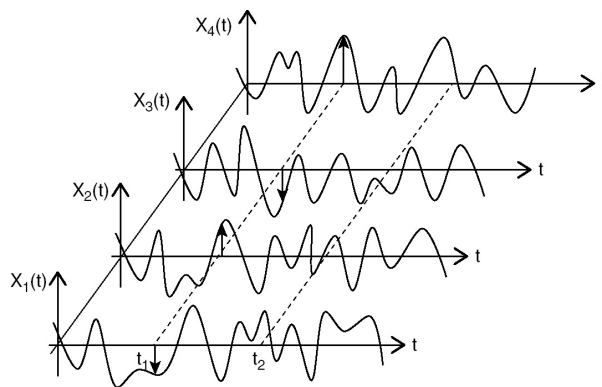

TRABAJO PRÁCTICO N°2

PROCESOS ESTOCÁSTICOS



Armida Abril Paula
Rodriguez Santiago
2022

índice

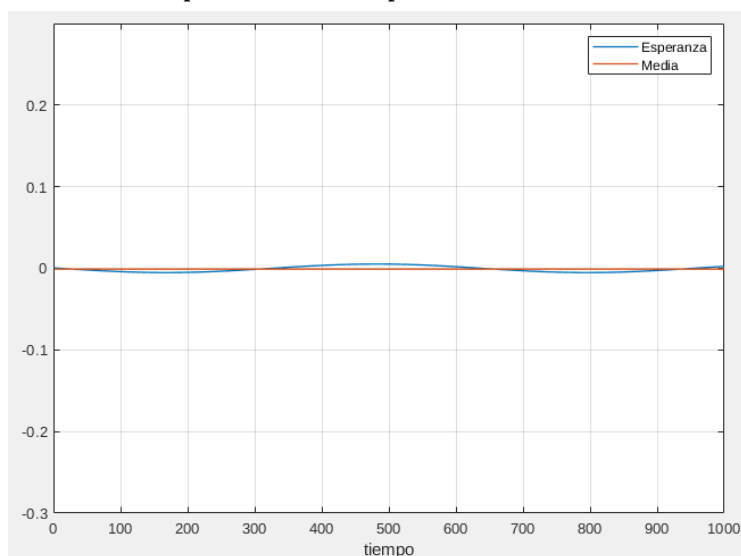
Ejercicio 1 :	2
a)	2
b)	2
c)	3
d)	3
Ejercicio 2:	4
a)	4
b)	5
c)	5
d)	6
Ejercicio 3:	6
a)	6
b)	8
c)	9
Ejercicio 4:	10
a)	10
b)	11
Ejercicio 5:	12
a)	12
b)	13
c)	14
Ejercicio 7:	15
a)	15
b)	16
c)	17
d)	18
Códigos	20

Ejercicio 1:

Dado el archivo *signal_exercise 1.mat*, calcular y dibujar.

a) Media como función del tiempo.

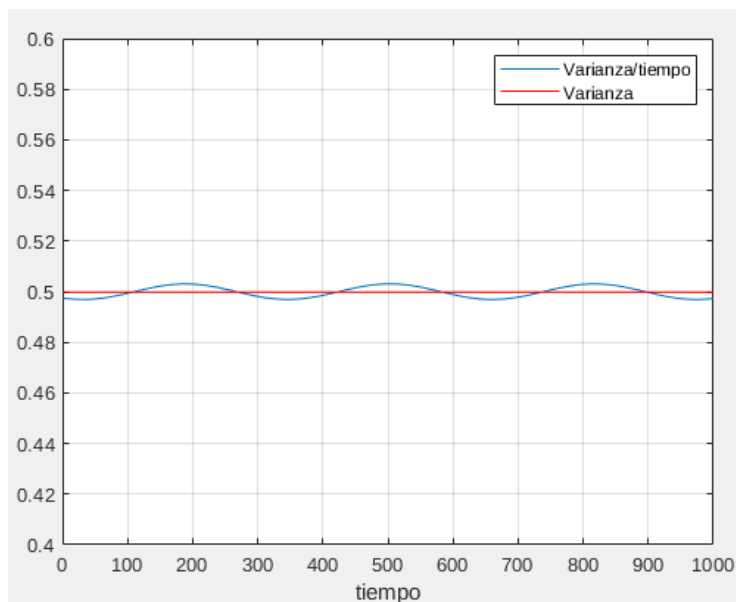
En la siguiente imagen se tiene en color rojo la media calculada con la función “*mean*”. Esta hace un promedio de todas las muestras de todos los experimentos. El resultado de la misma es de $\mu = -9.8458 \times 10^{-4}$. Luego, en color azul se tiene la esperanza en función del tiempo. Esta se calculó promediando en cada instante de tiempo todos los experimentos.



b) Varianza como función del tiempo.

En la siguiente imagen se muestra en color rojo la varianza calculada con la función “*var*” y su valor es $\sigma^2 = 0.4998$. Luego, en color azul se tiene la varianza en función del tiempo calculada como el promedio de la señal en todos los instantes de tiempo menos la media al cuadrado.

$$\sigma^2 = \sum \frac{(signal - media)^2}{N_{exp}}$$

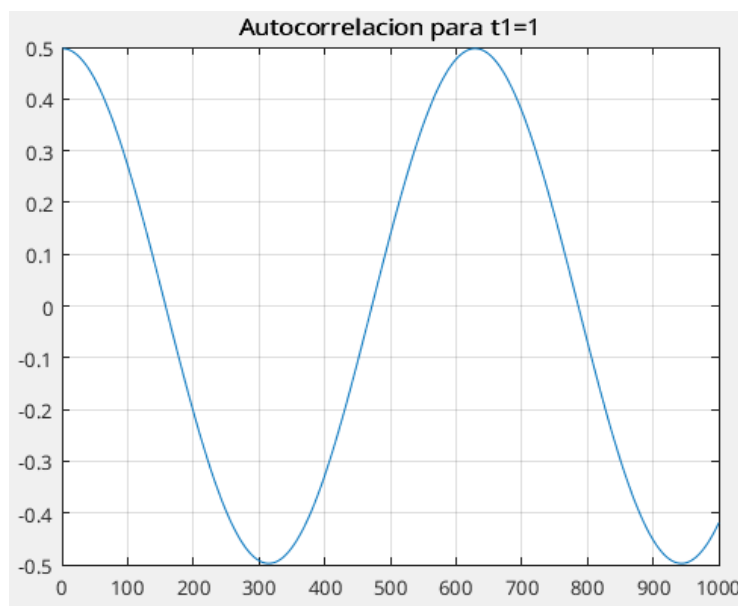


c) Función de autocorrelación $R_x(t_1, t_2) = E\{x(t_1) \cdot x^*(t_2)\}$

Para calcular la autocorrelación se utilizó el código del “TIP 1” de la consigna del trabajo práctico. El mismo se puede ver a continuación.

```
t1 = 1; % índice de tiempo a barrer
term1 = signal_to_process(:,t1);
z = term1.*conj(signal_to_process);
correlation_for_t1 = sum(z,1)/Nexp;
figure;
plot(correlation_for_t1); grid on; hold all;
```

La función de autocorrelación para una muestra en el tiempo devuelve el siguiente gráfico. Este es el caso de $t_1 = 1$.



d) ¿Considera que es un proceso WSS?

