by convolution
figure(8);subplot(3,1,3);stem(0:399,y_square(length(mfilter):length(y_square)));
```

Question 9.

Explain the position of the transient when the detection takes its highest and its lowest value.

2.2 Electrocardiograph (ecg) signal.

The known signal (the transient) to detect is taken from the ecg-measurement at a special position and of a certain length.

```
%Measurement
load ecg1; %ecg-signal, column vector
```

ecg1=ecg1'; %row vector
%Transient is taken from measurement at position 1270 and 146 samples long
tr_ecg=ecg1(1270:1415); %transient
%Design of the matched filter
mfilt_ecg=fliplr(tr_ecg); %folded transient; length=M
%Generate a measurement 'pos_ecg' with the transient at position 343
pos_ecg=zeros(1,1000);
pos_ecg(343:343+length(tr_ecg)-1)=tr_ecg;

%Detect the transient by matched filtering
y_ecg1=conv(pos_ecg,mfilt_ecg); %crosscorr by convolution
[d1,pos1]=find(y_ecg1==max(y_ecg1)); %search for maximum
%Compute the position of the transient
det_pos=pos1-(length(mfilt_ecg)-1); %add –(M-1)

%Show results
det_pos %detected position, should be 343
exptr_ecg=[tr_ecg, zeros(1,1000-length(tr_ecg))]; %expanded transient
figure(9);
subplot(3,1,1);plot(0:length(exptr_ecg)-1,exptr_ecg); %transient
subplot(3,1,2);plot(0:length(pos_ecg)-1,pos_ecg); %measurement
%crosscorrelation
subplot(3,1,3);plot(0:length(pos_ecg)-1,y_ecg1(length(mfilt_ecg):length(y_ecg1)));

Question 10.
In which signal did you analyse if the transient was present?
Change the position of the transient in the measurement and do the detection once more.

Now detect the transient in the ecg-measurement.
y_ecg2=conv(ecg1,mfilt_ecg); %crosscorr by convolution
%Show results
figure(10);
subplot(3,1,1);plot(0:length(ecg1)-1,ecg1); %ecg-measurement
%crosscorrelation
subplot(3,1,2);plot(0:length(ecg1)-1,y_ecg2(length(mfilt_ecg):length(y_ecg2)));

Add random noise to the ecg-measurement and do the detection of the transient.
%add random noise of low amplitude, $N\left(0,\sqrt{0.01}\right)$
rn3=sqrt(0.01)*randn(1,length(ecg1));
ecg_rn3=ecg1+rn3; %noisy ecg-signal
y_rn3=conv(ecg_rn3,mfilt_ecg); %crosscorr by convolution
%Show results
figure(11);
subplot(4,1,1);plot(0:length(ecg1)-1,ecg1); %ecg-signal

subplot(4,1,2);plot(0:length(ecg_rn3)-1,ecg_rn3); %noisy ecg-signal
%crosscorrelation noisy ecg-signal
subplot(4,1,3);plot(0:length(ecg_rn3)-1,y_rn3(length(mfilt_ecg):length(y_rn3)));

%crosscorrelation no noise
subplot(4,1,4);plot(0:length(ecg_rn3)-1,y_ecg2(length(mfilt_ecg):length(y_ecg2)));

Question 11.
Increase the amplitude of the noise by changing its variance in step of 0.01 and do the detection.
Note the variance of the noise when the detection goes wrong.

More questions:
Capture a signal via a microphone.
Listen to the original signal and a filtered version of it.

Useful commands:
%Set up the recording and check it.
%Sampling frequency=22050, number of bits=8, one channel
r=audiorecorder(22050,8,1); %set up the recording
record(r,1); %record in 1 second; **talk into the microphone now!**
play(r); %listen to the record sample

voice_sample=getaudiodata(r); %save the record sample
sound(voice_sample,22050); %listen to the saved sample
figure(27);stem(voice_sample); %show the voice signal

y_voice=conv(voice_sample,g); %do filtering by filter g
sound(y_voice,22050); %listen to the filtered version

Capture an image via a web-camera.
Do filtering of the captured image and do cross-correlation (matched filter) of subimages in the original image.

