
Table of Contents

Practica 5 -- Rafael Jose Martin Pelaez	1
Ejercicio 1.1	3
Ejercicio 1.2	7
Ejercicio 1.3	11
Ejercicio 1.4	15
Ejercicio 1.5	20
Ejercicio 1.6	25
Ejercicio 1.7	28

Practica 5 -- Rafael Jose Martin Pelaez

```
clear all; close all; clc;
start;
```

Warning: The value of local variables may have been changed to match the globals. Future versions of MATLAB will require that you declare a variable to be global before you use that variable.

```
*****
*
*           Laboratorio de Señales y Comunicaciones
*
*           Versión Modificada por
*           Javier Ramos, Marzo 1998
*           Autores del Freeware:
*
*           M. ZEYTINOGLU and B. MA
*           Department of Electrical & Computer & Computer Engineering
*
*           Ryerson Polytechnic University
*           Toronto, Ontario, CANADA
*
*****
```

Bienvenido al laboratorio. Antes de que empiece con las simulaciones, se deben inicializar un numero de variables.

Warning: The value of local variables may have been changed to match the globals. Future versions of MATLAB will require that you declare a variable to be global before you use that variable.

Warning: The value of local variables may have been changed to match the

globals. Future versions of MATLAB will require that you declare a variable to be global before you use that variable.

Ejercicio 1.1

```
%Generamos la secuencia y establecemos Rb
nbits=10; %Tamaño de la secuencia binaria
b=round(rand(1,nbits));
Rb=1000;

%NRZ Polar
x=wave_gen(b, 'polar_nrz', Rb);
sprintf('EL codigo de linea NRZ Polar resultante es:')
fprintf('%d ', x)
figure, waveplot(x); title('NRZ Polar');

%NRZ Unipolar
x=wave_gen(b, 'unipolar_nrz', Rb);
sprintf('EL codigo de linea NRZ Unipolar resultante es:')
fprintf('%d ', x)
figure, waveplot(x); title('NRZ Unipolar');

%RZ Polar
x=wave_gen(b, 'polar_rz', Rb);
sprintf('EL codigo de linea RZ Polar resultante es:')
fprintf('%d ', x)
figure, waveplot(x); title('RZ Polar');

%RZ Unipolar
x=wave_gen(b, 'unipolar_rz', Rb);
sprintf('EL codigo de linea RZ Unipolar resultante es:')
fprintf('%d ', x)
figure, waveplot(x); title('RZ Unipolar');

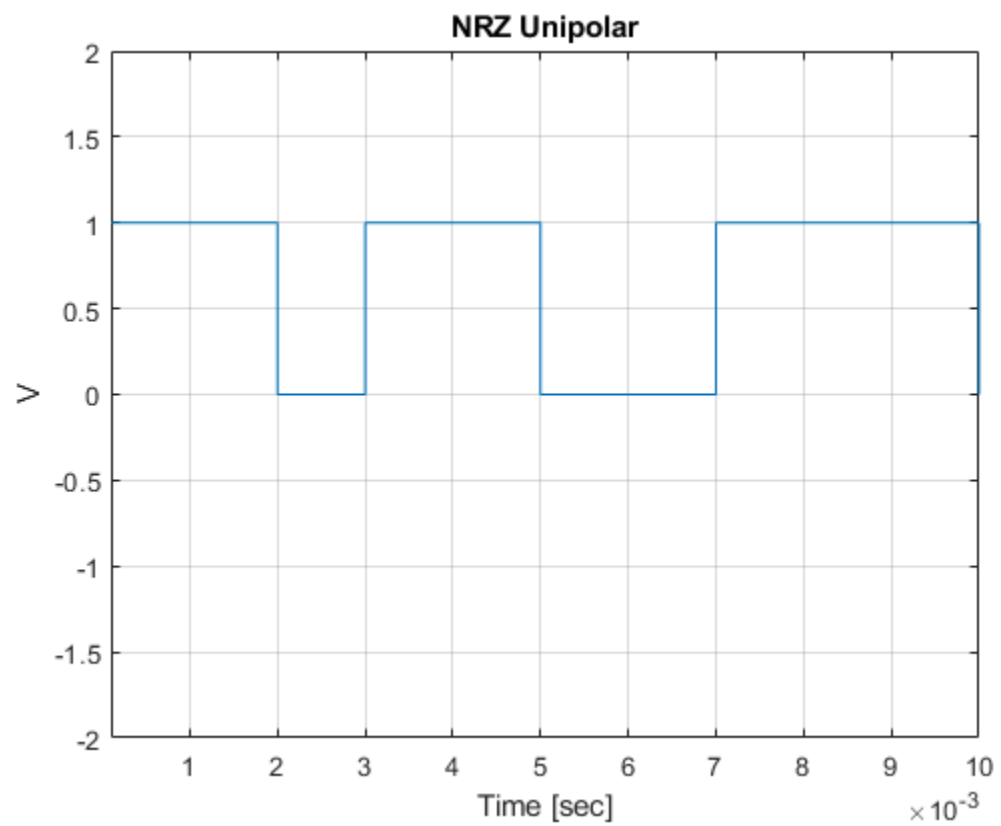
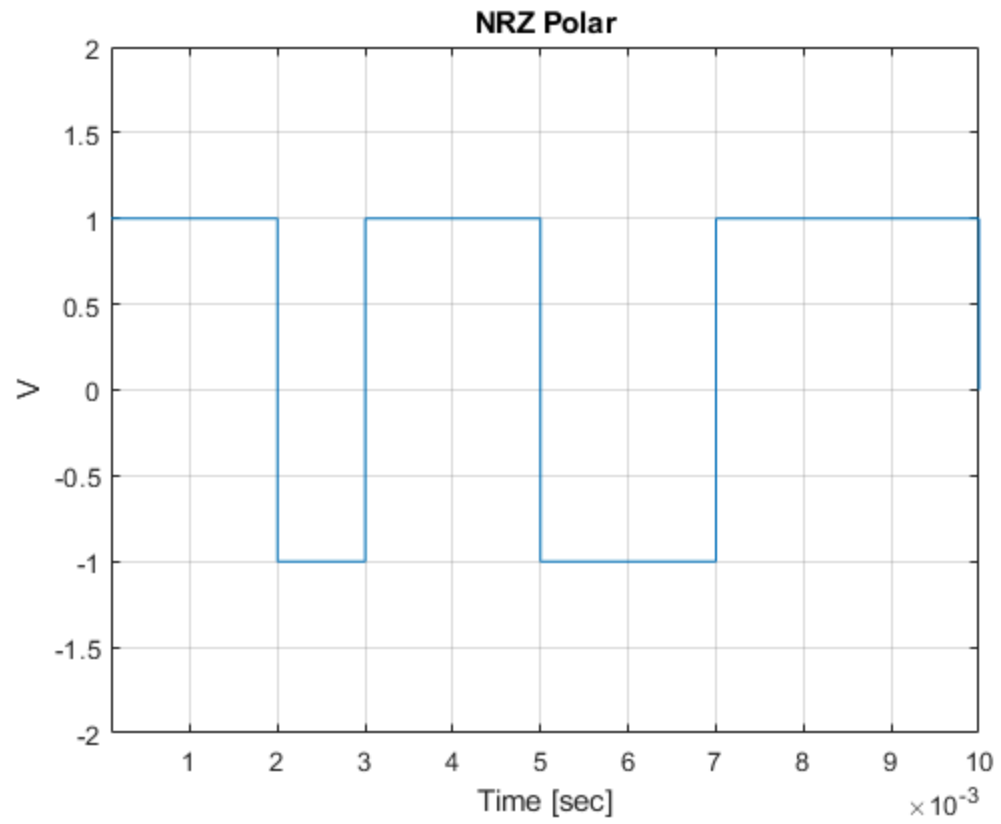
%Manchester
x=wave_gen(b, 'manchester', Rb);
sprintf('EL codigo de linea Manchester resultante es:')
fprintf('%d ', x)
figure, waveplot(x); title('Manchester');

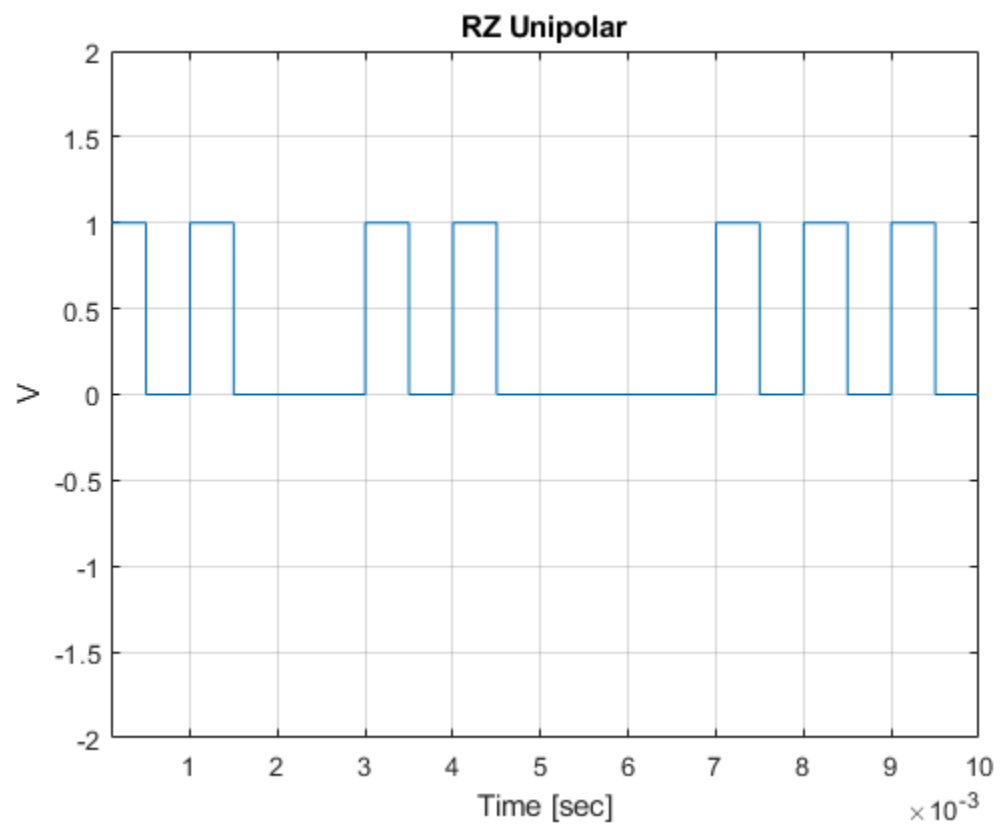
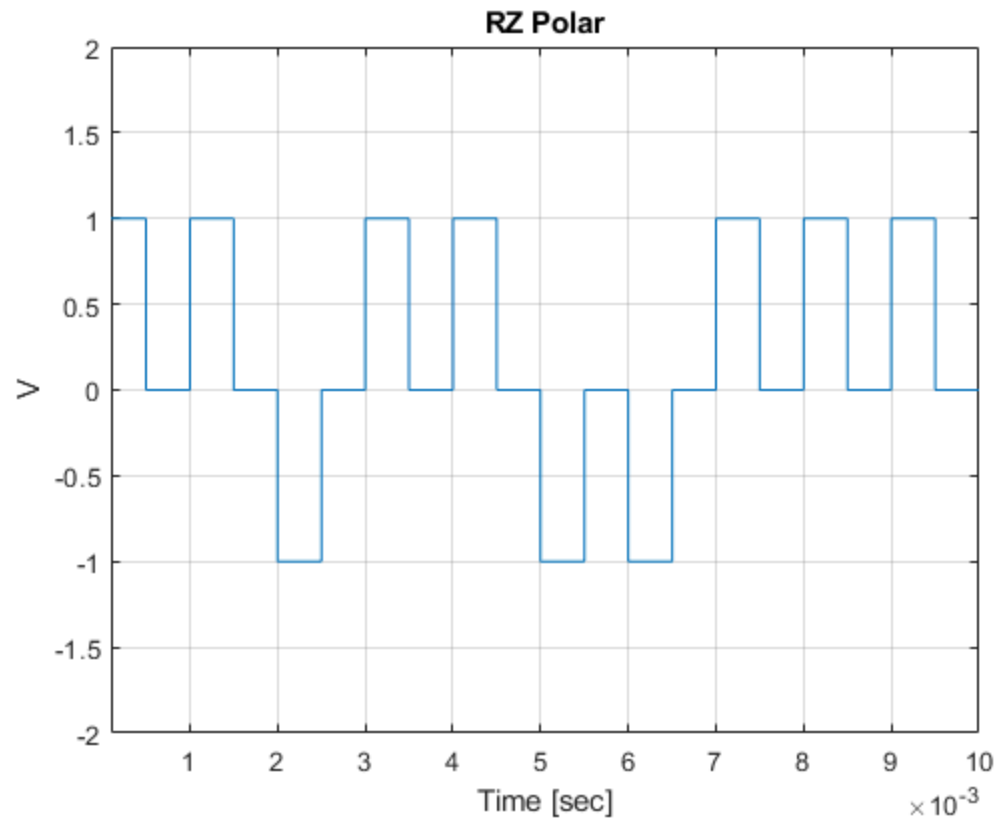
%Los valores teoricos medios de potencia no coinciden debido a que en
la
%teoria lo vemos para un tiempo infinito, al verlo en un tiempo finito
%cambia

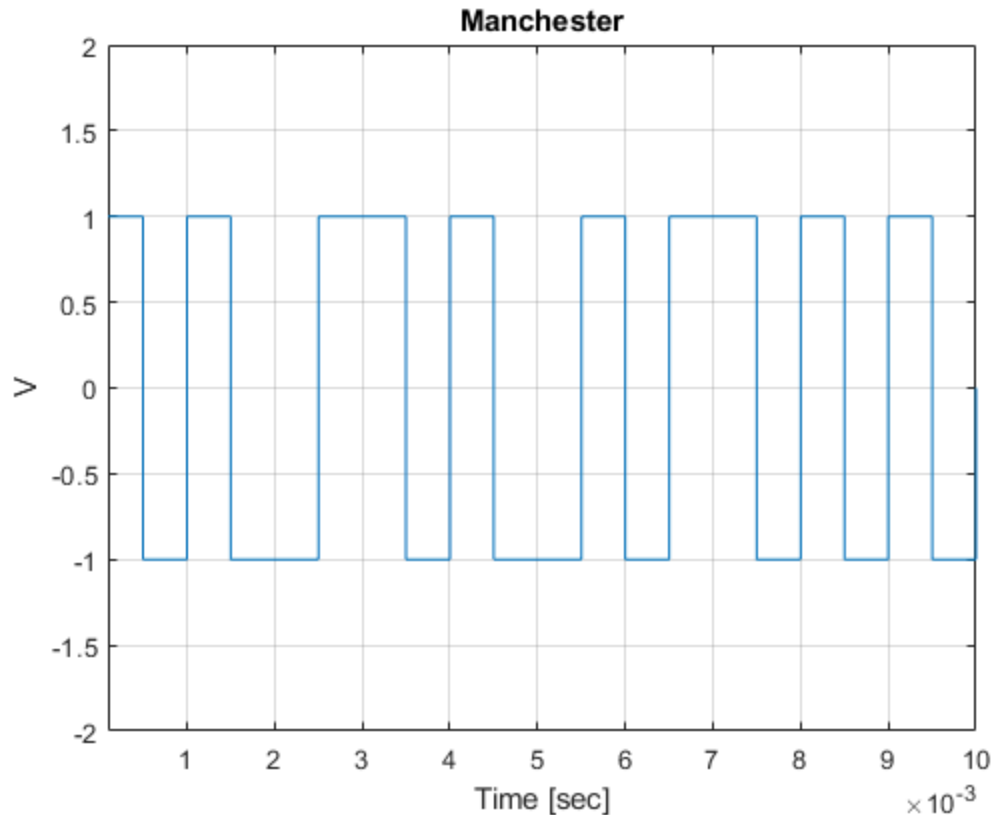
ans =

    'EL codigo de linea NRZ Polar resultante es:'

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
```