Para que un proceso sea un WSS se tiene que cumplir que:

$$E[x(t)] = \mu_x$$

y la autocorrelación depende sólo de $\tau = t_1 - t_2$:

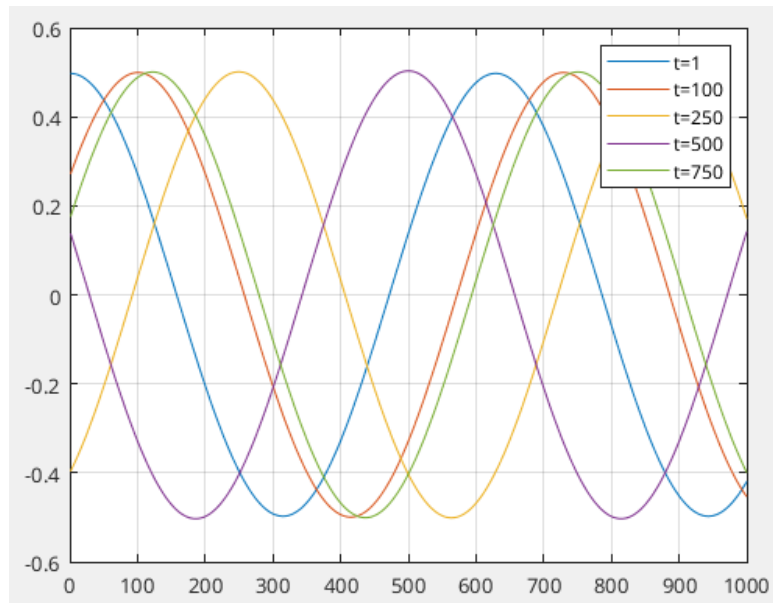
$$R_{xx}(t_1, t_2) = E[X(t_1) \cdot X^*(t_2)] = E[X(t + \tau) \cdot X^*(t)] = R_{xx}(\tau)$$

También,

$$R_{xx}(-\tau) = R_{xx}^*(\tau) \quad y \quad R_{xx}(0) = E[|X(t)|^2]$$

La primera condición se cumple y se puede observar en el inciso a) donde $E[X(t)]$ oscila entre valores muy cercanos a μ .

Para corroborar si se cumple la segunda condición se procedió a obtener la autocorrelación para múltiples instantes de t . Se decidió utilizar los valores de $t = 1, t = 100, t = 250, t = 500$ y $t = 750$. A continuación se graficaron todos los resultados.



Como se puede ver, las autocorrelaciones para todos los instantes de t son iguales con la diferencia de su respectivo desfase en el tiempo, dicho de otra forma, el valor de $R_x(t_1 + \tau) = R_x(t_2 + \tau) \quad \forall(t_1; t_2)$. Esto quiere decir que sin importar el t inicial elegido, la autocorrelación es la misma.

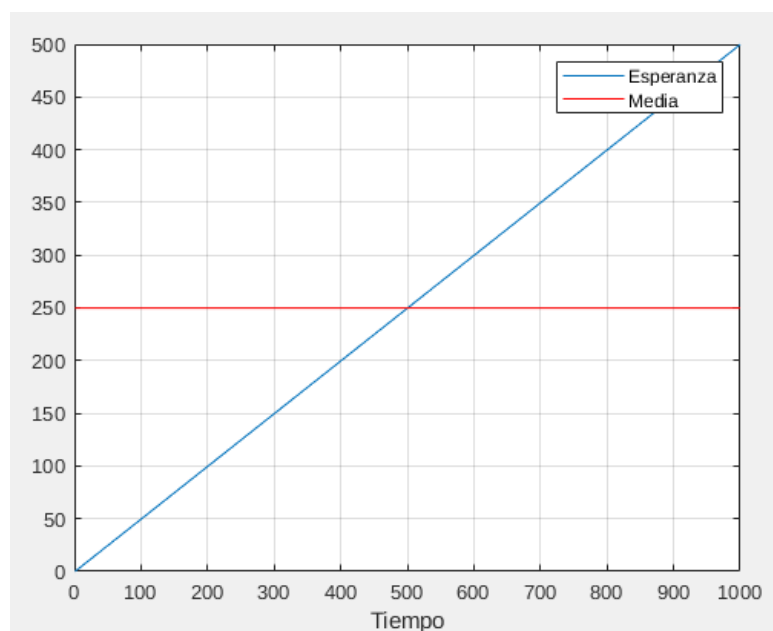
Por lo tanto se concluye que es un proceso estocástico estacionario en sentido amplio (WSS).

Ejercicio 2:

Idem Ejercicio 1 pero usando la señal `signal_exercise_2.mat`.

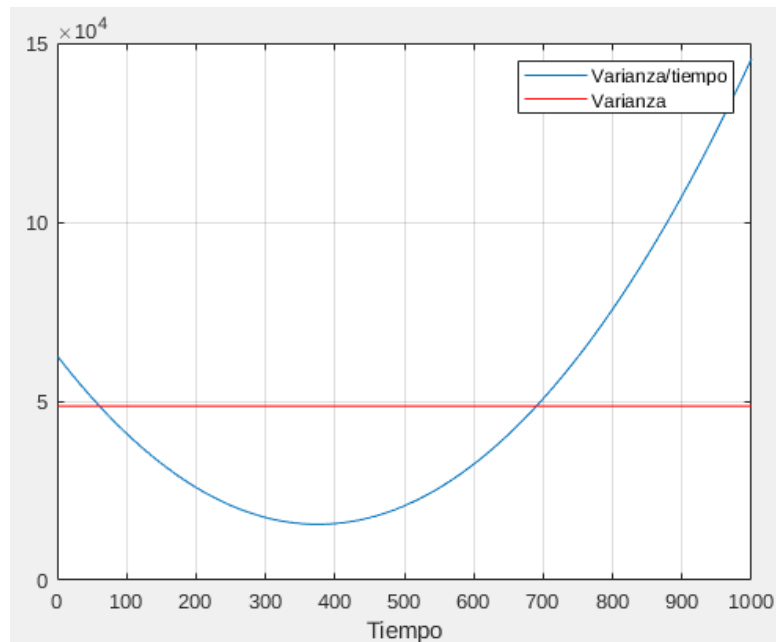
a) Media como función del tiempo.

En la siguiente imagen se tiene en color rojo la media calculada con la función “*mean*” con su resultado de $\mu = 249.8752$. Luego, en color azul, se tiene la esperanza en función del tiempo. El cálculo de ambas fue explicado en el “Ejercicio 1.a”.



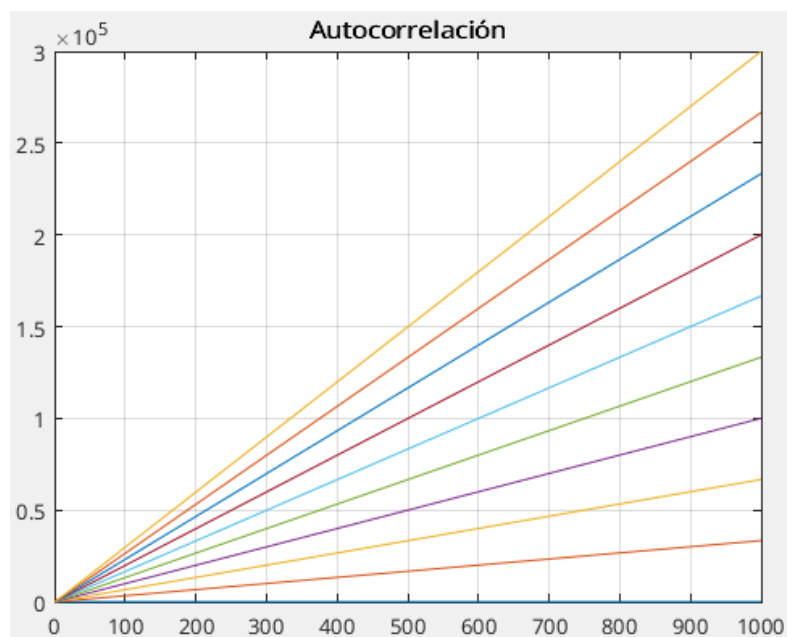
b) Varianza como función del tiempo.

