Telecommunications Engineering
Universidad Carlos III de Madrid
Statistics

Assignment 4: Random Vectors & Stochastic Processes

Group	Students	Signatures
	• Alonso Herreros Copete	Alonso

IMPORTANT: *The teachers of this course apply a 'zero tolerance' policy regarding academic dishonesty. Students that sign up this document agree to deliver an original work. The breach of this commitment will result in academic punishment.*

<u>Observations</u>:

Solve the exercises in the **Assignment3.pdf** file. **Note:** It is advisable to consult the manual for basic operation of MATLAB / Octave available on the website of the course.

Available on GitHub: https://github.com/alonso-herreros/uni-stat-lab4

1. Random Vectors

1.1. Exercise 1.

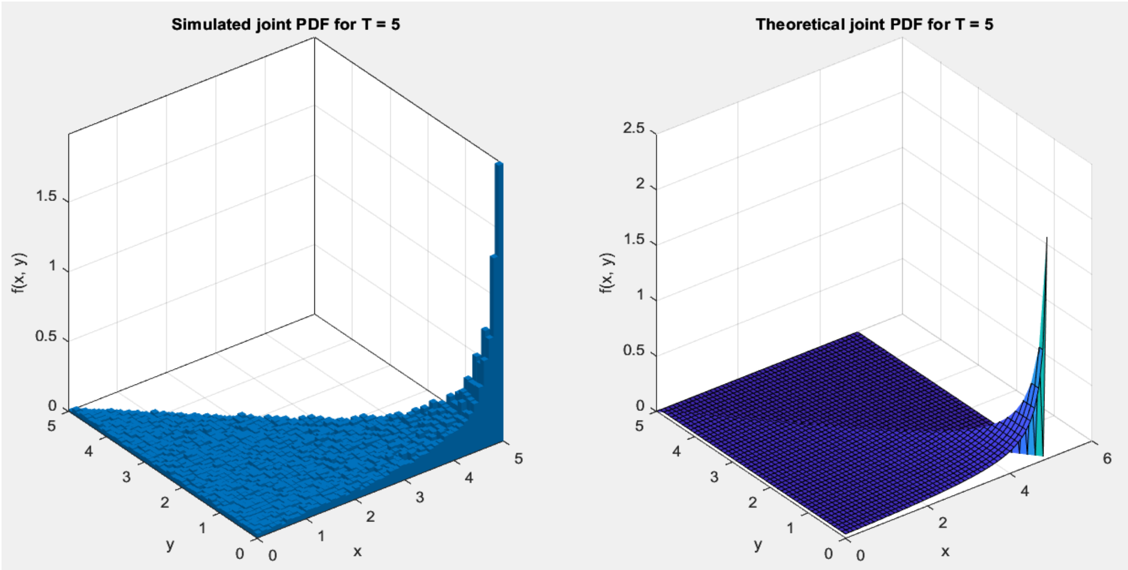
Choose a point x randomly in the interval $(0, T)$ and a second point also randomly in the interval (X, T). Define the r.v. X as "the position of the first point" and the r.v. Y as "the distance of the second point respect to the first".