Ejercicio 1.2

```
%Generamos la secuencia y establecemos Rb
nbits=1000; %Tamaño de la secuencia binaria
b=round(rand(1,nbits));
Rb=1000;
```

```
%NRZ Polar
x=wave_gen(b, 'polar_nrz', Rb);
figure, psd(x); title('NRZ Polar');
```

```
%NRZ Unipolar
x=wave_gen(b, 'unipolar_nrz', Rb);
figure, psd(x); title('NRZ Unipolar');
```

```
%RZ Polar
x=wave_gen(b, 'polar_rz', Rb);
figure, psd(x); title('RZ Polar');
```

```
%RZ Unipolar
x=wave_gen(b, 'unipolar_rz', Rb);
figure, psd(x); title('RZ Unipolar');
```

```
%Manchester
x=wave_gen(b, 'manchester', Rb);
```

```
figure,psd(x);title('Manchester');
```

```
%Tabla
```

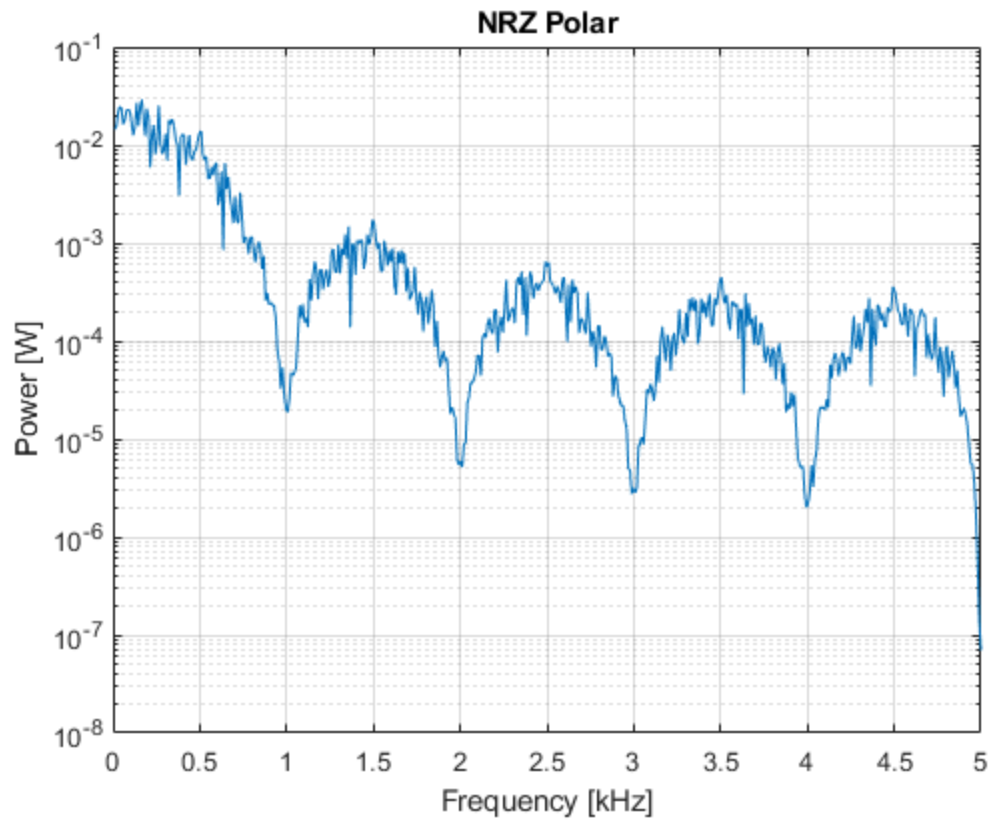
```
%NRZ Unipolar --> PPE=0      PNE=1  SPE=1.5  SNE=2  AB=1  (Datos en KHz)
```