En la siguiente imagen se muestra en color rojo la varianza calculada con la función “var” y su valor es $\sigma^2 = 4.8614 \times 10^4$. Luego, en color azul se tiene la varianza en función del tiempo calculada como se explicó en el “Ejercicio 1.b”.



c) Función de autocorrelación $R_x(t_1, t_2) = E\{x(t_1) \cdot x^*(t_2)\}$

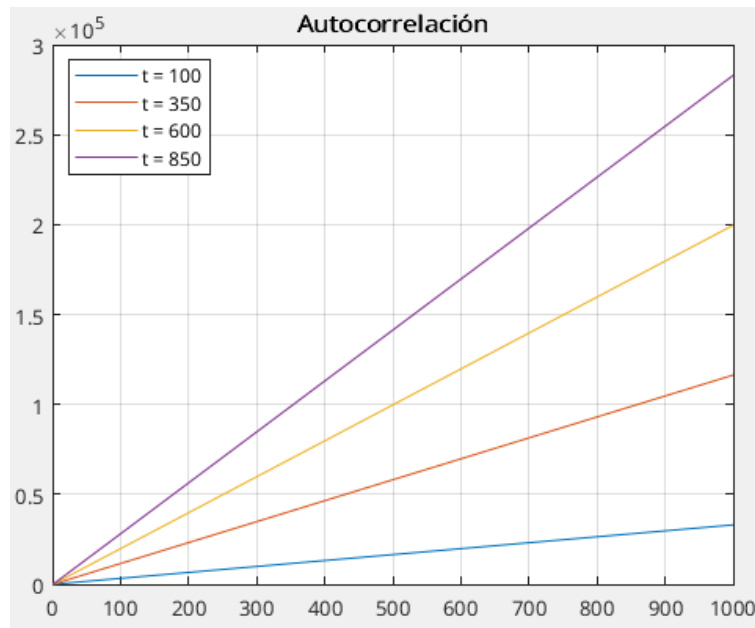
Para la función de autocorrelación se realizó el mismo procedimiento que en el “Ejercicio 1.c” pero para $t_1 = \{1, 100, 200, \dots, 1000\}$ y se obtuvieron los siguientes resultados.



Como se puede observar la función de autocorrelación es distinta para cada t elegido.

d) ¿Considera que es un proceso WSS?

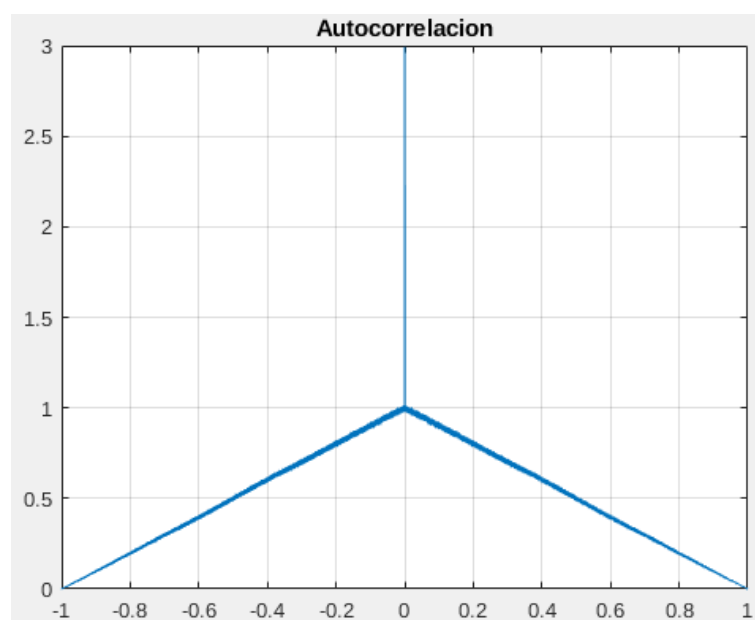
Por no cumplir la condición $E[x(t)] = \mu_x$ se considera que no es un proceso WSS como se puede observar en el inciso a). Además, tampoco se cumple la segunda condición. Esto se puede comprobar y corroborar a continuación, $R_x(t_1 + \tau) \neq R_x(t_2 + \tau) \quad \forall(t_1; t_2)$.



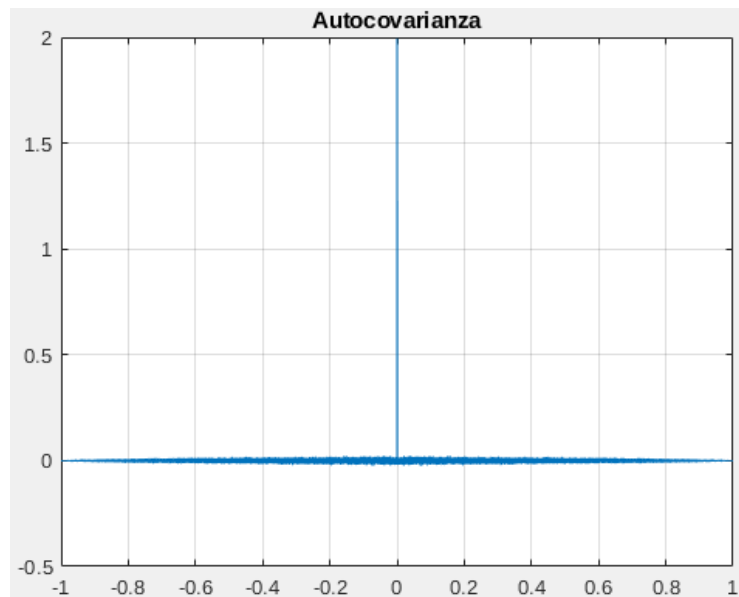
Ejercicio 3:

a) Comprobar la expresión de $R_x(m)$ y $C_x(m)$ usando simulaciones.

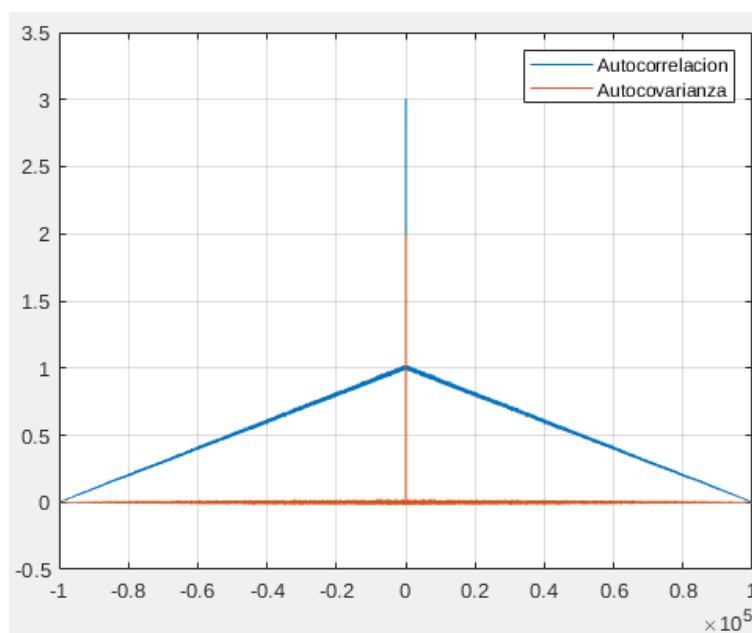
Se calcula $R_x(m)$ con la función "xcorr" con la opción 'biased' y se obtiene la siguiente autocorrelación.