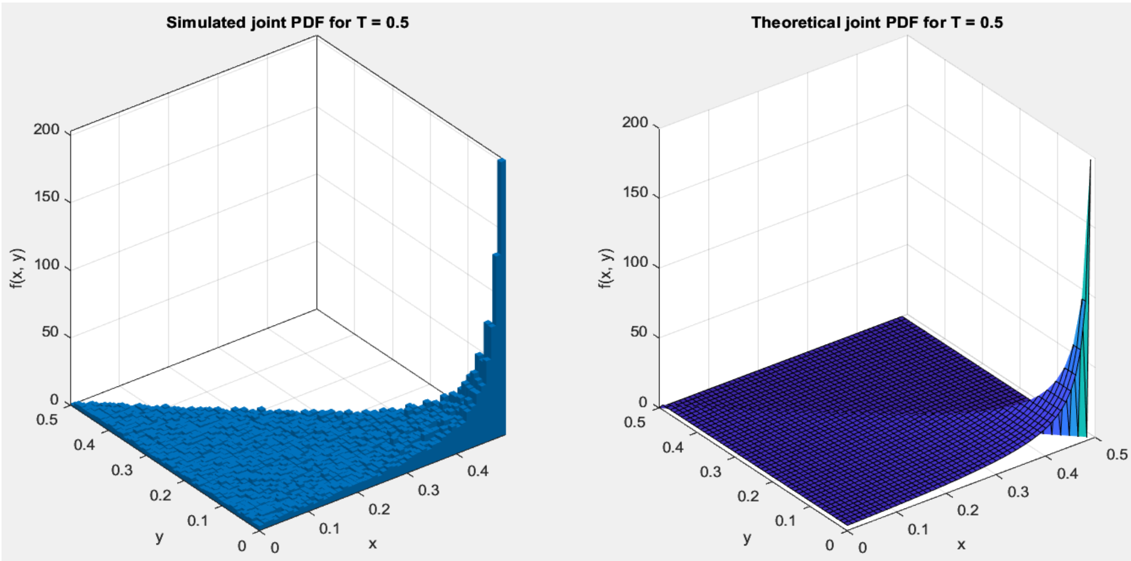
a) Determine, theoretically, the joint density function of (X, Y).

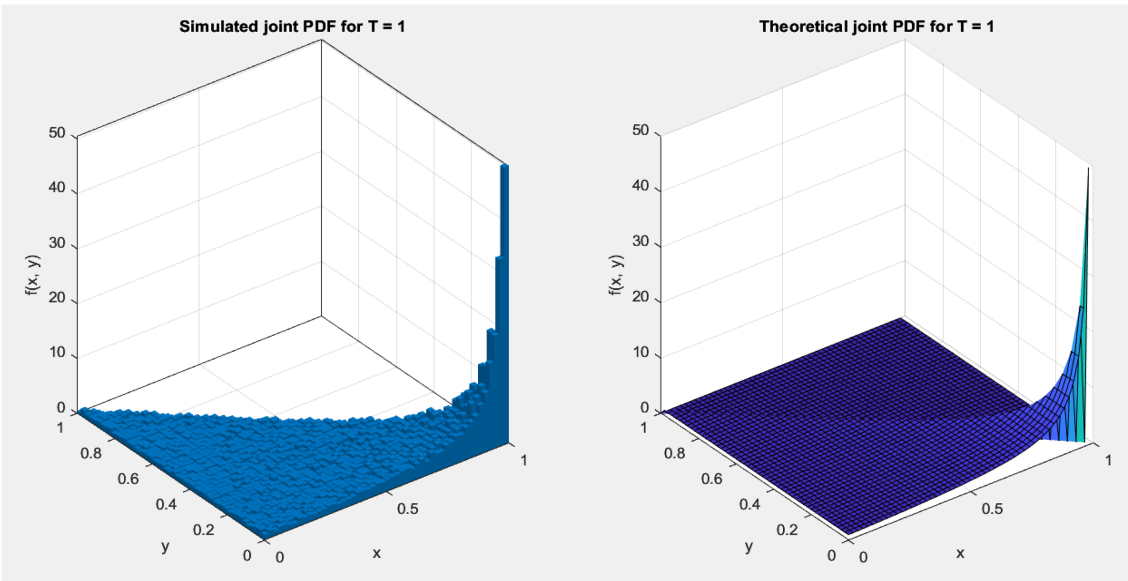
Let $X \sim U(0,T)$ and $P \sim U(x,T)$
Now, let $Y = |P - X| = P - X$
For $X = x$, we have $(Y \mid X = x) \sim U(x,T) - x = U(0, T - x)$
We may conceptually use
$f(x,y) = P(X = x, Y = y) = P(X = x \cap Y = y) = P(Y = y \cap X = x) =$
$= P(Y = y \mid X = x) \cdot P(X = x)$
Since these are uniform distributions, we may use the formula $f(x) = \frac{1}{T}$, and the analogous for Y, while setting the limits x and y.

$$f(x,y) = \begin{cases} \frac{1}{T\cdot(T-x)} & if\ 0 \le x \le T\ and\ 0 \le y \le T - x \\ 0 & otherwise \end{cases}$$

This formula was tested by plotting it, along with the graph generated from a simulation of this process, and visually comparing both results, with different values of T.







b) Compute by simulation using MATLAB/Octave, P (Y − X < 0) for T = 5.