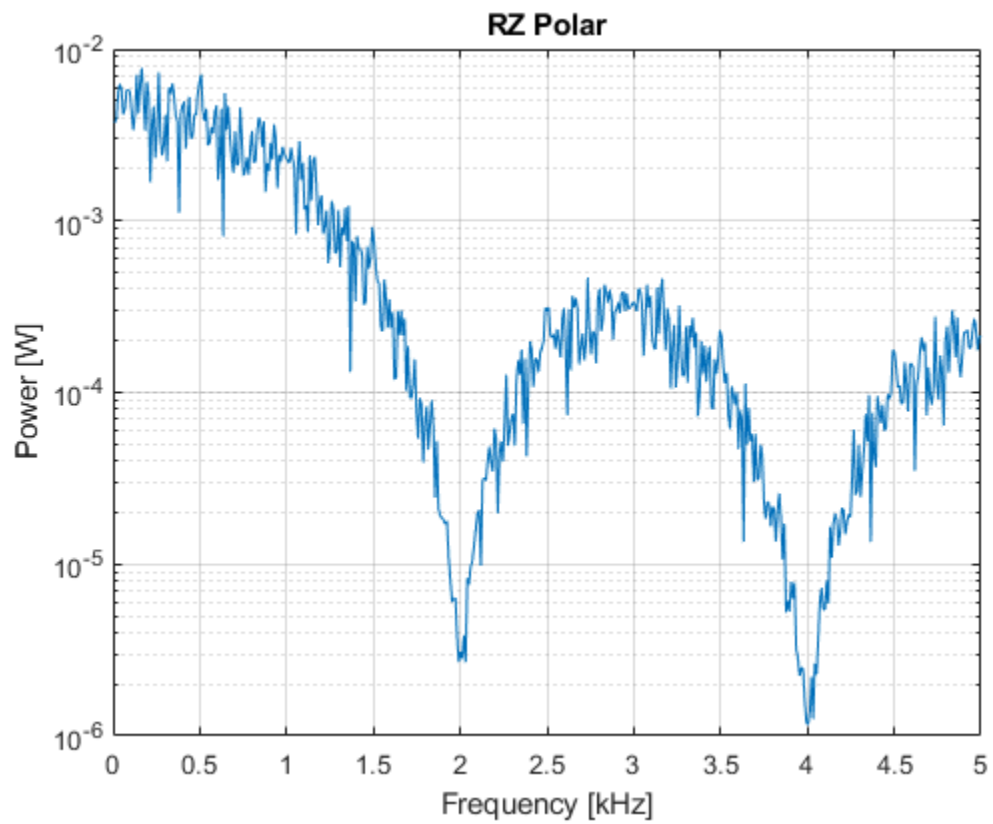
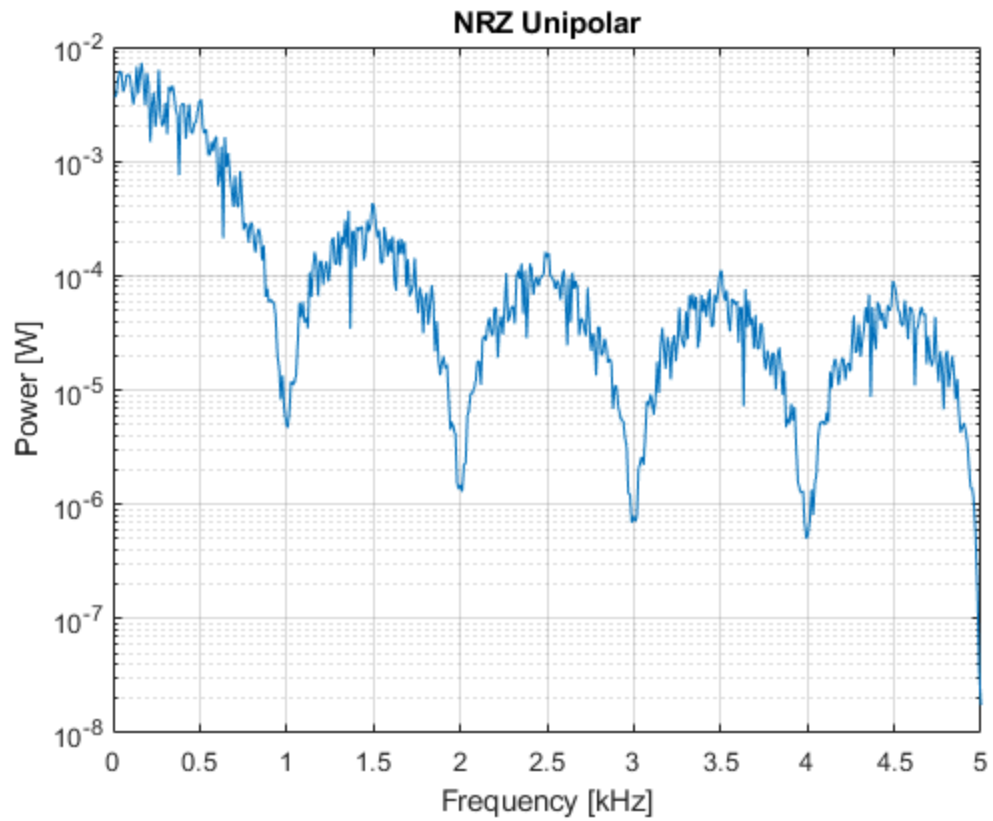
```
%NRZ Polar    --> PPE=0      PNE=1  SPE=1.5  SNE=2  AB=1  (Datos en KHz)
```

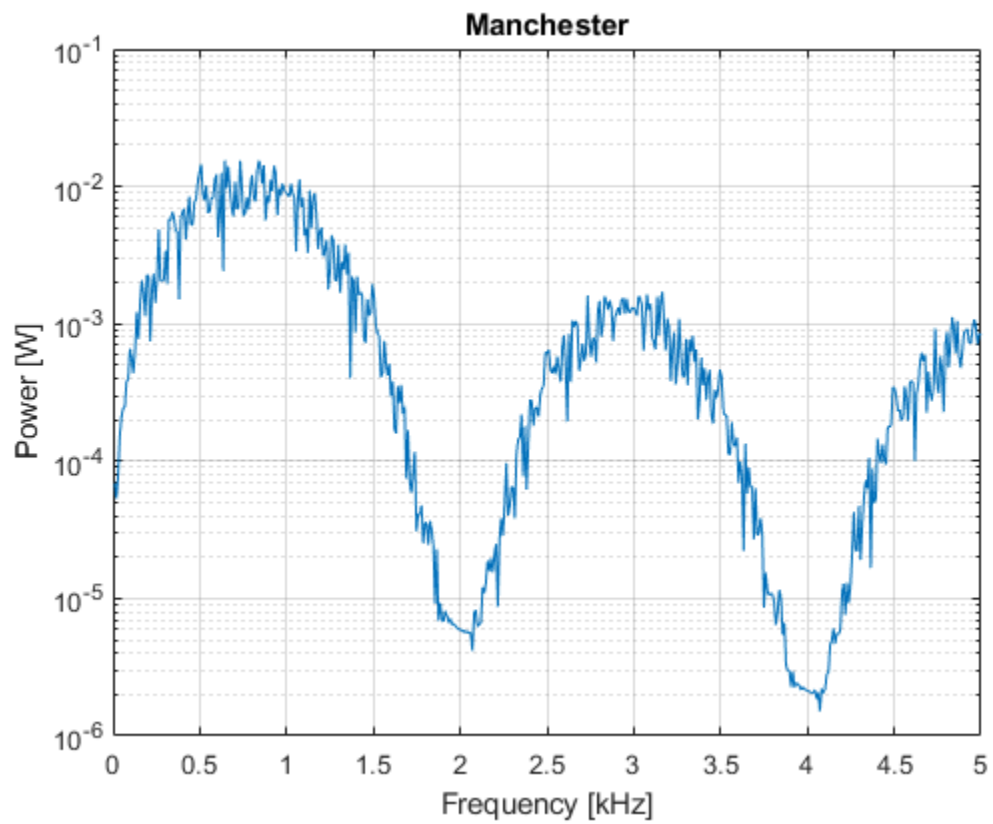
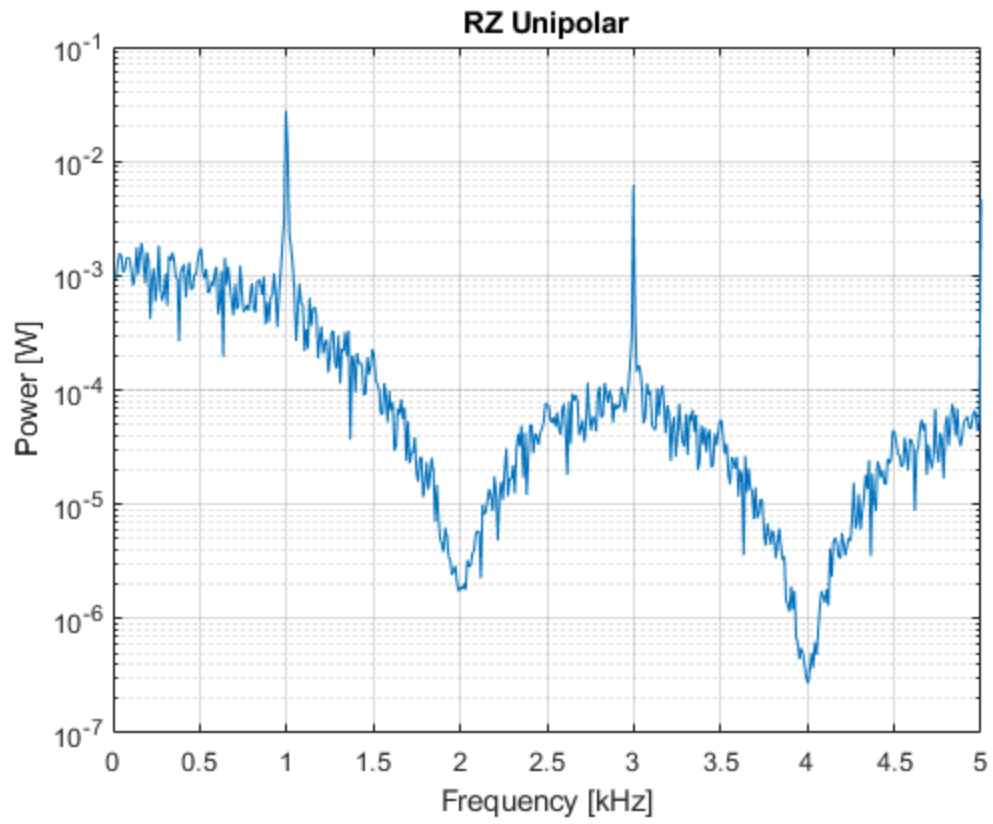
```
%RZ Unipolar   --> PPE=1      PNE=2  SPE=3     SNE=4  AB=2  (Datos en KHz)
```

```
%RZ Polar     --> PPE=0      PNE=2  SPE=3     SNE=4  AB=2  (Datos en KHz)
```

```
%Manchester    --> PPE=0.75  PNE=2  SPE=3     SNE=4  AB=2  (Datos en KHz)
```







Ejercicio 1.3

```
%Generamos la secuencia y establecemos Rb
nbits=10; %Tamaño de la secuencia binaria
b=round(rand(1,nbits));
Rb=1000;

%NRZ Polar
x=wave_gen(b, 'polar_nrz', Rb);

%Generamos el canal
ancho_banda=4500;
G=1;

for pot_ruido=0:0.2:1

    y=channel(x,G,pot_ruido,ancho_banda);

    figure,waveplot(x);hold on;waveplot(y);
    title(['Con un Potencia de Ruido de ' num2str(pot_ruido) 'W'])
    legend('S. Antes del Canal','S. Despues del Canal');
end

%%A
%Se observa una diferencia clara, ya que aparece el ruido como picos
    en los
%que aumenta y disminuye la amplitud de la señal.

%%B
%A partir de 0.6W de Potencia de Ruido, es dificil distinguir la señal
%original ya que a veces aparece un pico de amplitud demasiado grande,
    que
%hace que pueda confundirse de valor el decisor.

Warning: AXIS('STATE') is obsolete and will be eliminated in future
versions.
Use GET(GCA,...) instead.
Warning: AXIS('STATE') is obsolete and will be eliminated in future
versions.
Use GET(GCA,...) instead.
Warning: AXIS('STATE') is obsolete and will be eliminated in future
versions.
Use GET(GCA,...) instead.
Warning: AXIS('STATE') is obsolete and will be eliminated in future
versions.
Use GET(GCA,...) instead.
Warning: AXIS('STATE') is obsolete and will be eliminated in future
versions.
Use GET(GCA,...) instead.
Warning: AXIS('STATE') is obsolete and will be eliminated in future
versions.
Use GET(GCA,...) instead.
```