Luego, se calcula $C_x(m)$ con la función “xcov” también con la opción ‘biased’ y se obtiene la siguiente autocovarianza.



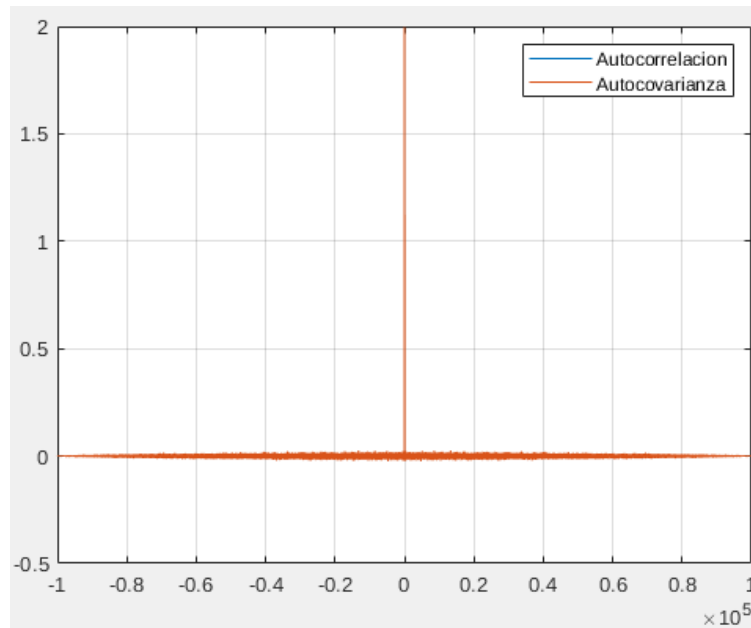
Si se superponen ambos gráficos se puede observar que para $R_x(0) = C_x(0) + \mu_x^2$.



Por lo que si se calcula la autocorrelación con:

```
Rx = xcorr(rbg-media^2, 'biased')
```

Al graficarlas superpuestas no se distinguen una de la otra.



b) Comprobar por simulación que $R_Y(m) = R_X(m) * h(m) * h^*(-m)$.

Se sabe que la respuesta al impulso de un filtro promediador móvil es la función rectángulo. Su fórmula es la siguiente y se debe a un rectángulo de altura 1/25 y largo 25.

$$h = 1/25 * \text{ones}(1, 25);$$

Primero se filtró el ruido gaussiano blanco con “h” usando la función “filter” con el siguiente código.

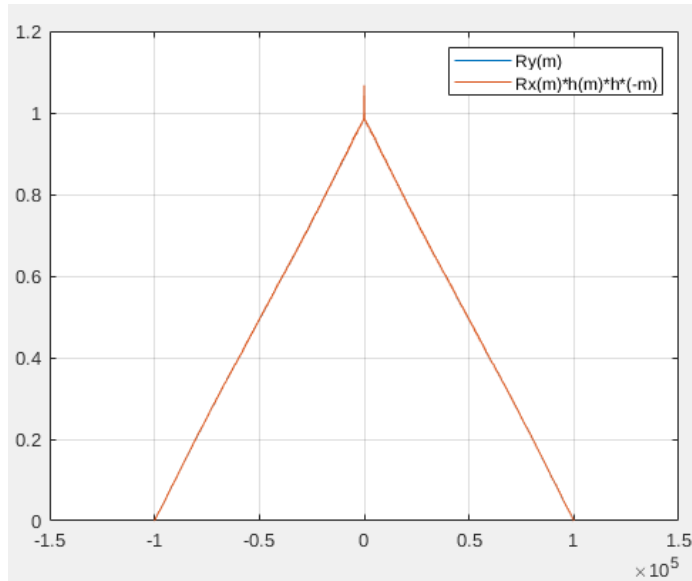
```
tline = 1:100e3;
rbg = wgn(length(tline), 1, 3) + 1;
h = 1/25 * ones(1, 25);
rbgf = filter(h, 1, rbg);
```

Con esto se obtuvo la salida del sistema $y(m) = \text{rbgf}$ (ruido blanco gaussiano filtrado). De la misma manera que se calculó la autocorrelación anteriormente, se calculó $R_Y(m)$.

Luego, del otro lado de la igualdad, se hizo la convolución entre $R_X(m)$ y $h(m) * h^*(-m) = g$ con el siguiente código.

```
t = -100023:1:100023;
hCI = conj(h(end:-1:1));
g = conv(hCI, h);
Rxconv = conv(Rx, g);
```

Por último, para comprobar la igualdad, se graficó la $R_Y(m)$ y la $R_{X\text{conv}} = R_X(m) * h(m) * h^*(-m)$ en un mismo gráfico. Como se puede ver son idénticas.



c) Calcular $E\{y(m)\}$ y justificar el valor que se obtiene.

Para el cálculo de $E\{y(m)\}$ se usó el siguiente código obteniéndose un resultado de $E\{y(m)\} \approx 1$, lo cual es igual a la media del ruido blanco gaussiano.

```
esperanzaf = sum(rbgf)/length(tline);
```

Para justificar el valor obtenido se decidió a usar la expresión de la definición de $E\{y(m)\}$.

$$\Rightarrow E\{y(t)\} = E\left\{\int_{-\infty}^{\infty} h(u)x(t - u)du\right\}$$

Como la esperanza es distributiva respecto de la suma, y en sí, una integral es una suma, se puede distribuir la esperanza dentro de la integral.

$$E\{y(t)\} = \int_{-\infty}^{\infty} E\{h(u)x(t - u)du\}$$

Luego, la esperanza también es distributiva respecto del producto.

$$E\{y(t)\} = \int_{-\infty}^{\infty} \{E(h(u))E(x(t - u))E(du)\}$$

Cómo $h(u)$ es la respuesta al impulso del filtro, es un proceso determinístico, por lo que la $E(h(u))$ es una constante y es $h(u)$. De la misma manera, $E(du)$ es determinístico y es du . Por lo que, la $E\{y(t)\}$ será:

$$E\{y(t)\} = \int_{-\infty}^{\infty} \{h(u)E(x(t - u))du\}$$

Por último, como el proceso es estacionario, la $E(x(t - u))$ no depende de u y su valor es u_x . Es así que:

$$E\{y(t)\} = \int_{-\infty}^{\infty} \{h(u)du\}u_x$$

Entonces la esperanza de la salida es la esperanza de la entrada multiplicado por la integral de la respuesta al impulso.

Cómo nuestro filtro es $h = 1/25 \cdot \text{ones}(1,25)$, calculando la integral se tiene un área de 1. Por lo tanto la esperanza de la salida es igual a la media de la entrada.