The simulation is quite simple. First, a vector `x` is created with random values from the variable X, according to a uniform distribution between 0 and T, using the function `unifrnd`. Then, the vector `x` is created by generating a random value between 0 and T-x for each entry in the vector `x`, using `unifrnd` and `arrayfun`. The intermediate step of generating a random point between x and T was not necessary, and since this is a very simple process, it was skipped.

```
N = 100000; % number of simulations
T = 5;

x = unifrnd(0, T, N, 1);
y = arrayfun(@(x) unifrnd(0, T-x), x);
disp("Simulated one case at a time.")
disp("P(Y - X < 0) = " + sum(y-x < 0)/N);
```

```
Simulated one case at a time.
P(Y - X < 0) = 0.69196
```

1.2. EXERCISE 2.

Let X and Y be independent r.v.'s with X continuous and Y discrete, given by X ~ U (−1, 1) and

$$Y \sim \begin{pmatrix} 0 & 1 \\ \frac{1}{4} & \frac{3}{4} \end{pmatrix}$$

where the notation indicates that the r.v. Y takes value 0 with probability 1/4 and value 1 with probability 3/4. Considering $R = X + Y$:

a) Determine $f_R(r)$.

$$X \sim U(-1, 1) \Rightarrow f_X(x) = \begin{cases} \frac{1}{2} & if\ -1 \le x \le 1 \\ 0 & otherwise \end{cases}$$

$$f_Y(y) = \begin{cases} \frac{1}{4} & if\ y = 0 \\ \frac{3}{4} & if\ y = 1 \\ 0 & otherwise \end{cases}$$

We have $R = X + Y \Rightarrow r \in [-1, 2]$, so we may think of three ranges:

- $-1 \le r < 0 \Rightarrow Y = 0 \cap X = r \Rightarrow$
  $\Rightarrow f_R(r) = f_Y(0) \cdot f_X(r) = \frac{1}{4} \cdot \frac{1}{2} = \frac{1}{8}$
- $0 \le r < 1 \Rightarrow (Y = 0 \cap X = r) \cup (Y = 1 \cap X = r - 1) \Rightarrow$
  $\Rightarrow f_R(r) = f_Y(0) \cdot f_X(r) + f_Y(1) \cdot f_X(r - 1) = \frac{1}{4} \cdot \frac{1}{2} + \frac{3}{4} \cdot \frac{1}{2} = \frac{4}{8} = \frac{1}{2}$
- $1 \le r < 1 \Rightarrow Y = 1 \cap X = r - 1 \Rightarrow$
  $\Rightarrow f_R(r) = f_Y(1) \cdot f_X(r - 1) = \frac{3}{4} \cdot \frac{1}{2} = \frac{3}{8}$

With all of this:

$$f_R(r) = \begin{cases} \frac{1}{8} & if\ -1 \le r < 0 \\ \frac{1}{2} & if\ 0 \le r < 1 \\ \frac{3}{8} & if\ 1 \le r < 2 \\ 0 & otherwise \end{cases}$$

b) Calculate theoretically E [R] and V ar [R].