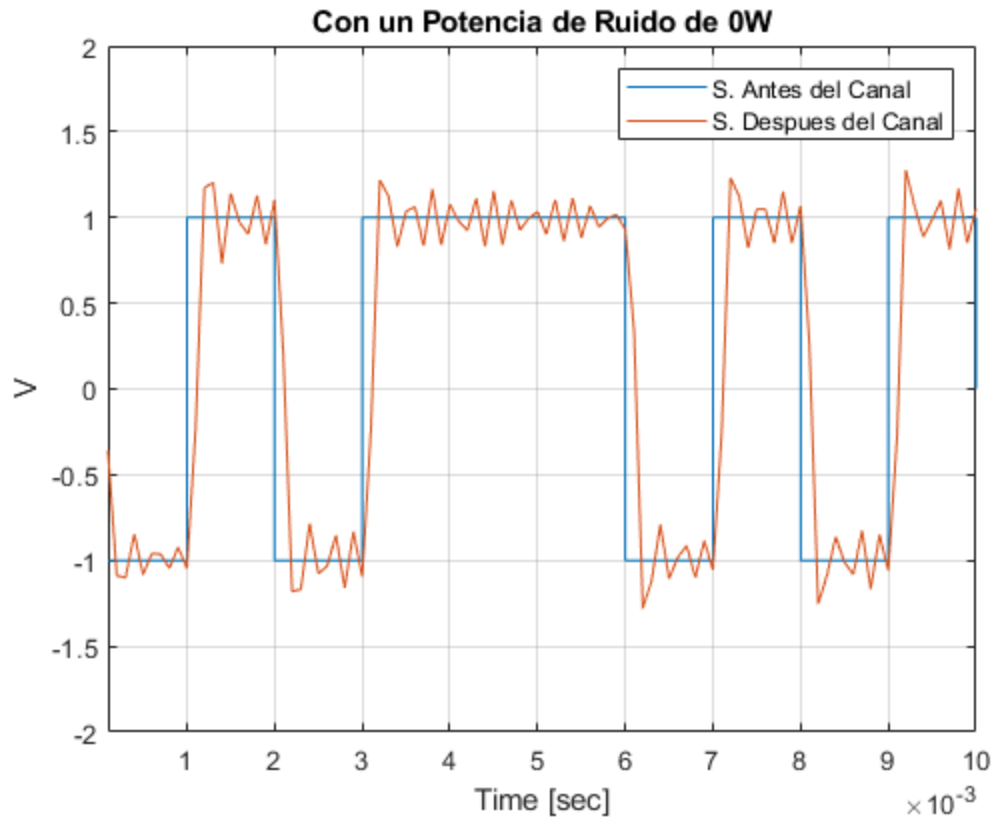
Warning: `AXIS('STATE')` is obsolete and will be eliminated in future versions.
Use `GET(GCA,...)` instead.

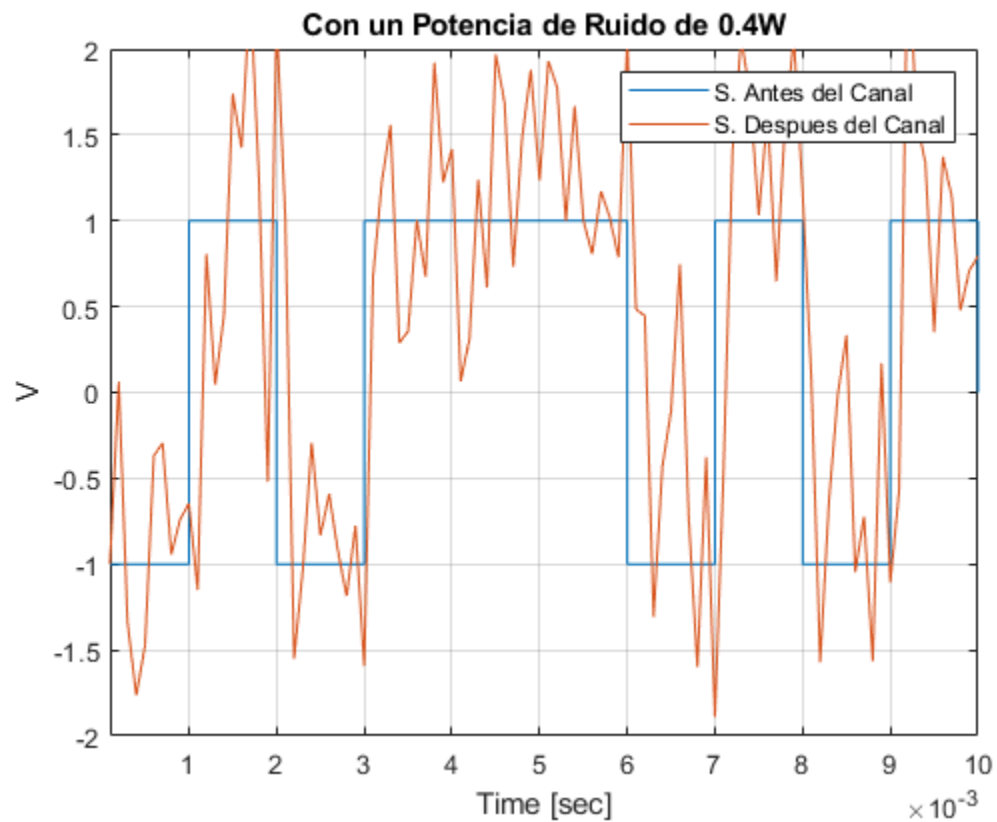
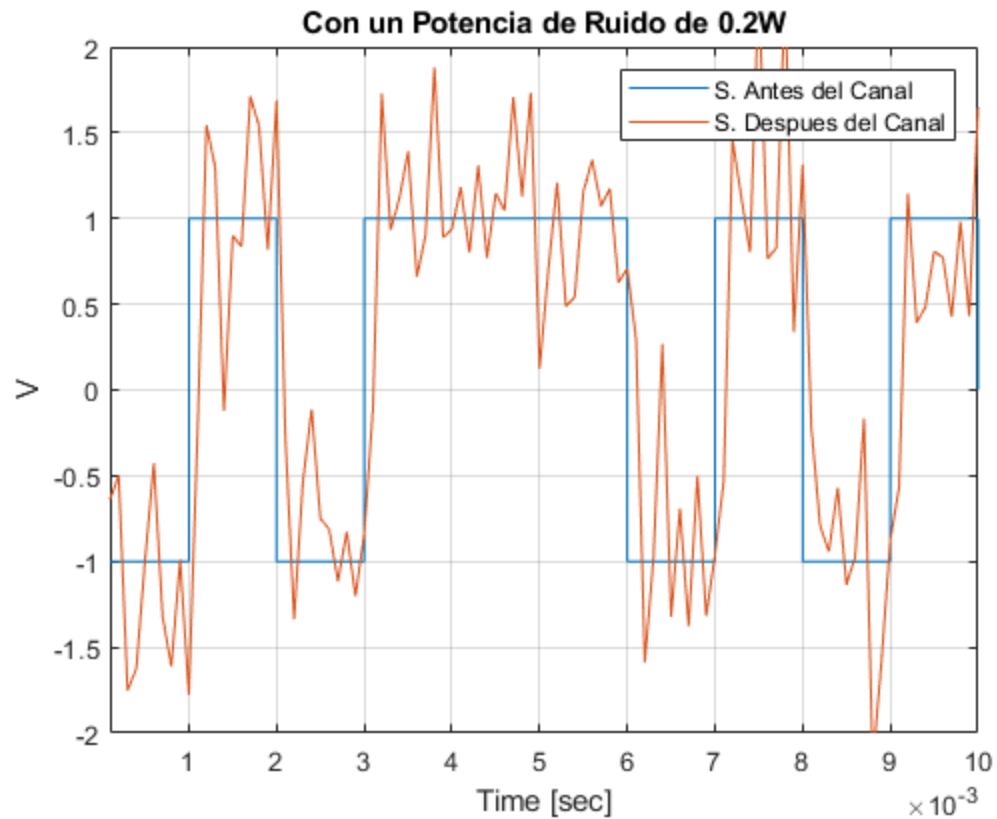
Warning: `AXIS('STATE')` is obsolete and will be eliminated in future versions.
Use `GET(GCA,...)` instead.

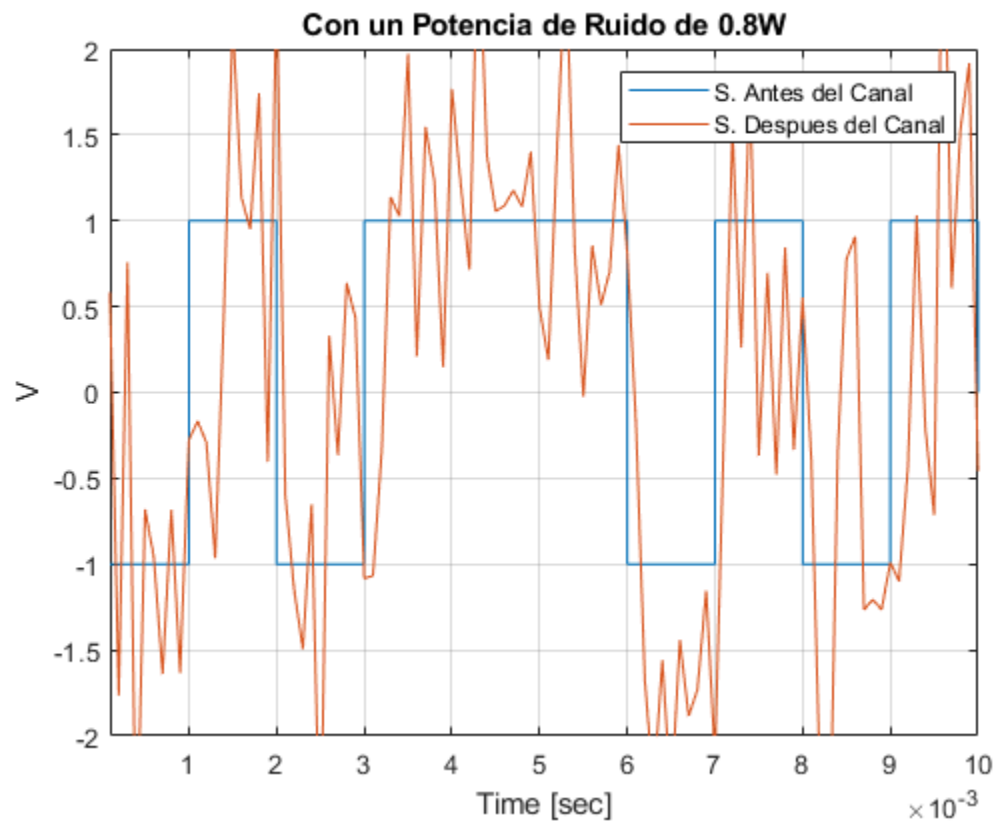
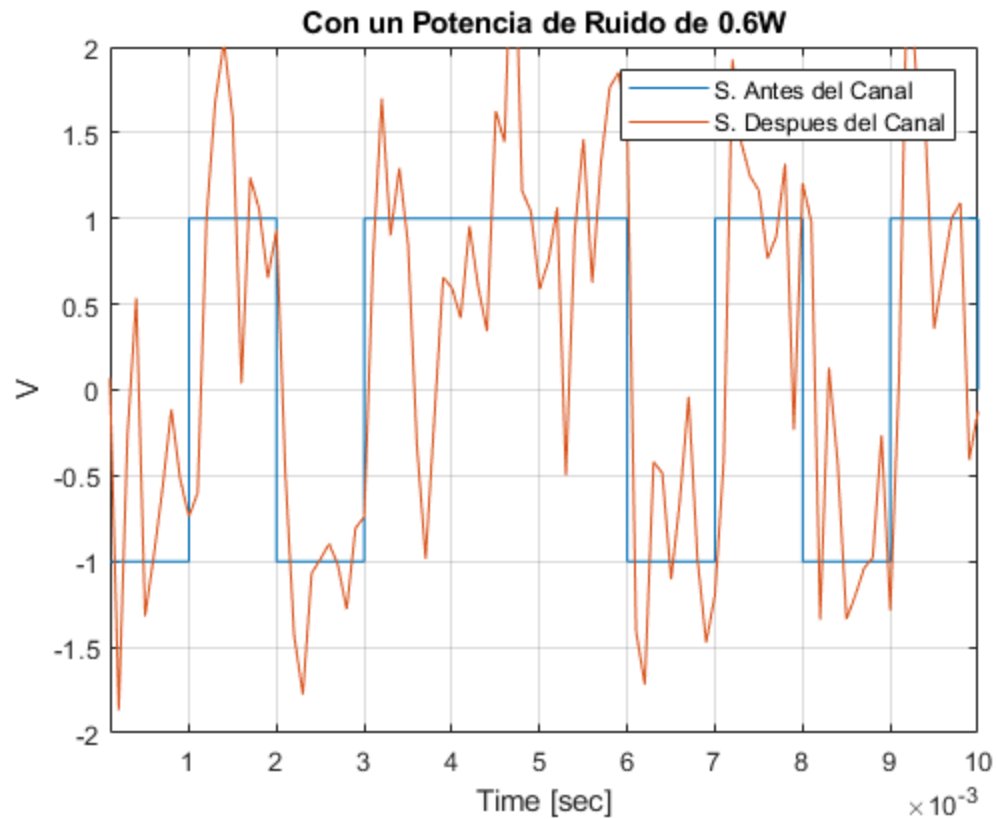
Warning: `AXIS('STATE')` is obsolete and will be eliminated in future versions.
Use `GET(GCA,...)` instead.

