REPORT

Zajęcia: Analog and digital electronic circuits
Teacher: prof. dr hab. Vasyl Martsenyuk

Lab 3
08.03.2024
Topic: "Random signals"
Variant 2

Wiktor Merta
Informatyka II stopień,
stacjonarne,
1 semestr,
Gr.2

1. Problem statement: The objective is to get an idea on
• the probability density function (PDF)
• what is a sample function of a random process
• first / second order ensemble averages (moments)
• the concept of stationarity and ergodicity
• the concept of temporal average vs. ensemble average
• the autocorrelation / cross correlation as as a higher order ensemble and temporal averages

2. Input data:
f = 400
A = 400.25
B = 399.75
N = 3000

3. Commands used (or GUI):

a) source code

Generating ensemble of random signals

```
# create random process based on normal distribution
A = 400.25
B = 399.75
Ns = 3000  # number of samples to set up an ensemble
Nt = 400  # number of time steps to set up 'ensemble over time'-characteristics
np.random.seed(1)

s = np.arange(Ns)  # ensemble index (s to indicate sample function)
t = np.arange(Nt)  # time index

loc, scale = 0, 1  # mu, sigma
x = np.random.normal(loc=loc, scale=scale, size=(Ns, Nt))

# the plots nicely show the concept of
# temporal average (left column) vs. ensemble average (right column)
# we make sure to understand it with the
```

cos-like patterns either over samples or over time instances

x = B * x + A * np.tile(np.cos(2 * np.pi/Nt * np.arange(Nt)), (Ns, 1))

Plotting linear mean, quadratic mean and variance

```
fig, axs = plt.subplots(4, 2, figsize=(9, 13))
# plot signals
for i in range(4):
    axs[0, 0].plot(x[:, i], s, label='time index '+str(i))
    axs[0, 1].plot(t, x[i, :], label='ensemble index '+str(i))
# plot means
axs[1, 0].plot(np.mean(x, axis=1), s)
axs[1, 1].plot(t, np.mean(x, axis=0))
axs[1, 0].plot([loc, loc], [0, Ns])
axs[1, 1].plot([0, Nt], [loc, loc])
# plot variance
axs[2, 0].plot(np.var(x, axis=1), s)
axs[2, 1].plot(t, np.var(x, axis=0))
axs[2, 0].plot([scale**2, scale**2], [0, Ns])
axs[2, 1].plot([0, Nt], [scale**2, scale**2])
# plot quadratic mean
axs[3, 0].plot(np.mean(x**2, axis=1), s)
axs[3, 1].plot(t, np.mean(x**2, axis=0))
axs[3, 0].plot([loc**2+scale**2, loc**2+scale**2], [0, Ns])
axs[3, 1].plot([0, Nt], [loc**2+scale**2, loc**2+scale**2])
# labeling
axs[3, 1].set_xlabel('time index')
for i in range(4):
    #axs[i,1].set_xlabel('time index')
    axs[i, 0].set_ylabel('ensemble index')
    for j in range(2):
        axs[i, j].grid(True)
axs[0, 0].set_title(r'temporal average for fixed ensemble index')
axs[0, 1].set_title(r'ensemble average for fixed time instance')
for i in range(2):
    axs[0, i].legend(loc='upper left')
```

```
    axs[1, i].set_title(r'linear mean $E\{x\} = \mu$')
    axs[2, i].set_title(r'variance  $E\{(x -E\{x\} )^2\} = \sigma^2$')
    axs[3, i].set_title(r'quadratic mean $E\{x^2\} = \mu^2+\sigma^2$')
```