$$E[R] = \int_{-\infty}^{+\infty} r \cdot f_R(r)dr = \int_{-1}^{0} r \cdot f_R(r)dr + \int_{0}^{1} r \cdot f_R(r)dr + \int_{1}^{2} r \cdot f_R(r)dr =$$

$$= \int_{-1}^{0} r \cdot \frac{1}{8}dr + \int_{0}^{1} r \cdot \frac{1}{2}dr + \int_{1}^{2} r \cdot \frac{3}{8}dr = \frac{r^2}{16}\Big|_{r=-1}^{0} + \frac{r^2}{4}\Big|_{r=0}^{1} + \frac{3r^2}{16}\Big|_{r=1}^{2} =$$

$$= \frac{1}{16} \cdot \left(0 - (-1)^2\right) + \frac{1}{4} \cdot (1 - 0) + \frac{3}{16} \cdot \left(2^2 - 1\right) = \frac{-1+4+9}{16} = \frac{3}{4} = 0.75$$

$$Var[R] = E\left[R^2\right] - E[R]^2 = \int_{-\infty}^{+\infty} r^2 \cdot f_R(r)dr - \left(\frac{3}{4}\right)^2 =$$

$$= \int_{-1}^{0} r^2 \cdot \frac{1}{8}dr + \int_{0}^{1} r^2 \cdot \frac{1}{2}dr + \int_{1}^{2} r^2 \cdot \frac{3}{8}dr - \frac{9}{16} = \frac{r^3}{24}\Big|_{r=-1}^{0} + \frac{r^3}{6}\Big|_{r=0}^{1} + \frac{3r^3}{24}\Big|_{r=1}^{2} - \frac{9}{16} =$$

$$= \frac{1}{24} \cdot \left(0 - (-1)^3\right) + \frac{1}{6} \cdot (1 - 0) + \frac{3}{24} \cdot \left(2^3 - 1\right) - \frac{9}{16} = \frac{1}{24} + \frac{1}{6} + \frac{21}{24} - \frac{9}{16} =$$

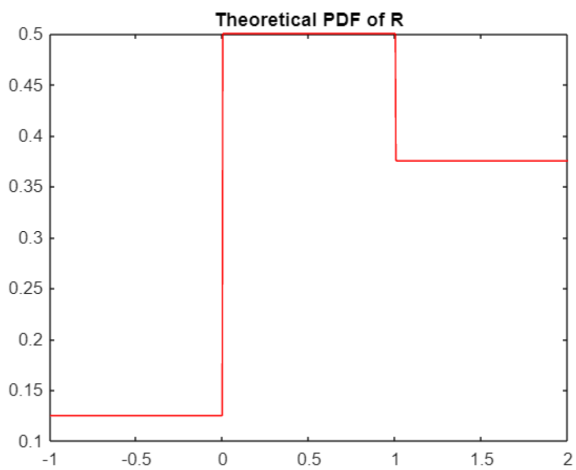
$$= \frac{2+8+42-27}{48} = \frac{25}{48} \approx 0.5208$$

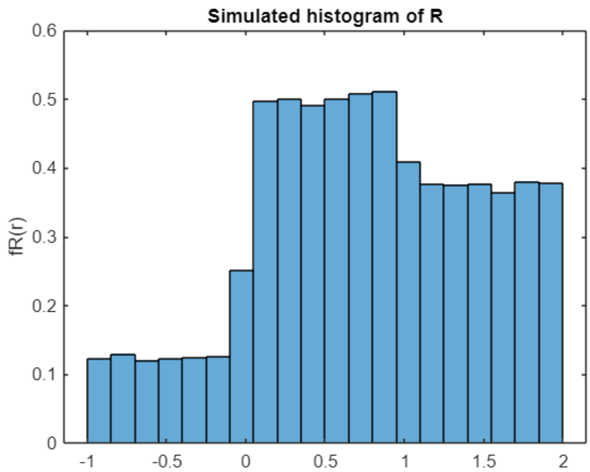
c) Check with MATLAB/Octave the results obtained in a) and b)

This was done directly when calculating the theoretical formulas, to check their validity. The code is quite simple, and self explanatory at this point.

