Auto-regressive Random Process

M.Tognolini HEIG-VD 2022 MA-StatDig rev. 1.2 (21.03.2022)

Estimation of autocorrelation of random process (CompEx 3.2)

In many applications it is necessary to be able to efficently estimate the autocorrelation sequence of a random process from a finite number of samples e.g. x(n) for n = 0 ..N-1.

The autocorrelation may be estimated using the sample autocorrelation:

$$\hat{r}_x(k) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)x^*(n-k) \qquad (1)$$

For this estimation we assume the samples outside the interval 0 ..N-1 are equal to 0. Since this is assumed we could write the sample autocorrelation as a convolution :

$$\hat{r}_x(k) = \frac{1}{N}[x(k) * x^*(-k)] \qquad (2)$$

The discrete-time Fourier transform of $\hat{r}_x(k)$ is the magnitude squared of the discrete -time Fourier tansform of x(n) scaled by 1/N:

$$\sum_{k=-N+1}^{N-1} \hat{r}_x(k)e^{-jk\omega} = \frac{1}{N}|X(e^{-jk\omega})|^2 \quad (3)$$

where $X(e^{-jk\omega}) = \sum_{n=0}^{N-1} x(n)e^{-jn\omega}$

First experiment : test with a simple signal x(n)

with the signal x(n) = 1 (deterministic) for n= 1 ..N-1 with N = 8 compute the sample autocorrelation using (1) write a function [rxhat,k]= SampleAutocorr(xn)

1. Compute the sample autocorrelation by the convolution 1/N (x(k)*x(-k)) and show that give the same result.
2. Verify the property given by (3) , for that compute the DFT of the sample autocorrélation rxhat and compare to the squared of the module DFT of x(n)
3. Find the sample autocorrelation using the FFT of x(n) with N = 8 or 16 , or 32 samples. Comare the results.
4. Try all the precedent steps with a gaussian white process: x(n) = randn(1,N)

```
N  = 8;

xn = ones(1,N)
```

```
xn = 1×8
     1     1     1     1     1     1     1     1
```

% sample autocorrelation using (1) with a foor loop

[rxhat, k] = SampleAutocorr(xn);

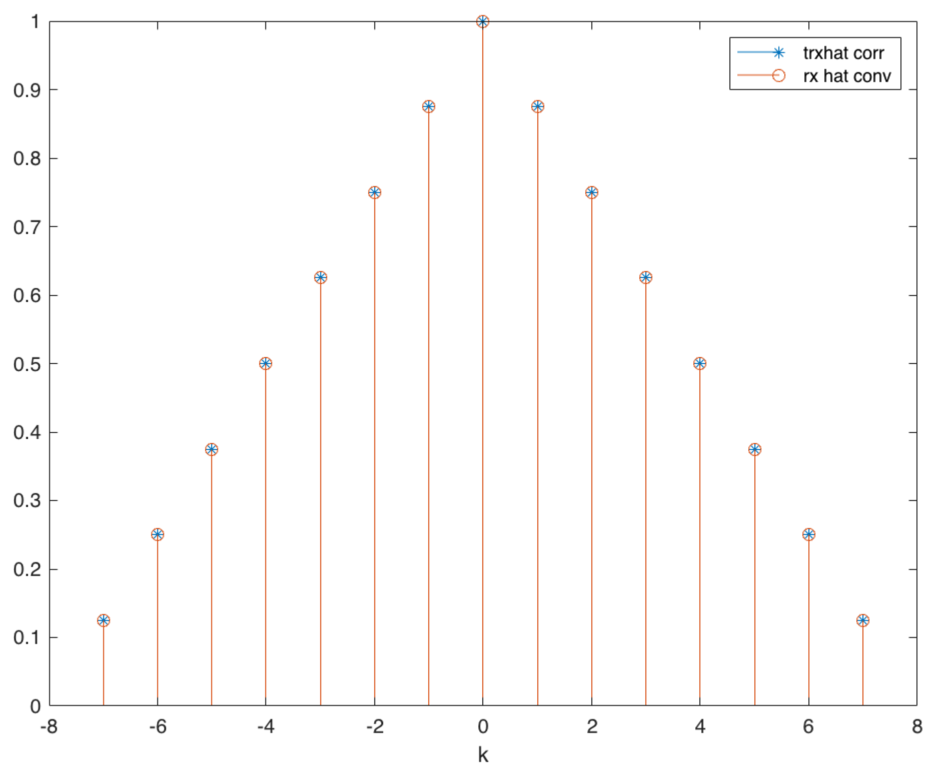
[k ; rxhat]

```
ans = 2×15
   -7.0000   -6.0000   -5.0000   -4.0000   -3.0000   -2.0000   -1.0000         0 ···
    0.1250    0.2500    0.3750    0.5000    0.6250    0.7500    0.8750    1.0000
```