Defining autocorrelation estimator function

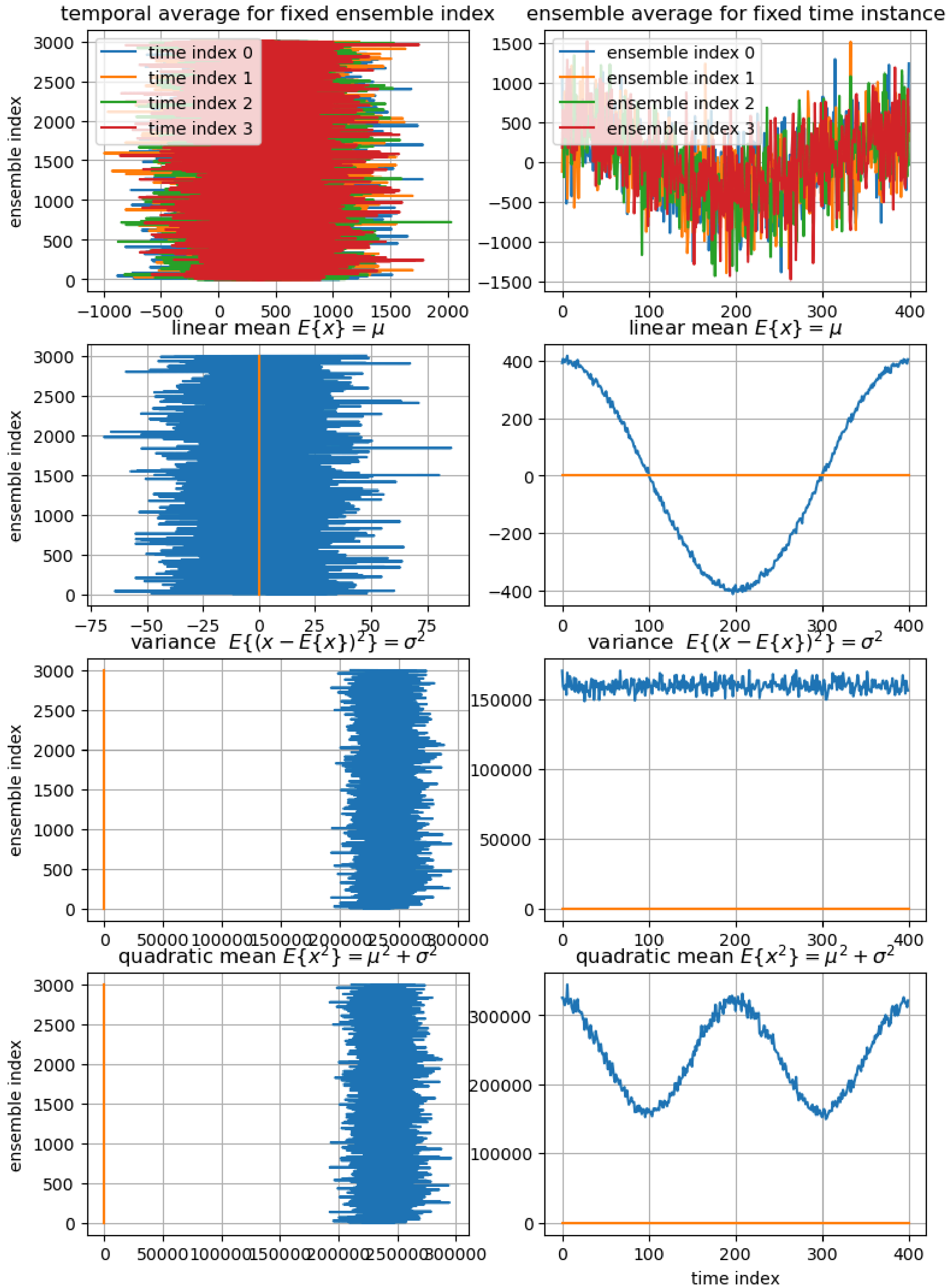
```
def my_xcorr(x, y):
    N, M = len(x), len(y)
    kappa = np.arange(N+M-1) - (M-1)
    ccf = signal.correlate(x, y, mode='full', method='auto')
    return kappa, ccf
```

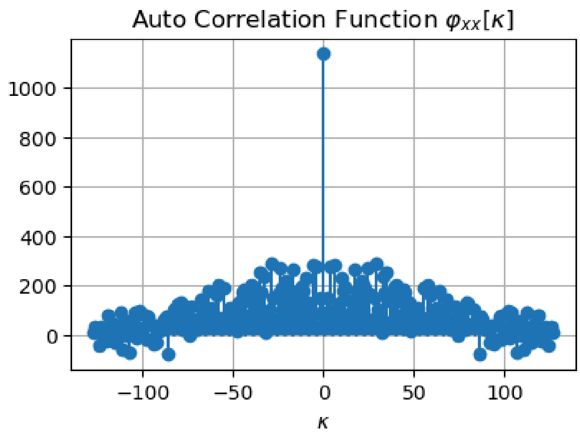
Plotting ACF

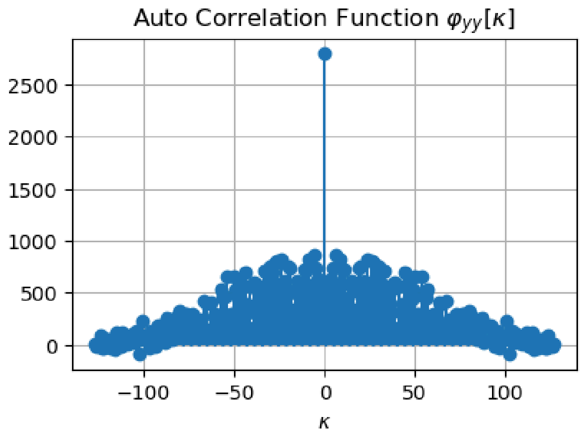
```
plt.figure(figsize=(10, 3))
plt.subplot(1, 2, 1)
kappa, ccf = my_xcorr(x[0, :], x[0, :])
plt.stem(kappa, ccf, basefmt='C0:', use_line_collection=True)
plt.xlabel(r'$\kappa$')
plt.title(r'Auto Correlation Function $\varphi_{xx}[\kappa]$')
plt.grid(True)
plt.subplot(1, 2, 2)
kappa, ccf = my_xcorr(y[0, :], y[0, :])
plt.stem(kappa, ccf, basefmt='C0:', use_line_collection=True)
plt.xlabel(r'$\kappa$')
plt.title(r'Auto Correlation Function $\varphi_{yy}[\kappa]$')
plt.grid(True)
```

https://github.com/wm64167/AADEC

4. Outcomes:


temporal average for fixed ensemble index
ensemble average for fixed time instance
time index 0
time index 1
time index 2
time index 3
ensemble index 0
ensemble index 1
ensemble index 2
ensemble index 3
ensemble index
linear mean $E\{x\} = \mu$
variance $E\{(x - E\{x\})^2\} = \sigma^2$
quadratic mean $E\{x^2\} = \mu^2 + \sigma^2$
time index





5. Conclusions: This lab explored random signals by generating ensembles of them. Each signal had a common form but included random noise. We then used these ensembles to estimate various statistical properties of the original signal. We looked at average values, how spread out the data was, and even how the signal itself correlated with itself over time. For the reasons given, we conclude that ensemble averages can be very useful in characterizing random processes and higher-order moments like variance and autocorrelation are indispensable for a comprehensive analysis.