For part a), the process was simulated and the theoretical function was plotted side by side with a PDF-normalized histogram of the simulation to check if the values matched.

```
% I will do a simulation to check the results of f(r)
N = 100000; % number of simulations
x = unifrnd(-1, 1, N, 1);
y = arrayfun(@(u) 0 + (u > 0.25), rand(N, 1));
r = x + y;
% Let's plot the histogram
figure(2);
histogram(r, 20, Normalization="pdf");
title("Simulated histogram of R");
xlabel("r");
ylabel("fR(r)");
% Let's plot the theoretical pdf
figure(3);
x = linspace(-1, 2, 1000);
y = arrayfun(@(x) (1/8)*(x >= -1 && x < 0) + (1/2)*(x >= 0 && x <= 1) + (3/8)*(x > 1 && x <= 2), x);
plot(x, y, "r");
title("Theoretical PDF of R");
```





The graph with the simulated data is quite consistent with the theoretical values.

The same simulation data that was created to check the result in part a) was used to check the result in part b).

```
disp("Theoretical E[R] = 0.75");
disp("Simulated   E[R] = " + mean(r) + " (should be close to 0.75)")

disp("Theoretical Var[R] = 25/48 = 0.52083");
disp("Simulated   Var[R] = " + var(r) + " (should be close to 0.52083)")
```

```
Theoretical E[R] = 0.75
Simulated   E[R] = 0.75035 (should be close to 0.75)

Theoretical Var[R] = 25/48 = 0.52083
Simulated   Var[R] = 0.52157 (should be close to 0.52083)
```

2. Stochastic Processes

2.1. Random Walk

A random walk is a discrete stochastic process $Y(n)$, given by the sum of n i.i.d. random variables, i.e.,

$$Y(n) = X_1 + \cdots + X_n = \sum_{i=1}^{n} X_i.$$

An example of random walk is such that $X_n \sim Ber\{-1, +1\}$ where X_n takes value $+1$ with probability p and value -1 with probability $1 - p$.

a) Determine analytically the mean and the variance of the process $Y(n)$. Are the mean and the variance of the stochastic process $Y(n)$ constant when p = 1/4?

$$E[X] = \sum_{i=-\infty}^{+\infty} x_i \cdot P\left(X = x_i\right) = (-1) \cdot P(X = -1) + (1) \cdot P(X = 1) = -1 \cdot (1 - p) + 1 \cdot p =$$

$$= p - 1 + p = 2p - 1$$

$$E[Y] = E\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} E\left[X_i\right] = \sum_{i=1}^{n} (2p - 1) = n \cdot (2p - 1)$$

$$Var[Y] = E\left[Y^2\right] - E[Y]^2 = E\left[\left(\sum_{i=1}^{n} X_i\right)^2\right] - (n \cdot (2p - 1))^2 =$$

$$= E\left[\sum_{i=1}^{n} X_i^2 + \sum_{i=1}^{n}\sum_{j\neq i}\left(X_i \cdot X_j\right)\right] - (n \cdot (2p - 1))^2 =$$

$$= \sum_{i=1}^{n} E\left[X_i^2\right] + \sum_{i=1}^{n}\sum_{j=1}^{n} E\left[X_i \cdot X_j\right] - \sum_{i=1}^{n}\sum_{j=i}^{i} E\left[X_i \cdot X_j\right] - (n \cdot (2p - 1))^2 =$$

$$= \sum_{i=1}^{n} 1 + \sum_{i=1}^{n}\sum_{j=1}^{n}\left(E\left[X_i\right] \cdot E\left[X_j\right]\right) - \sum_{i=1}^{n}\left(E\left[X_i\right] \cdot E\left[X_i\right]\right) - (n \cdot (2p - 1))^2 =$$

$$= n + n^2 \cdot (2p - 1)^2 - n \cdot (2p - 1)^2 - (n \cdot (2p - 1))^2 = n \cdot \left(1 - (2p - 1)^2\right)$$