Compute with the function conv() and the function fliplr() the autocorrelation of xn:

[k;rxhatconv]

```
ans = 2×15
   -7.0000   -6.0000   -5.0000   -4.0000   -3.0000   -2.0000   -1.0000         0 ···
    0.1250    0.2500    0.3750    0.5000    0.6250    0.7500    0.8750    1.0000
```



Compute the DFT of the signal x(n) (N samples) with NFFT sample of the DFT (try 8, 16, 32), compute the DFT and compute the square of the magnitude:

For ex for NFFT = 16 we have:

```
NFFT = 16

     w = -pi: 2*pi/NFFT : pi-2*pi/NFFT;
     f = w/2/pi

f = 1×16
   -0.5000   -0.4375   -0.3750   -0.3125   -0.2500   -0.1875   -0.1250   -0.0625 ···

     Xsq_fft = (abs(fftshift(fft(xn,NFFT)))).^2 /N
     [f;Xsq_fft]


Xsq_fft = 1×16
         0    0.1299         0    0.1808         0    0.4050         0    3.2843 ···
```

Compute the DFT of the sample autocorrélation sequence Px and verify that is real:

For FFT use it is important to have rxhat as a entire periode of a periodic sequence of length 2N, rxhat must be symetric and has 2N-1 samples so to ensure simmetricity and have a right computation of the DFT we need to add a 0 sample at the beginning of rxhat previously computed :

$\widehat{r'}_x(k) = [\mathbf{0}, \widehat{r}_x(-N/2+1, \cdots, \widehat{r}_x(0), \cdots, \widehat{r}_x(N/2-1))$ this vector now has 2N samples and has the same energy than $\widehat{r}_x(k)$

```
     rxhat_p = [zeros(1,NFFT/2 - N+1),rxhat,zeros(1,NFFT/2 - N)]


rxhat_p = 1×16
         0    0.1250    0.2500    0.3750    0.5000    0.6250    0.7500    0.8750 ···

     Pxfft =fftshift( fft(fftshift(rxhat_p),NFFT))

Pxfft = 1×16
         0    0.1299         0    0.1808         0    0.4050         0    3.2843 ···
```

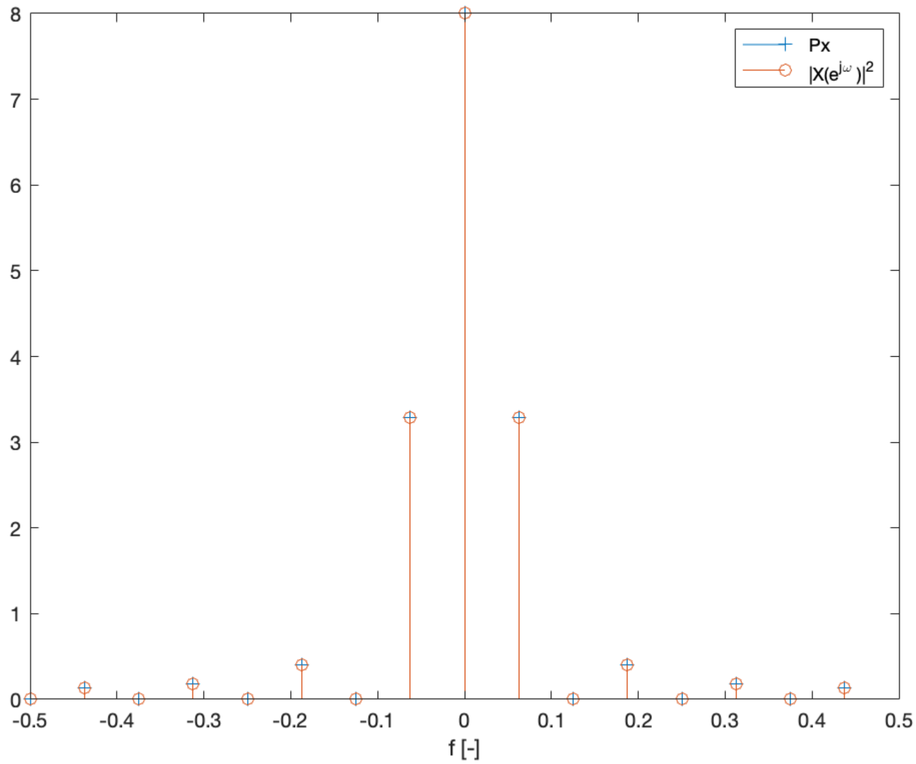
Must be real values and positives ! So take real part after verification.

```
Pxfftr = 1×16
         0    0.1299         0    0.1808         0    0.4050         0    3.2843 ···
```