Ejercicio 4:

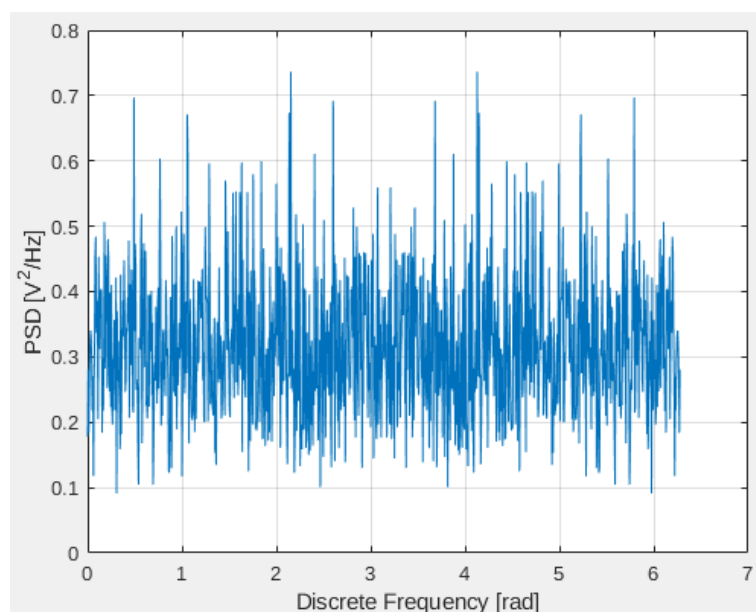
- a) **Computa la PSD del proceso de entrada $x(n)$ y verificá que la potencia del proceso es la integral debajo de la PSD.**

Se sabe que la potencia de un proceso estacionario (cómo el ruido blanco) es igual a la varianza del mismo ($R_x(0) = \sigma^2(x) = E\{|x(t)|^2\}$). Por lo tanto se calcula la varianza del proceso con la función “var” y se obtiene un resultado de $\sigma^2 \approx 2$ (este valor puede variar muy levemente para cada experimento por ser un proceso estocástico).

Para el cálculo de la PSD se utilizó el siguiente código extraído del script “psd_white_colored”.

```
NFFT=1024;  
[pxx,freqv] = pwelch(rbg(:)-media,[],0,NFFT,'twoside');  
figure  
plot(freqv,pxx)  
grid on  
xlabel('Discrete Frequency [rad]')  
ylabel('PSD [V^2/Hz]')
```

Graficándolo se obtuvo:



Hay dos maneras de calcular la potencia del PSD. La primera es obtener la media de la PSD y multiplicarla por 2π (debido a que el valor se encuentra en radianes).

```
potencia = mean(pxx)*2*pi;
```

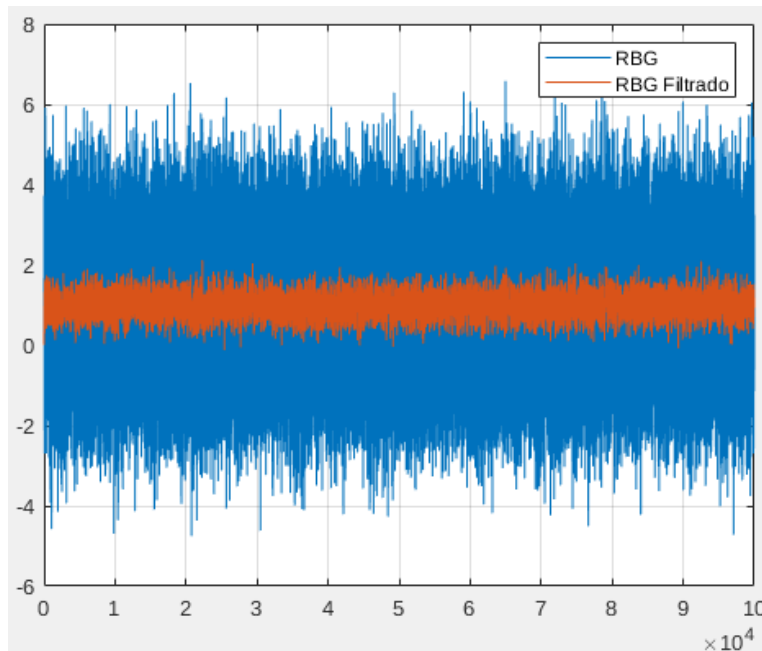
De esta forma nos da un resultado de $potencia \simeq 2$.

La segunda manera es calcular el área bajo la PSD, la cual también da el mismo resultado.

```
potenciaPSD = sum(pxx)*(2*pi/NFFT);
```

b) Computá la PSD del proceso de entrada $y(n)$ y verificá que la potencia del proceso es la integral debajo de la PSD. Además, verificá que $S_Y(f) = S_X(f)|H(f)|^2$.

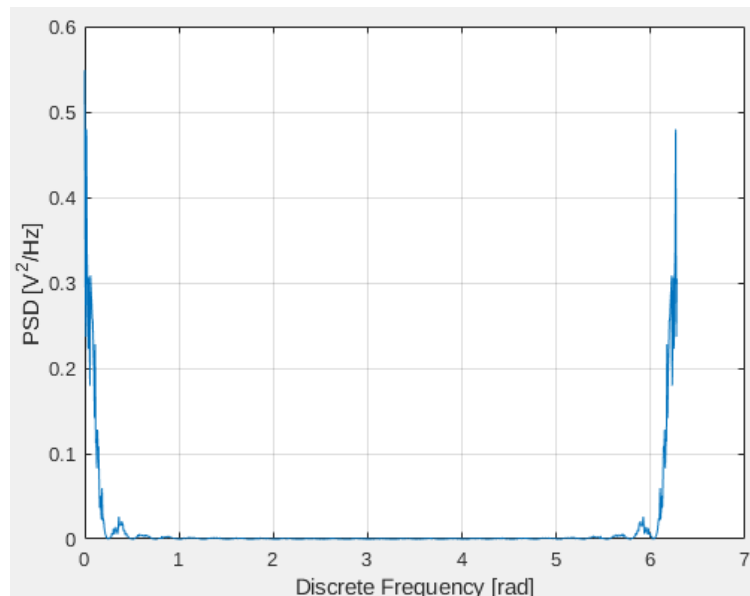
Al igual que en el punto a), el ruido filtrado sigue siendo un proceso estacionario y por lo tanto la potencia se calcula con la función “var”. El resultado obtenido fue $\sigma^2 = 0.079$, este valor es menor al anterior ya que se filtró gran parte de la potencia como se puede ver en la siguiente imagen.



Para el cálculo de la PSD se utilizó el siguiente código extraído del script “psd_white_colored”.

```
NFFT=1024;
[pxxf,freqvf] = pwelch(rbgf(:)-media,[],0,NFFT,'twoside');
figure
plot(freqvf,pxxf)
grid on
xlabel('Discrete Frequency [rad]')
ylabel('PSD [V^2/Hz]')
```

Graficándolo se obtuvo:



Si se calcula la potencia de las dos maneras explicadas en el punto a) se obtiene:

$$\text{potenciaf} = \text{mean}(\text{pxxf}) * 2 * \pi;$$

De esta forma se tiene un resultado de $\text{potencia} = 0.0831$.