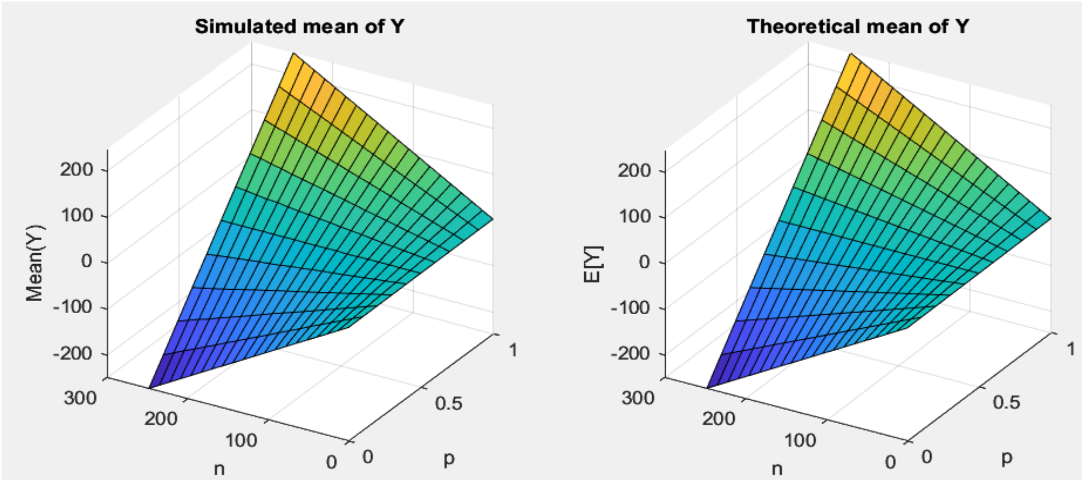
When $p = \frac{1}{4}$, we have the following expected value and variance:

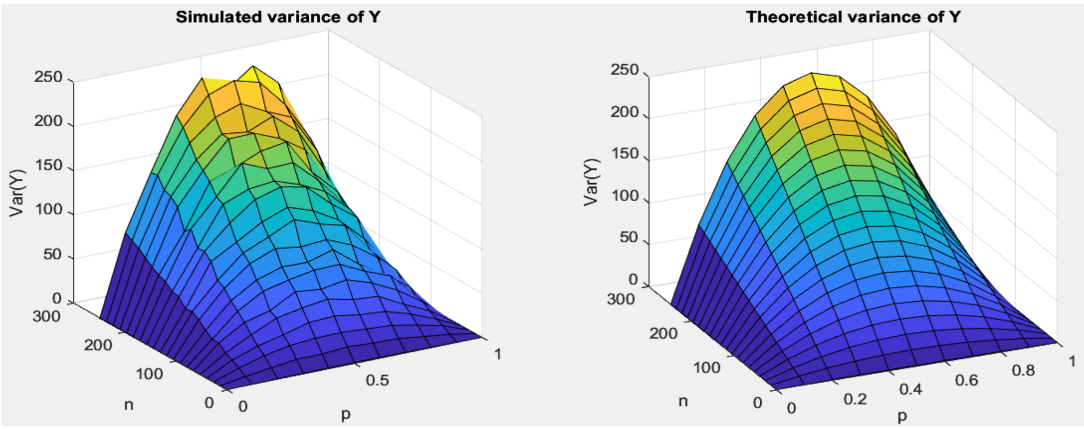
$$E[Y] = n \cdot \left(2 \cdot \frac{1}{4} - 1\right) = n \cdot \frac{-1}{2} = -\frac{n}{2}$$

$$Var[Y] = n \cdot \left(1 - \left(2\frac{1}{4} - 1\right)^2\right) = n \cdot \left(1 - \left(\frac{-1}{2}\right)^2\right) = n \cdot \left(1 - \frac{1}{4}\right) = \frac{3n}{4}$$

Therefore, for $p = \frac{1}{4}$, the variance and expected value are not constant, since they depend on n.

These theoretical formulas were plotted against simulated data to check their validity, using MATLAB.