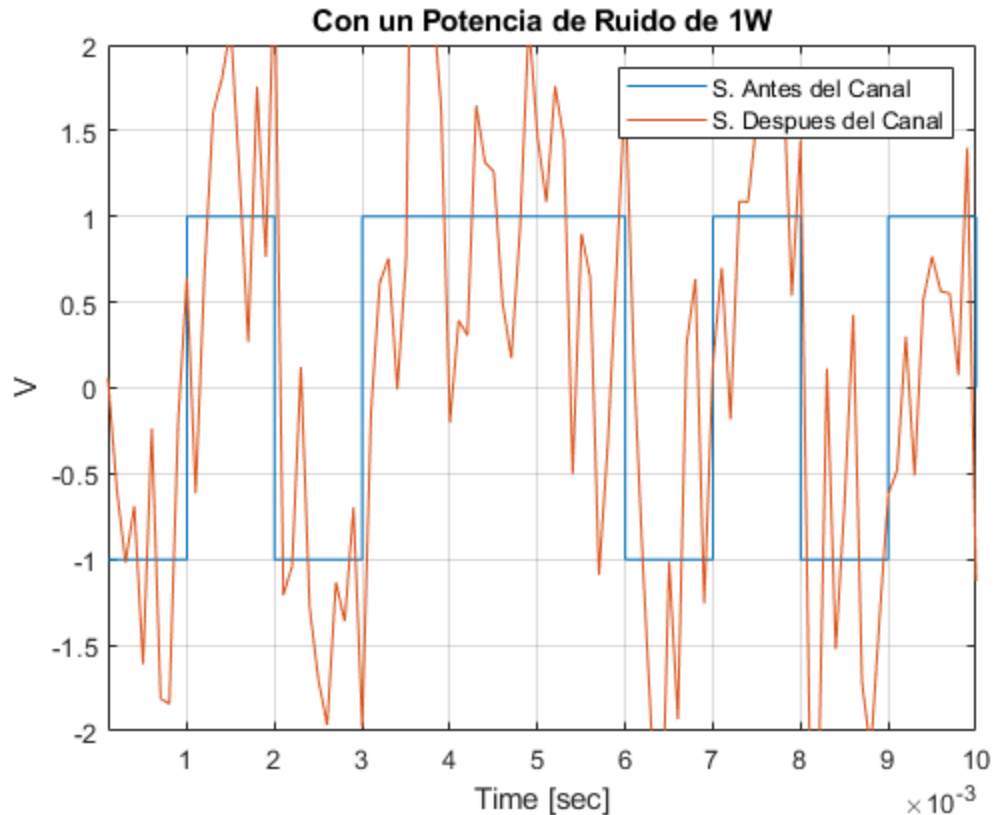
Warning: `AXIS('STATE')` is obsolete and will be eliminated in future versions.
Use `GET(GCA,...)` instead.

Warning: `AXIS('STATE')` is obsolete and will be eliminated in future versions.
Use `GET(GCA,...)` instead.









Ejercicio 1.4

```
%Generamos la secuencia y establecemos Rb
nbits=1000; %Tamaño de la secuencia binaria
b=round(rand(1,nbits));
Rb=1000;

%NRZ Polar
x=wave_gen(b, 'polar_nrz', Rb);

%Generamos el canal
ancho_banda=4500;
G=1;

for pot_ruido=0:0.2:1.4

    y=channel(x,G,pot_ruido,ancho_banda);

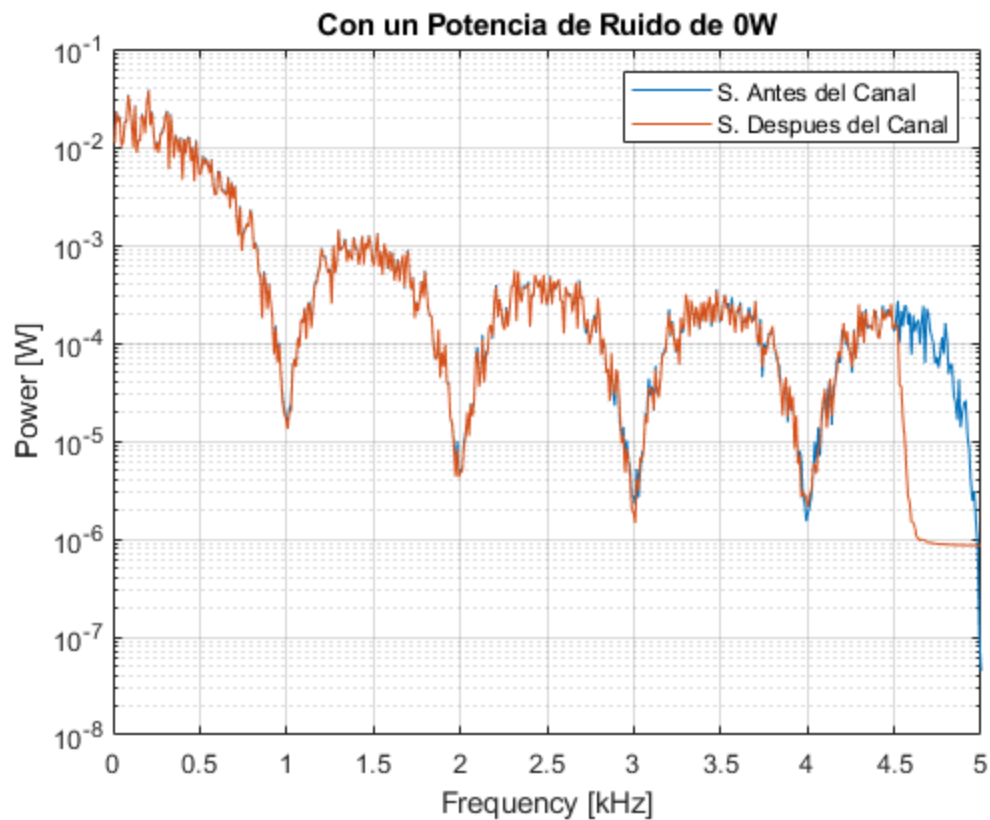
    figure,psd(x);hold on;psd(y);
    title(['Con un Potencia de Ruido de ' num2str(pot_ruido) 'W'])
    legend('S. Antes del Canal','S. Despues del Canal');
end

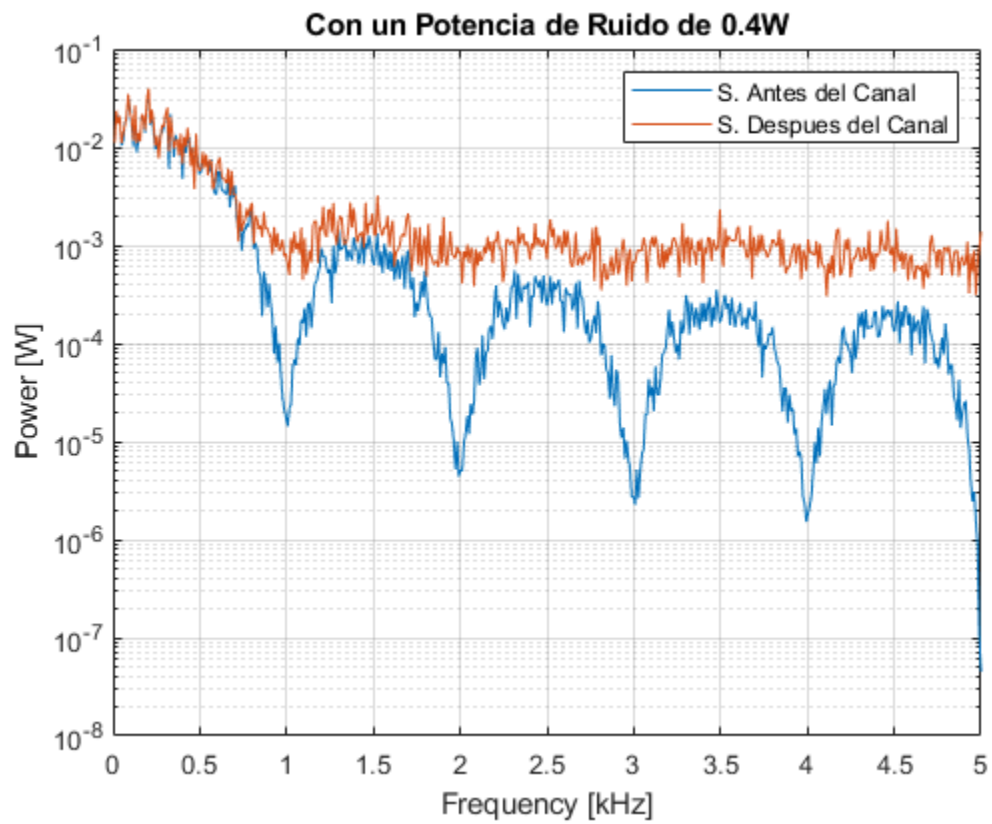
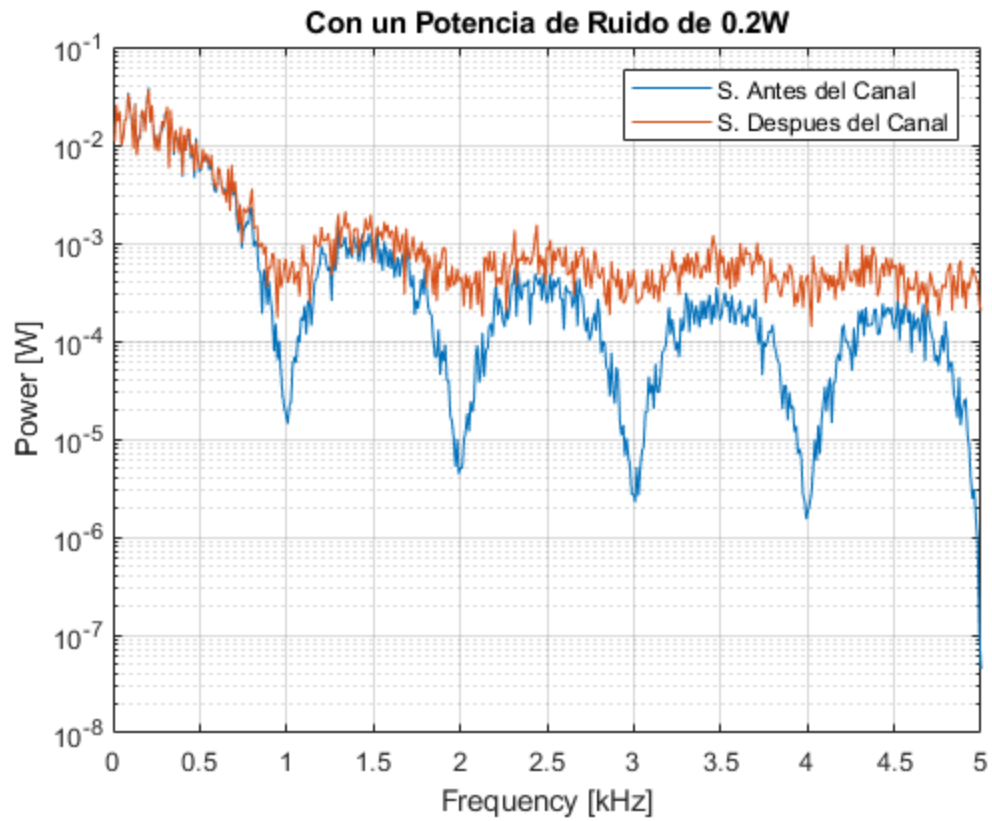
%%A
%Las que sufren un gran cambio son las frecuencias mas "altas", que
```