Calculándola de la otra manera se obtiene el mismo resultado.

$$\text{potenciaPSDF} = \text{sum}(\text{pxxf}) * (2 * \pi / \text{NFFT});$$

Ejercicio 5:

- a) **Generá en Matlab un vector de muestras WGN con la potencia adecuada y demostrá que el vector está correctamente generado usando una PSD. Es importante que consideres la frecuencia de muestreo en el dibujo de la PSD. ¿Qué valor numérico tiene la densidad de potencia de ruido para este caso?**

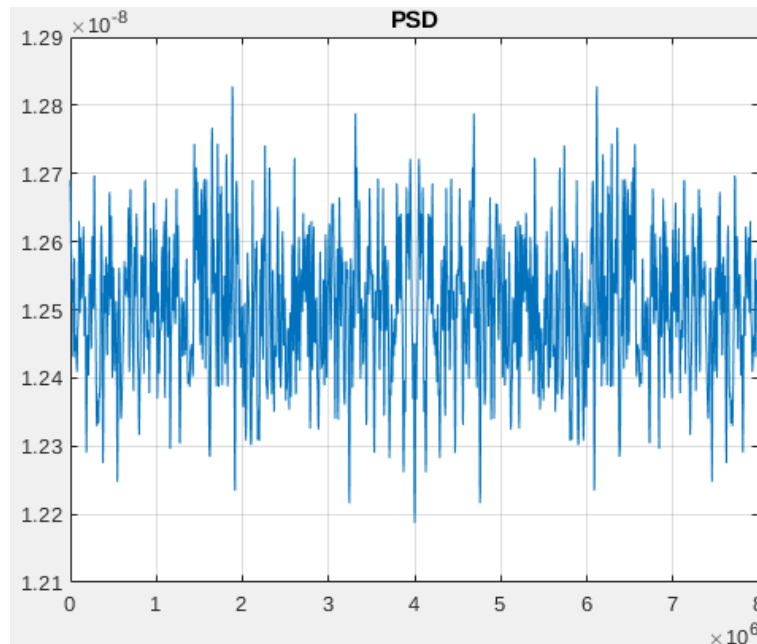
Para generar el vector de muestras WGN con una potencia de 100 [mW] se utiliza la función “randn” y se la implementa de la siguiente manera:

```
fs = 8e6; %frec de muestreo 8 [GHz]
t = 0:1/fs:1;
rbg = randn(size(t))*sqrt(100e-3);
```

Calculando la potencia con la función “var” se obtiene un valor de 0.1 W. Luego, se procede a generar su PSD usando la función “pwelch” como se muestra a continuación.

```
NFFT = 2048;
[pxx, freq] = pwelch(rbg-media_rbg, hanning(NFFT/2), 0, NFFT, fs, 'twoside');
```

La PSD obtenida se puede ver en la siguiente figura.

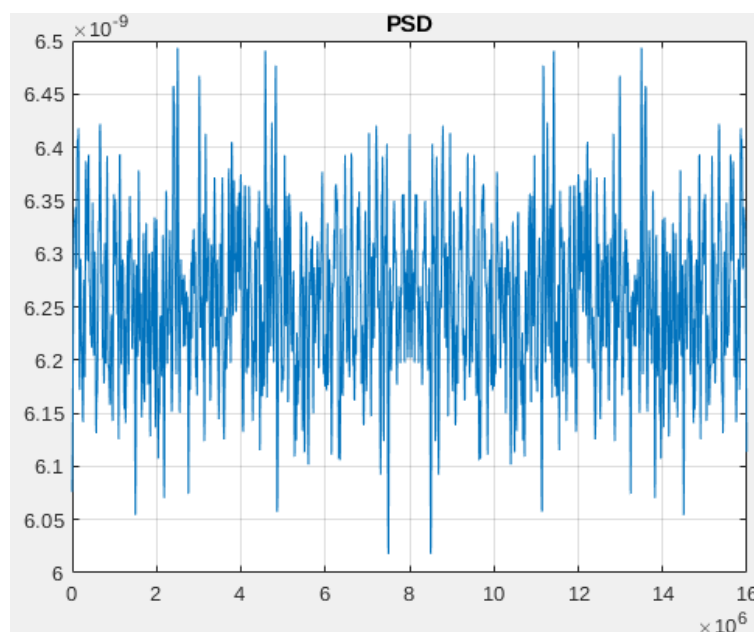


El vector está correctamente generado porque la PSD tiene potencia relativamente constante (± 0.2 [nW]) en todo el espectro hasta 8 [GHz], lo que corresponde con un ruido blanco.

Para el cálculo del valor de la densidad de potencia se calculó la media de la PSD y se obtuvo una potencia de 12.5 [nW].

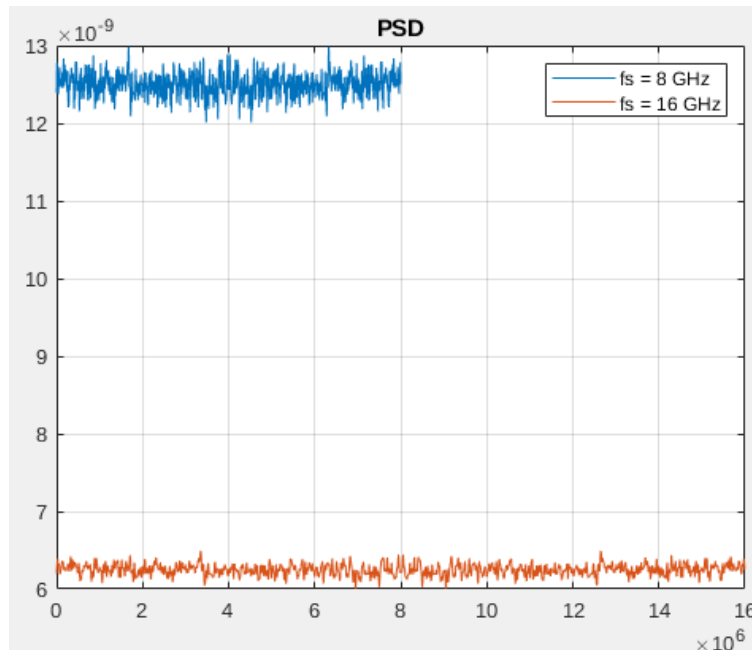
b) Suponé que el usuario del modelo quiere llevar la frecuencia de muestreo a 16 GHz. Repetí el inciso anterior y compará la PSD del ruido entre ambos casos. Interpretá el resultado.

Se utilizó el mismo vector de muestras WGN que en el inciso anterior y se cambió la frecuencia de muestreo a 16 [GHz] en la función "pwelch". El resultado obtenido fue el siguiente.



Con la función “mean” se calculó el valor de la PSD y se obtuvo un resultado de 6.25 [nW], lo cual es la mitad del valor de la PSD con frecuencia de muestreo de 8 [GHz].

Para comparar ambas PSD obtenidas para las dos frecuencias de muestreo se las graficó juntas.



Como se explicó anteriormente el valor de la PSD para la $f_s = 8$ [GHz] es el doble que para la $f_s = 16$ [GHz], pero la potencia total es la misma ya que el área bajo la última cubre el doble de espectro de frecuencia, por lo tanto a la hora de calcular la integral ambas áreas son iguales.

c) Suponé que en el ejercicio inicial, el usuario quiere modelar un ruido blanco gaussiano complejo de potencia 100mW. ¿Cómo hay que modificar el código para generar este nuevo vector de ruido?

Para que el ruido sea blanco gaussiano complejo con potencia 100[mW] se cambió el código a lo siguiente.

```
x = randn(size(t8))*sqrt(50e-3);  
xc = randn(size(t8))*sqrt(50e-3);  
rbgc = x + 1j*xc;
```

Cómo x y xc son procesos independientes, la potencia de rbgc es la suma de la potencia de cada proceso. Es así que se tuvo que modificar la potencia los mismos a la mitad para que la suma de ambas dé 100[mW].