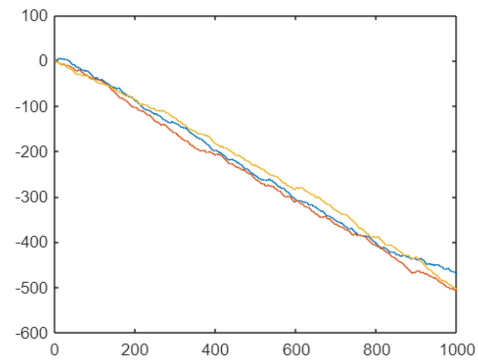


The plots show enough similarity to conclude that the theoretical formulas are accurate. The code wasn't required for this report, but it can be found with the rest of the exercises in the GitHub repository linked at the top of this document.

b) Generate with MATLAB/Octave three realizations (overlapping the graphs) of a Bernoulli process with p = 1/4 and n = 1000. Is the obtained result consistent with a)?

The process was simulated using a very simple piece of MATLAB code:

```
n = 1000; p = 1/4;
y = cumsum( 2*(rand(3, 1000) < p) - 1 , 2);
plot(1:n, y(1,:), 1:n, y(2,:), 1:n, y(3,:));
```



The resulting plot was also consistent with the theoretical expected value.

2.2. Harmonic Processes

Let X and Y be two independent r.v.'s normally distributed with parameters $\mu_X = \mu_Y = 0$ and $\sigma_X^2 = \sigma_Y^2 = 1$. Define the harmonic process $Z(t)$ as:

$$Z(t) = X\,cos(2\pi t) + Y\,sin(2\pi t)$$

Generate with MATLAB/Octave three realizations (overlapping the graphs) of the process $Z(t)$ with $t = 0 : 0.01 : 2\pi$

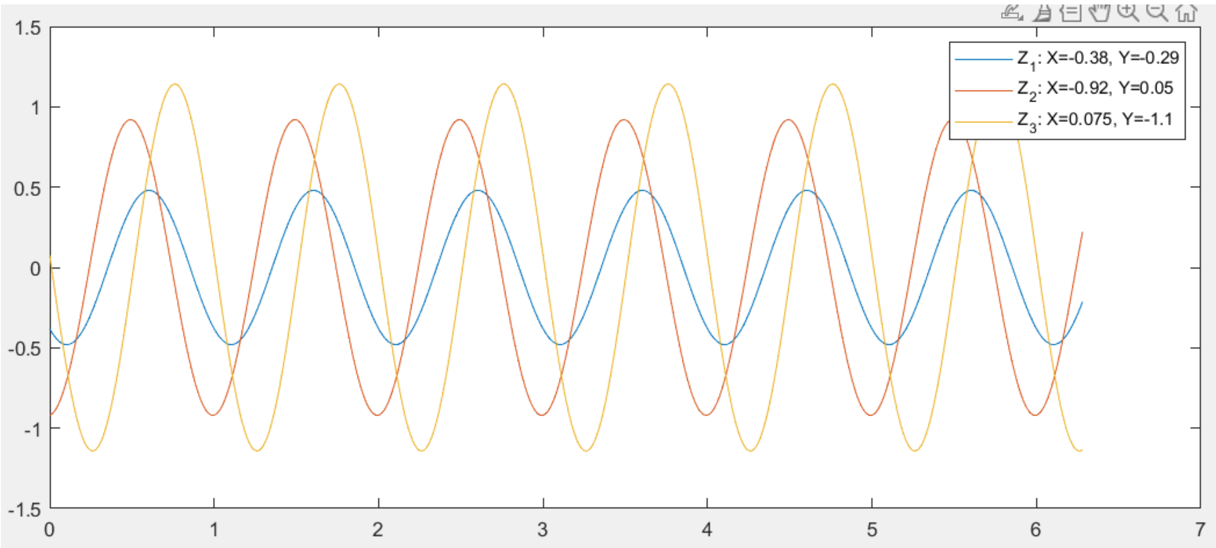
The first time tackling this problem, X and Y were assigned random values for each point in time, leading to a plot full of lines everywhere, from which no data could be obtained. This first plot, which made no sense, was an accurate representation of the process given to the program, but it was a wrong interpretation of the question.