Useful commands :
%Set up image capturing and test it.
vid=videoinput('winvideo',1); %set up for image capturing
preview(vid); %test the set up
closepreview(vid); %close the capturing

I=getsnapshot(vid); %capture an rgb-image
face=double(I(:,:,1)); %make to gray-scale

%Show the captured image
figure(1);
imagesc(face);colormap(gray); truesize;

y_face=filter2(g',filter2(g,face)); filter the image by the a 2D filter g*g'

impixelinfo; %by the curser find the midpoint (c,r) of the subimage

%Compute a normalized scalar product (=one value in the normalized cross-correlation)
%between two subimages of equal size
Lefteye=face(65:85,90:110); %subimag1 with midpoint (c,r)=(100,75)
Righteye=face(65:85,140:160); %subimage2 with midpointn (c,r)=(150,75)

L1=sqrt(sum(sum(Lefteye.*Lefteye))); %length of subimage1
L2=sqrt(sum(sum(Righteye.*Righteye))); %length of subimage2

SProd=sum(sum(Lefteye.*Righteye)); %scalar product of two subimages
SC=SProd/(L1*L2); %normalized scalar product; [0, 1]

C Normalfördelningstabell

$$F(x)=\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-\frac{t^2}{2}}\,dt$$

x	0	1	2	3	4	5	6	7	8	9
0,0	0,5000	5040	5080	5120	5160	5199	5239	5279	5319	5359
0,1	5398	5438	5478	5517	5557	5596	5636	5675	5714	5753
0,2	5793	5832	5871	5910	5948	5997	6026	6064	6103	6141
0,3	6179	6217	6255	6293	6331	6368	6406	6443	6480	6517
0,4	6554	6591	6628	6664	6700	6736	6772	6808	6844	6879
0,5	6915	6950	6985	7019	7054	7088	7123	7157	7190	7224
0,6	7257	7291	7324	7357	7389	7422	7454	7486	7517	7549
0,7	7580	7611	7642	7673	7704	7734	7764	7794	7823	7852
0,8	7881	7910	77939	7967	7995	8023	8051	8078	8106	8133
0,9	8159	8186	8212	8238	8264	8289	8315	8340	8365	8389
1,0	8413	8438	8461	8485	8508	8531	8554	8577	8599	8621
1,1	8643	8665	8686	8708	8729	8749	8770	8790	8810	8830
1,2	8849	8869	8888	8907	8925	8944	8962	8980	8997	9015
1,3	9032	9049	9066	9082	9099	9115	9131	9147	9162	9177
1,4	9192	9207	9222	9236	9251	9265	9279	9292	9306	9319
1,5	9332	9345	9357	9370	9382	9394	9406	9418	9429	9441
1,6	9452	9463	9474	9484	9495	9505	9515	9525	9535	9545
1,7	9554	9564	9573	9582	9591	9599	9608	9616	9625	9633
1,8	9641	9649	9656	9664	9671	9678	9686	9693	9699	9706
1,9	9713	9719	9726	9732	9738	9744	9750	9756	9761	9767
2,0	9772	9778	9783	9788	9793	9798	9803	9808	9812	9817
2,1	9821	9826	9830	9834	9838	9842	9846	9850	9854	9857
2,2	9861	9864	9868	9871	9875	9878	9881	9884	9887	9890
2,3	9893	9896	9898	9901	9904	9906	9909	9911	9913	9916
2,4	9918	9920	9922	9925	9927	9929	9931	9932	9934	9936
2,5	9938	9940	9941	9943	9945	9946	9948	9949	9951	9952
2,6	9953	9955	9956	9957	9959	9960	9961	9962	9963	9964
2,7	9965	9966	9967	9968	9969	9970	9971	9972	9973	9974
2,8	9974	9975	9976	9977	9977	9978	9979	9979	9980	9981
2,9	9981	9982	9983	9983	9984	9984	9985	9985	9986	9986
3,0	9987	9987	9987	9988	9988	9989	9989	9989	9990	9990
3,1	9990	9991	9991	9991	9992	9992	9992	9992	9993	9993
3,2	9993	9993	9994	9994	9994	9994	9994	9995	9995	9995
3,3	9995	9995	9995	9996	9996	9996	9996	9996	9996	9997
3,4	9997	9997	9997	9997	9997	9997	9997	9997	9997	9998