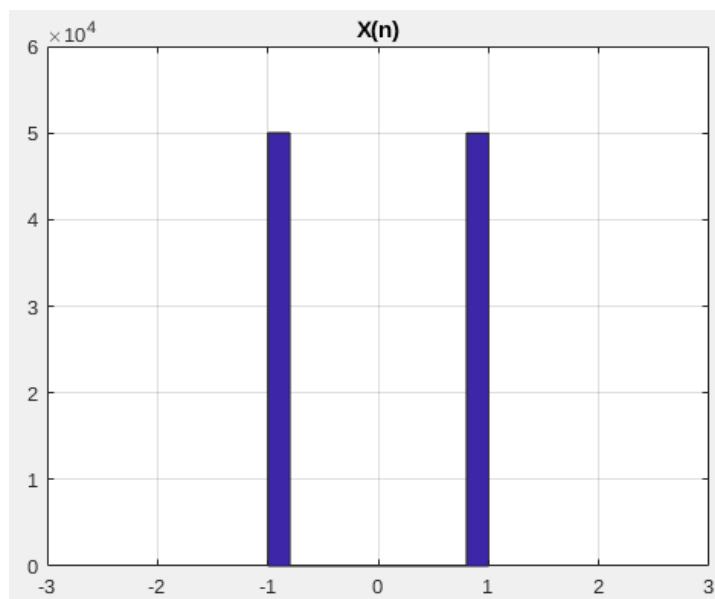
Ejercicio 7:

- a) Realizar el histograma de X, N e Y, y explicar los resultados obtenidos. Asumir que la potencia del ruido es $\sigma_N^2 = 0.1$

Para el experimento se decidió utilizar 100×10^3 muestras. La señal de datos aleatoria $X_N(n)$ se generó con el siguiente código.

```
imin = 1;  
imax = 2;  
t = 1:100e3;  
X_n = randi([imin, imax],[1, length(t)]);  
  
for i=1:length(t)  
    if X_n(i) == 2  
        X_n(i) = -1;  
    end  
end
```

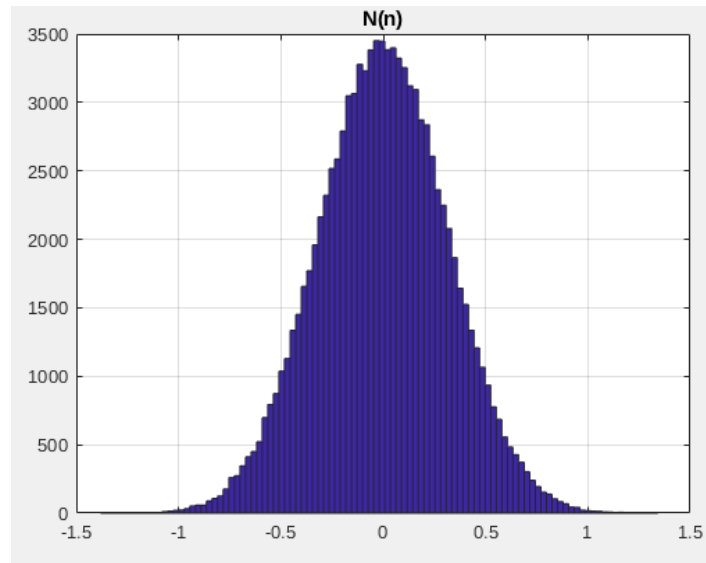
Luego, se graficó el histograma y se observa que la $P(-1) = P(1) \approx 0.5$.



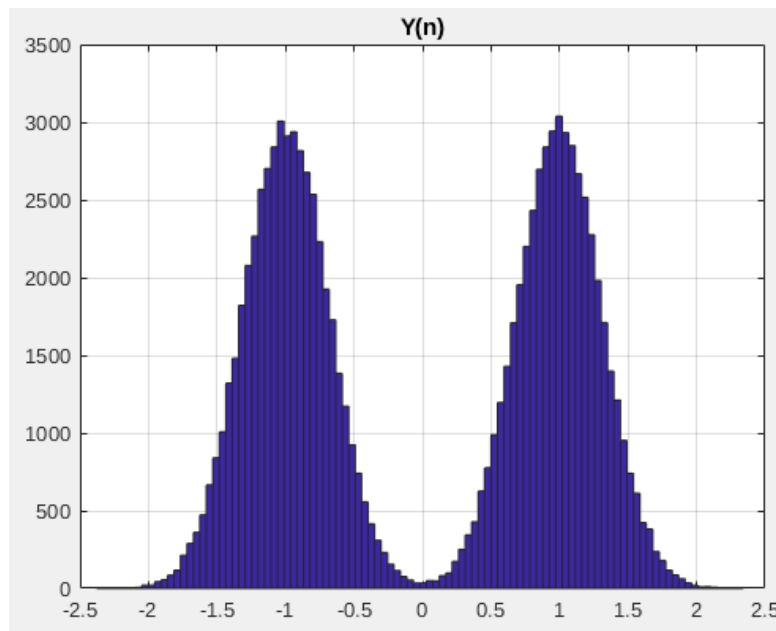
Seguido, se generó el ruido blanco gaussiano de la misma manera que se hizo en los incisos anteriores.

```
N_n = sqrt(0.5)*randn(size(t));
```

También se graficó el histograma. Se puede ver que la distribución del ruido es gaussiana, con media cero.



Por último se tiene la señal $Y(n) = X(n) + N(n)$, la cual es la señal obtenida en el detector.



b) Calcular la SNR para este caso (teoría y simulación).

Para el cálculo de la SNR se deben obtener los valores de la potencia de la señal (P_s) y de la potencia del ruido (P_n). La potencia de la señal fue dada en la consigna y tiene un valor de $P_s = 1$. Luego, la potencia del ruido se calculó con la función “var” ya que es un ruido blanco y dió un resultado de $P_n = 0.1$.

$$P_n = \text{var}(N_n);$$

Por lo tanto, el valor de la SNR es la diferencia entre ambas.

$$SNR = \frac{P_s}{P_n} = 10$$

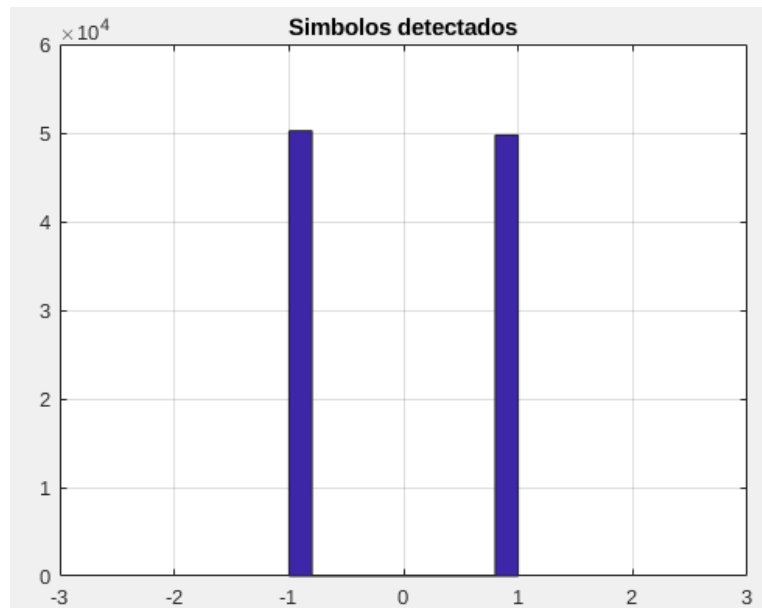
c) Medir el Bit Error Rate(BER) y determinar cuántos de los Y decididos son errores.