[f;Pxfft] = ...

```
ans = 2×16
   -0.5000   -0.4375   -0.3750   -0.3125   -0.2500   -0.1875   -0.1250   -0.0625 ···
         0    0.1299         0    0.1808         0    0.4050         0    3.2843
```

Plot the power spectrum Px computed with x(n) directly or with the sample autocorrelation sequence:



Compute the sample autocorrélation by the inverse DFT of the power specttrum:

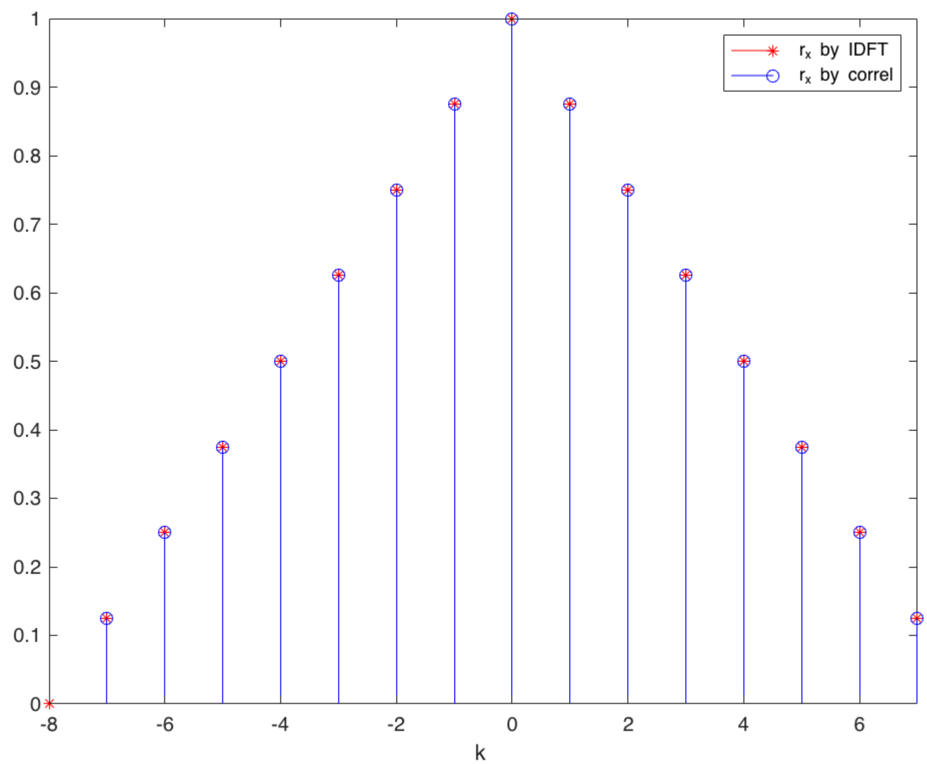
We could compute the sample autocorrelation estimation by the inverse of the quared of the DFT of xn:

To obtain the same autocorrelation verctor you need to remove the useless zeros at the beginning and the ed of the vector they was added when the power spectrum was computed.

```
rxhatfft = fftshift(ifft(fftshift(Pxfft)))
```

```
rxhatfft = 1×16
         0    0.1250    0.2500    0.3750    0.5000    0.6250    0.7500    0.8750 ···
```

Plot the sample autocorrelation directly computed from xn and from the power spectrum Px computed with the square of the DFT magnitude of x(n). Try with NFFT = 8 , 16 or 32.



1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-8
-6
-4
-2
0
2
4
6
k
r_x by IDFT
r_x by correl