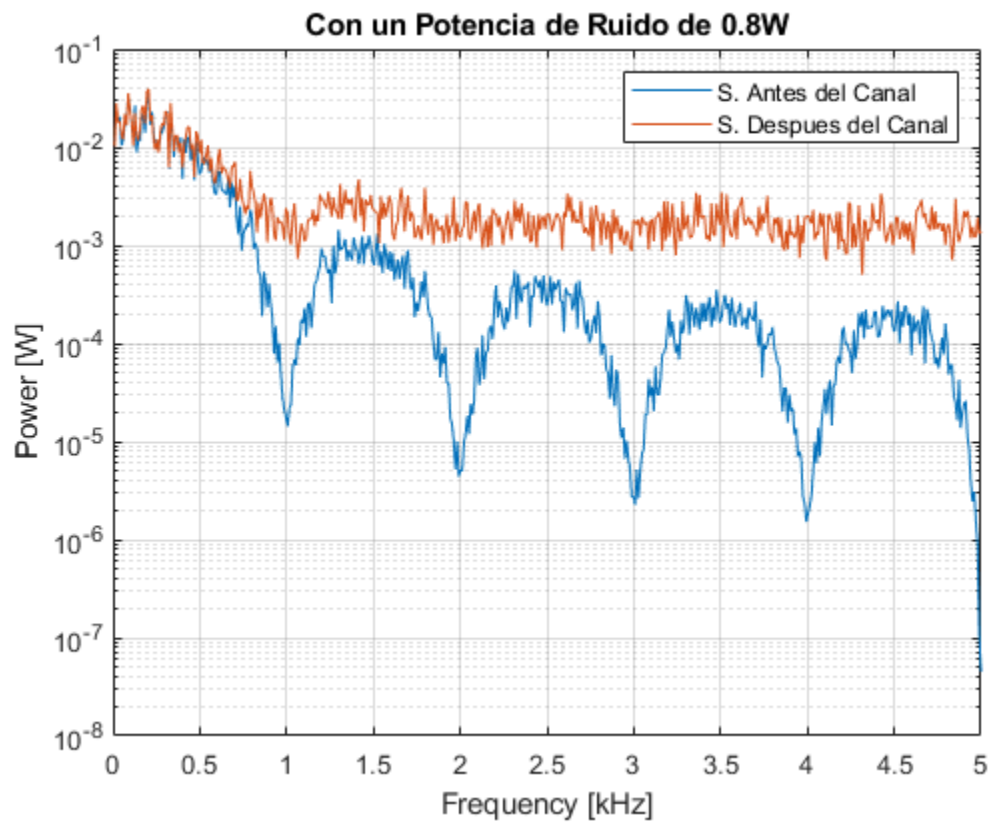
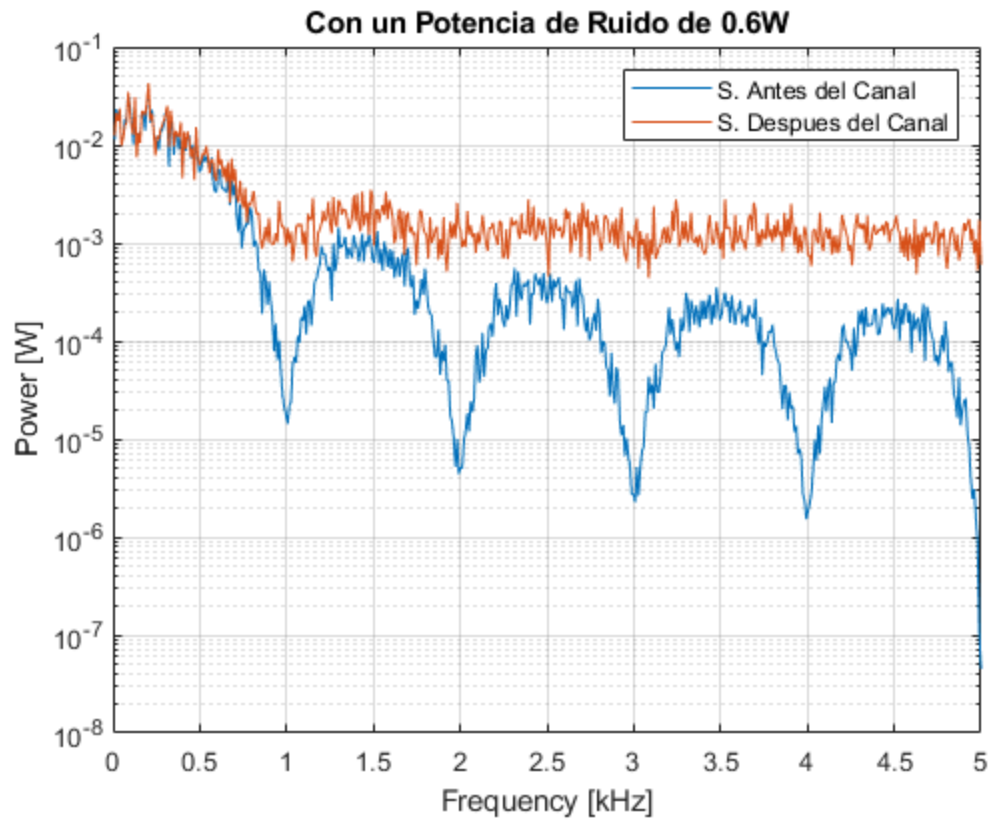
%aumenta su potencia, esto es debido a la limitacion del ancho
%de banda del canal, ya que el ancho de banda de una señal digital es
%infinito, al limitarlo aparecen estos cambios.