En este inciso se tiene como objetivo determinar el valor de los datos que se mandan por un canal AWGN. Para esto, como se sabe que los símbolos a transmitir pueden ser -1 ó 1, se toma como umbral de decisión el valor 0. Esto significa que si la muestra recibida es mayor a 0, se lo considera un 1. De forma análoga ocurre con los valores menores a 0, que se consideran un -1.

Para poder lograr lo comentado anteriormente, se utilizó el siguiente código.

```
Y_decidido = Y_n;  
Y_decidido(Y_n > 0) = 1;  
Y_decidido(Y_n < 0) = -1;
```

Se graficó un histograma para poder observar de mejor manera el comportamiento del algoritmo de decisión.



A simple vista se puede observar que este gráfico es muy similar al histograma de los símbolos enviados ($X_n(n)$). Esto indica que el detector de símbolos funciona correctamente.

Luego, es necesario medir la BER (tasa de error de bits). Para esto se debe obtener la cantidad de errores de la detección (F), lo cual se llevó a cabo con el siguiente código.

```
F = 0;  
M = length(t);  
for i = 1:length(t)  
    if X_n(i) ~= Y_decidido(i)  
        F = F + 1;  
    end  
end
```

En el mismo se compara el arreglo de símbolos enviados ($X_n(i)$) con el arreglo de símbolos detectados ($Y_{DECIDIDO}(i)$). Cuando se encuentra una diferencia se lo considera un error y se aumenta en uno la cantidad total de errores.

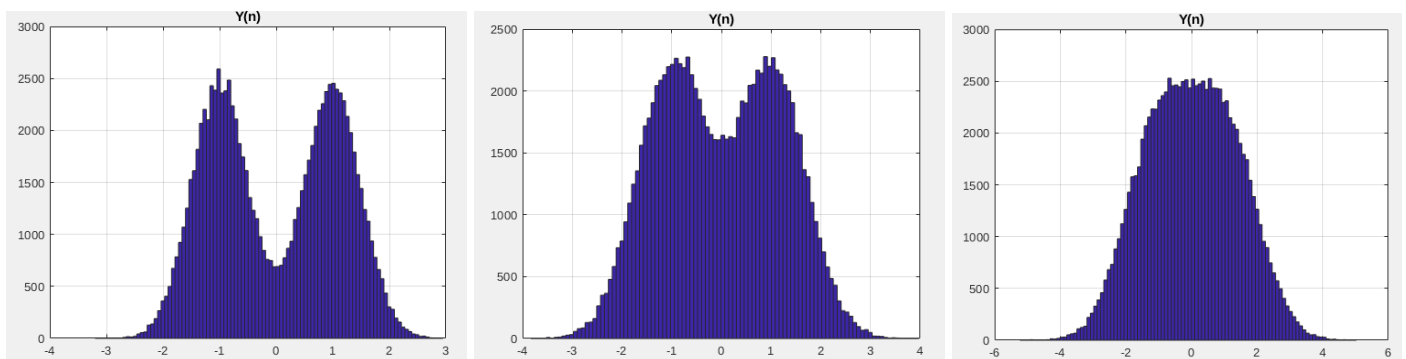
Por último, la BER se obtiene con la siguiente fórmula.

$$BER = \frac{F}{M}$$

Para el experimento con el que se viene trabajando, el valor es 6.4×10^{-4} .

d) ¿Qué pasa si la SNR es más grande? ¿Y si es más chica? Justificá tu respuesta usando simulaciones, o una explicación teórica.

Un aumento de la SNR implicaría en nuestro caso un aumento de la potencia del ruido, para lograr esto se aumenta la varianza del ruido blanco gaussiano. A continuación se muestran los histogramas de $Y(n)$ obtenidos con $\sigma^2 = 0.25$, $\sigma^2 = 0.5$ y $\sigma^2 = 1$.

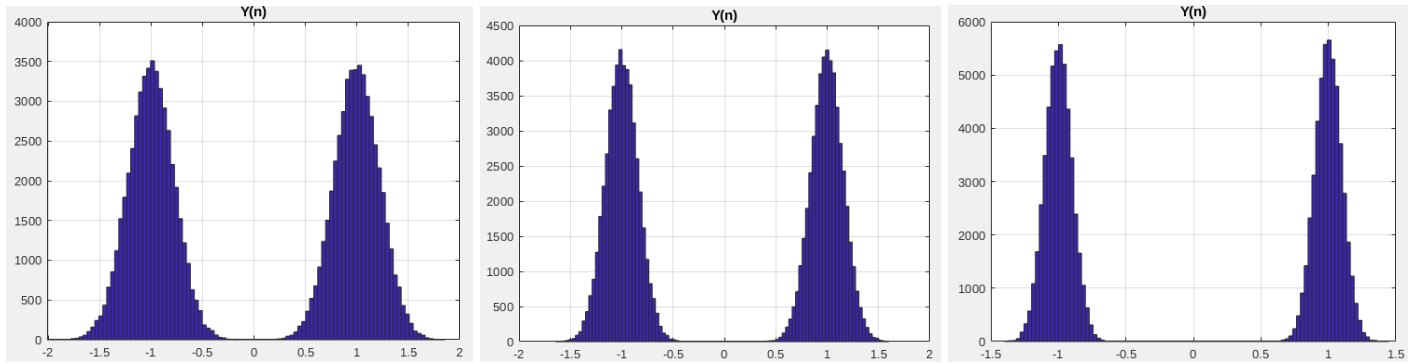


Como se puede observar, la distinción entre los distintos símbolos recibidos se hace más difícil a medida que la potencia del ruido aumenta debido al solapamiento de las detecciones, esto se traduce en más errores de detección y, por lo tanto, un aumento del BER. A continuación se muestran los resultados obtenidos para las 3 potencias de ruido.

σ^2	SNR	Errores de bit	BER
0.25	4	2250	$0.0225 = 2.25\%$
0.5	2	7850	$0.0785 = 7.85\%$
1	1	16000	$0.16 = 16\%$

Por lo tanto se puede concluir que una disminución de la SNR implica un aumento del BER.

Un aumento de la SNR implicaría en nuestro caso una disminución de la potencia del ruido, para lograr esto se disminuye la varianza del ruido blanco gaussiano. A continuación se muestran los histogramas de $Y(n)$ obtenidos con $\sigma^2 = 0.05$, $\sigma^2 = 0.025$ y $\sigma^2 = 0.01$.



Como se puede observar, la distinción entre los distintos símbolos recibidos se hace más evidente a medida que la potencia del ruido aumenta, el gráfico es cada vez más similar al histograma de los símbolos generados. Debido a que el solapamiento entre las detecciones es cada vez menor, se puede concluir que también será menor la detección de errores y por lo tanto también disminuirá el BER. A continuación se muestran los resultados obtenidos para las 3 potencias de ruido.

σ^2	SNR	Errores de bit	BER
0.05	20	0	0
0.025	40	0	0
0.01	100	0	0

En este caso, como la SNR de los incisos anteriores ya tenía un valor relativamente alto ($SNR = 10$) y los errores de bit y BER eran bajos, no se encontraron errores de bit para las SNR mayores a 10 con las que se realizaron los experimentos.

Códigos

- [Ejercicio 1](#)
- [Ejercicio 2](#)
- [Ejercicio 3](#)
- [Ejercicio 4](#)
- [Ejercicio 5](#)
- [Ejercicio 6](#)
- [Ejercicio 7](#)