The following simple MATLAB code was used to generate a plot with the three realizations of the stochastic process.

```
t = 0:0.01:2*pi;
X = normrnd(0,1,3,1);
Y = normrnd(0,1,3,1);
Z = X*cos(2*pi*t) + Y*sin(2*pi*t);

plot(t,Z(1,:), t,Z(2,:), t,Z(3,:));

legend(...
    "Z_1: X=" + num2str(X(1), 2) + ", Y=" + num2str(Y(1), 2), ...
    "Z_2: X=" + num2str(X(2), 2) + ", Y=" + num2str(Y(2), 2), ...
    "Z_3: X=" + num2str(X(3), 2) + ", Y=" + num2str(Y(3), 2));
```



2.3. Harmonic process with white noise

Consider the stochastic process of sinusoid type with random phase and white noise defined by:

$$Y(t) = a\,sin(\varpi + u) + W(t)$$

where $a = 1$, $\varpi = \frac{\pi}{5}$, $u \sim U(-\pi, \pi)$ and $W(t) \sim N(0, 1)$ independent from u.

a) Determine analytically the mean of the stochastic process Y (t). Is the mean of the process Y (t) constant?

$$Mean(Y(t)) = E[Y] = E[a\,sin(\varpi + u) + W(t)] = E[a\,sin(\varpi + u)] + E[W(t)] =$$
$$= 0 + 0 = 0$$

Yes, the mean is constant and equal to 0, independently of t, since the mean of a sine wave is 0, and so is the mean of $W(t)$.

b) Generate with MATLAB/Octave three realizations (overlapping the graphs) of the process Y (t) with t = 1 : 150. Is the obtained result consistent with a)?

Using the given parameters, the MATLAB code used was the following:

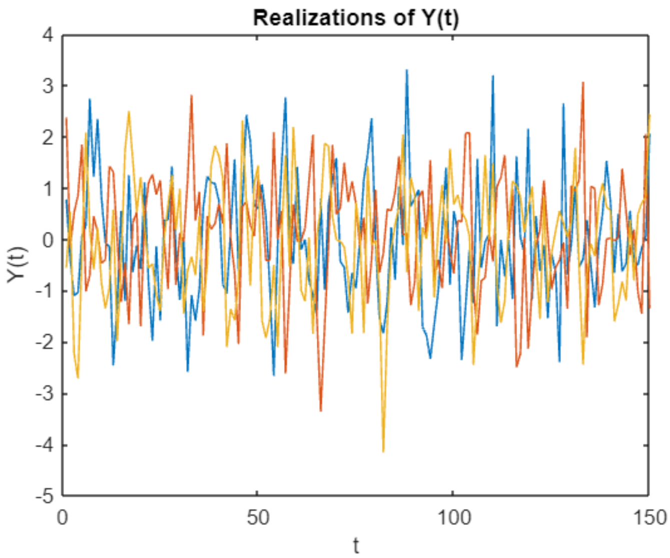
```
t = 1:150;
u = unifrnd(-pi, pi, 3, 1);
[t, u] = meshgrid(t,u);

Y = 1*sin(pi/5*t + u) + normrnd(0, 1, 3, 150);

plot(t(1,:), Y(1,:), t(2,:), Y(2,:), t(3,:), Y(3,:));
title("Realizations of Y(t)");
xlabel("t");
ylabel("Y(t)");

disp("Mean of each simulation: " + mean(Y(1,:)) +", "+ mean(Y(2,:)) +", "+  mean(Y(3,:)));
```

And the output was as shown here:




```
Mean of each simulation: -0.022358, 0.11627, -0.03597
```

The simulated means are close to 0, and the graph also shows that the signals oscillate around this value, so the theoretical and simulated values are consistent with each other.

In the previous graph the lines were very erratic, and it was difficult to interpret what the plot shows. By lowering the variance of the noise-inducing term $W(t)$ and lowering the frequency ϖ, more understandable plots can be generated. The plot below contains three realizations of this harmonic process with white noise, except the variance of $W(t)$ was changed to 0.2 and the frequency ϖ was changed to $\frac{\pi}{50}$. The signals are noisy and have a random phase, but their behavior is easier to see. The code (which is very similar to the one used above) can be found on the linked GitHub repository.