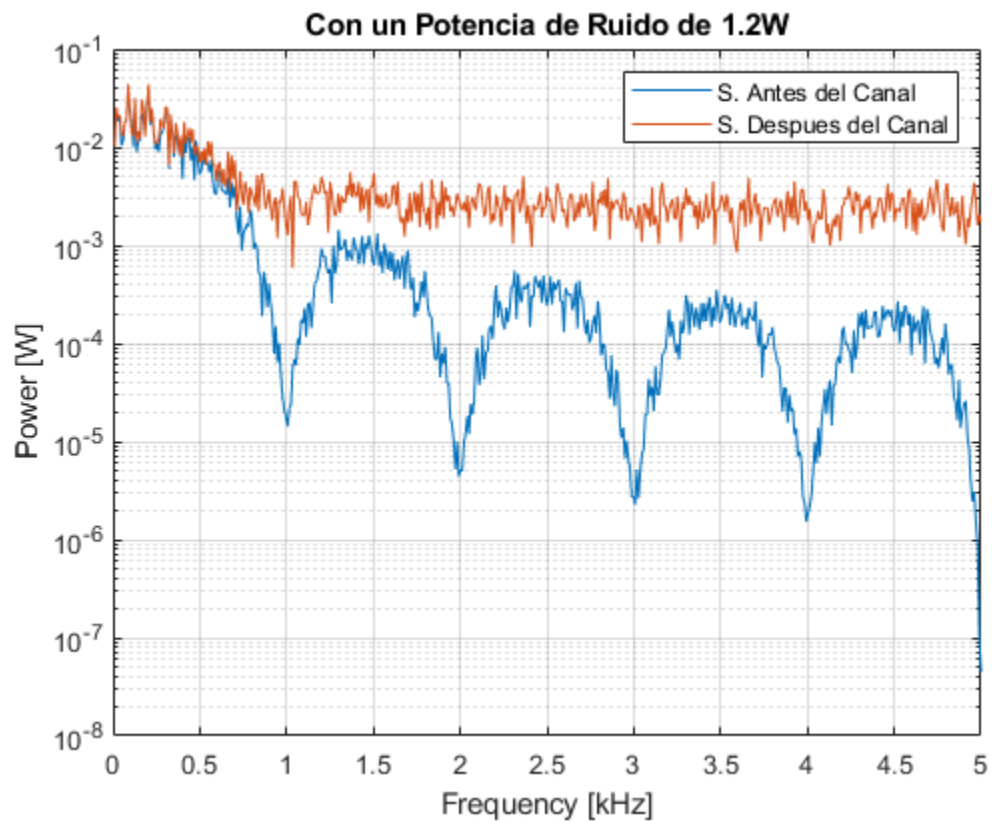
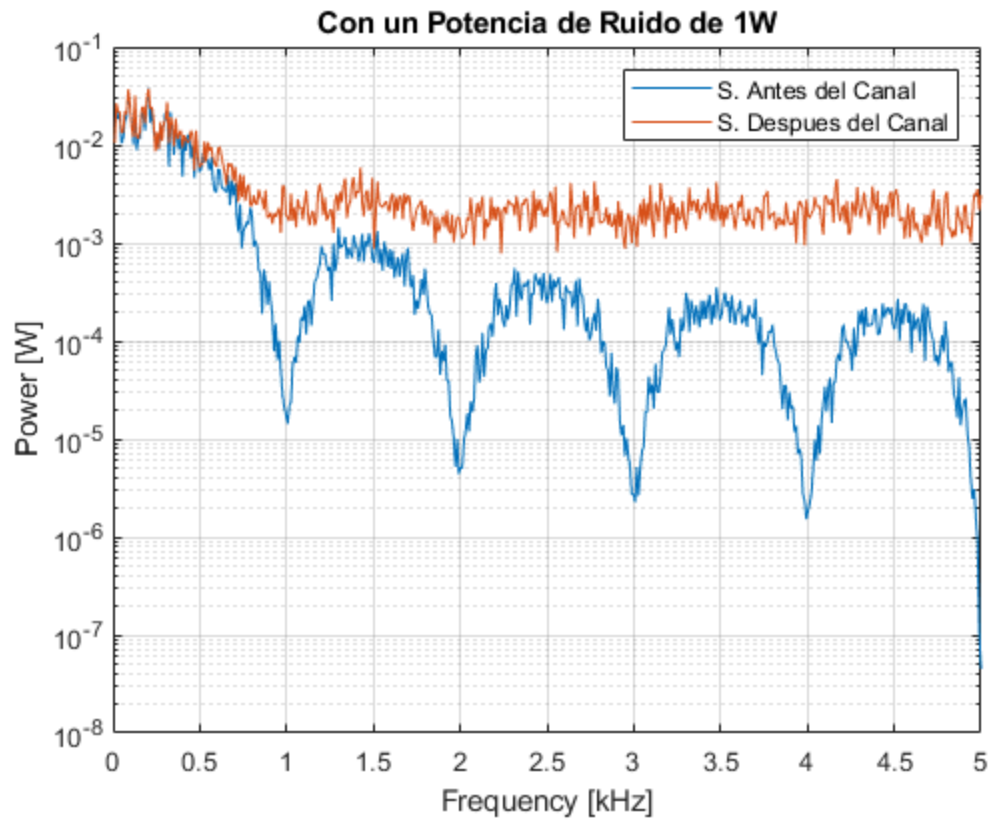
%%B

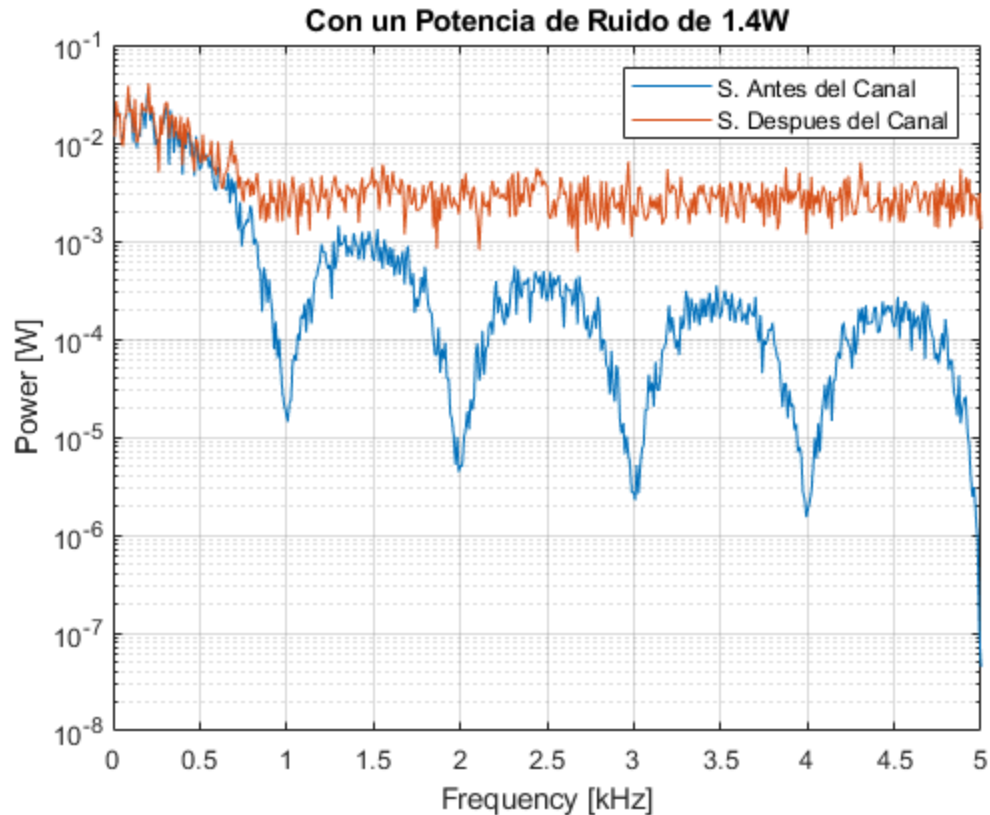
%A partir de 1W de potencia de ruido ya es imposible distinguir las
%frecuencias mas altas y el lobulo de mas energia practicamente ya
tiene la
%misma energia que el resto de frecuencias, por lo que es dificil
recuperar
%la señal correctamente











Ejercicio 1.5

```
%Generamos la secuencia y establecemos Rb
nbits=20; %Tamaño de la secuencia binaria
b=round(rand(1,nbits));
Rb=1000;

%NRZ Polar
x=wave_gen(b, 'polar_nrz', Rb);

%Generamos el canal
G=1;
pot_ruido=0;

for ancho_banda=[4500 3000 2000 1000 500 250]

    y=channel(x,G,pot_ruido,ancho_banda);

    figure,waveplot(x);hold on;waveplot(y);
    title(['Con un Ancho de banda de ' num2str(ancho_banda) 'Hz'])
    legend('S. Antes del Canal','S. Despues del Canal');
end

%%A
```

%A partir de un ancho de banda de 2000Hz aparece un desfase de $\pi/2$,
aunque
%la distorsion no hace que la se\u00f1al sea indescifrable, a partir de un
ancho
%de banda de 500Hz, la se\u00f1al no se parece en nada a la transmitida y
sera
%irrecuperable.
%Segun Nyquist el ancho de banda del canal ideal es de $W=1/(2T)$, que
en
%nuestro caso, la frecuencia de muestreo es de 10000, $T=1/f$,
 $W=f/2=5000\text{Hz}$
%que es justo 10 veces mas grande que 500Hz (donde aparecen los
problemas),
%esto es debido a que la frecuencia de muestreo es 10 el regimen
binario,
%si el ancho de banda es igual a la mitad de la frecuencia del regimen
%binario, aparecen los problemas

Warning: AXIS('STATE') is obsolete and will be eliminated in future versions.

Use GET(GCA,...) instead.

Warning: AXIS('STATE') is obsolete and will be eliminated in future versions.

Use GET(GCA,...) instead.

Warning: AXIS('STATE') is obsolete and will be eliminated in future versions.

Use GET(GCA,...) instead.

Warning: AXIS('STATE') is obsolete and will be eliminated in future versions.

Use GET(GCA,...) instead.

Warning: AXIS('STATE') is obsolete and will be eliminated in future versions.

Use GET(GCA,...) instead.

Warning: AXIS('STATE') is obsolete and will be eliminated in future versions.

Use GET(GCA,...) instead.

Warning: AXIS('STATE') is obsolete and will be eliminated in future versions.

Use GET(GCA,...) instead.

Warning: AXIS('STATE') is obsolete and will be eliminated in future versions.

Use GET(GCA,...) instead.

Warning: AXIS('STATE') is obsolete and will be eliminated in future versions.

Use GET(GCA,...) instead.

Warning: AXIS('STATE') is obsolete and will be eliminated in future versions.

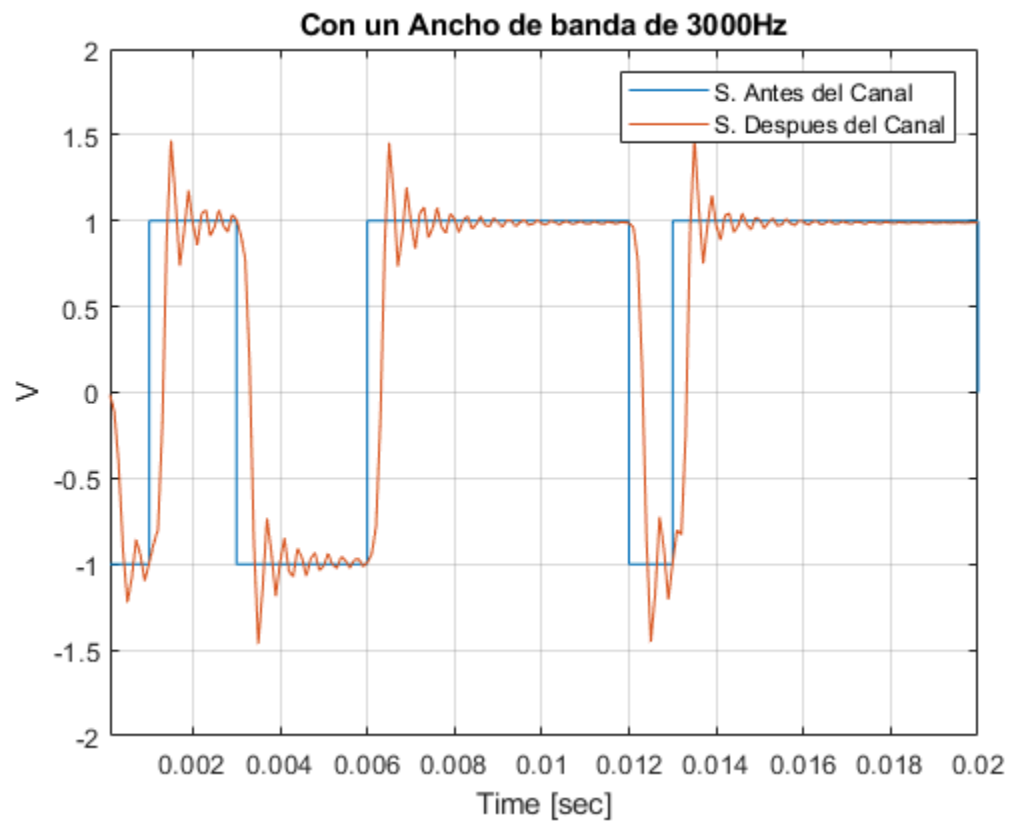
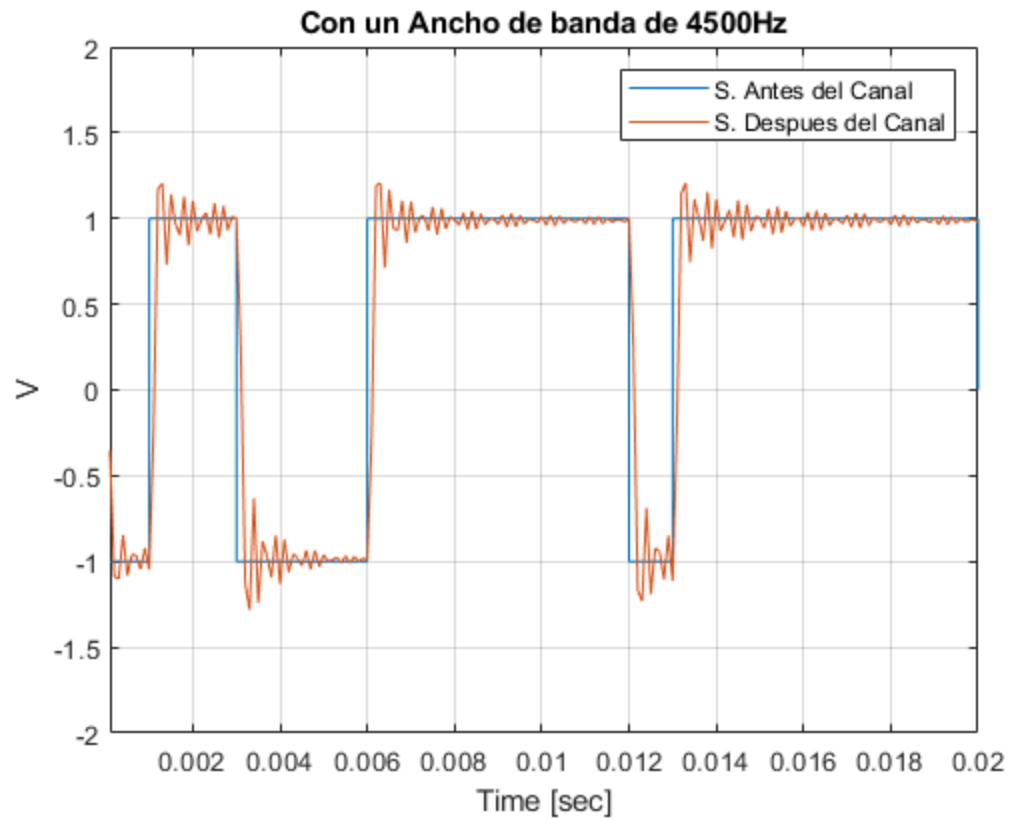
Use GET(GCA,...) instead.

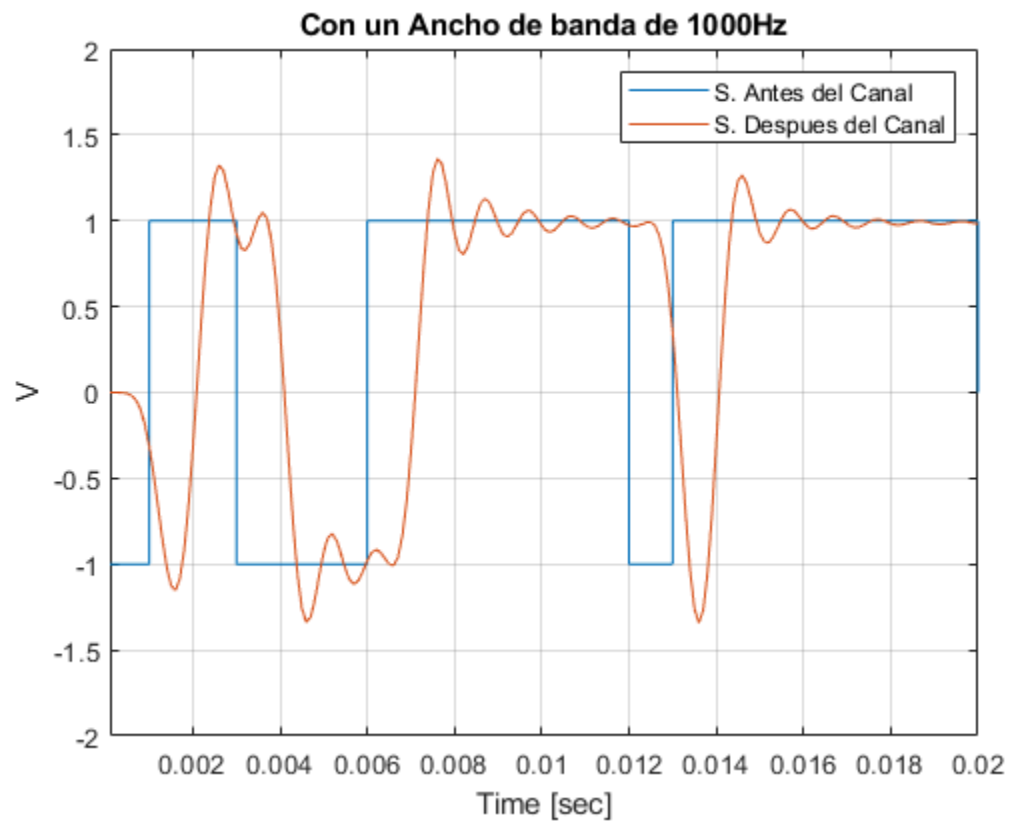
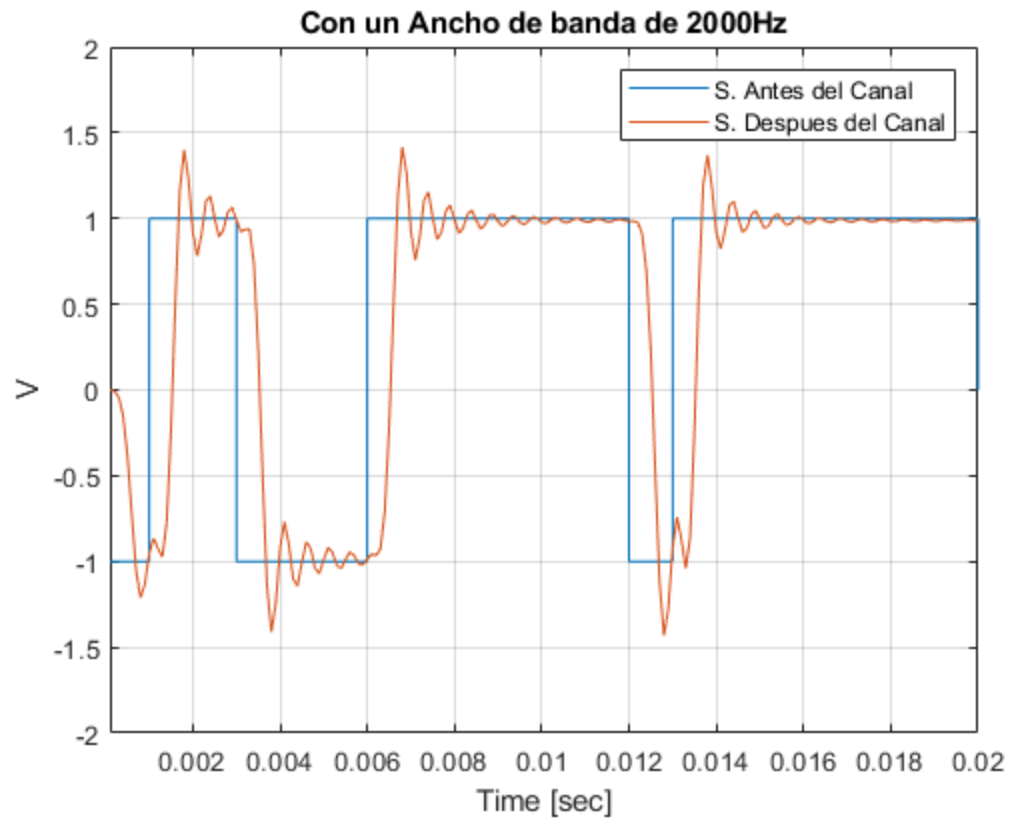
Warning: AXIS('STATE') is obsolete and will be eliminated in future versions.

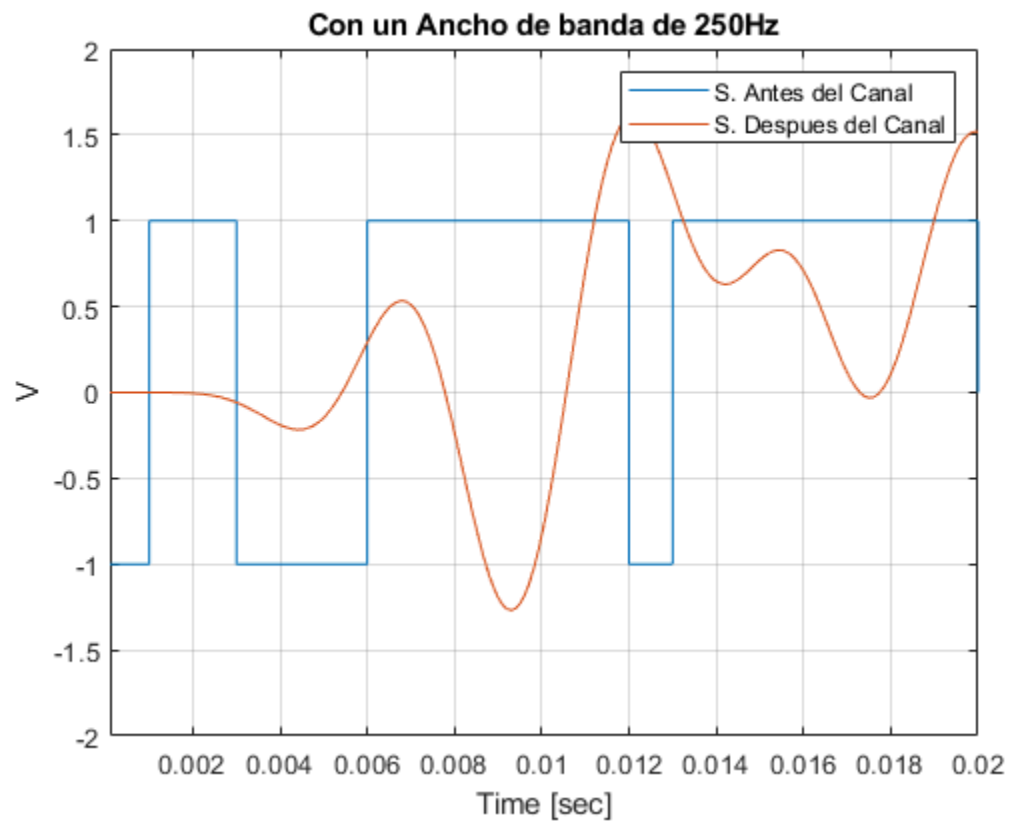
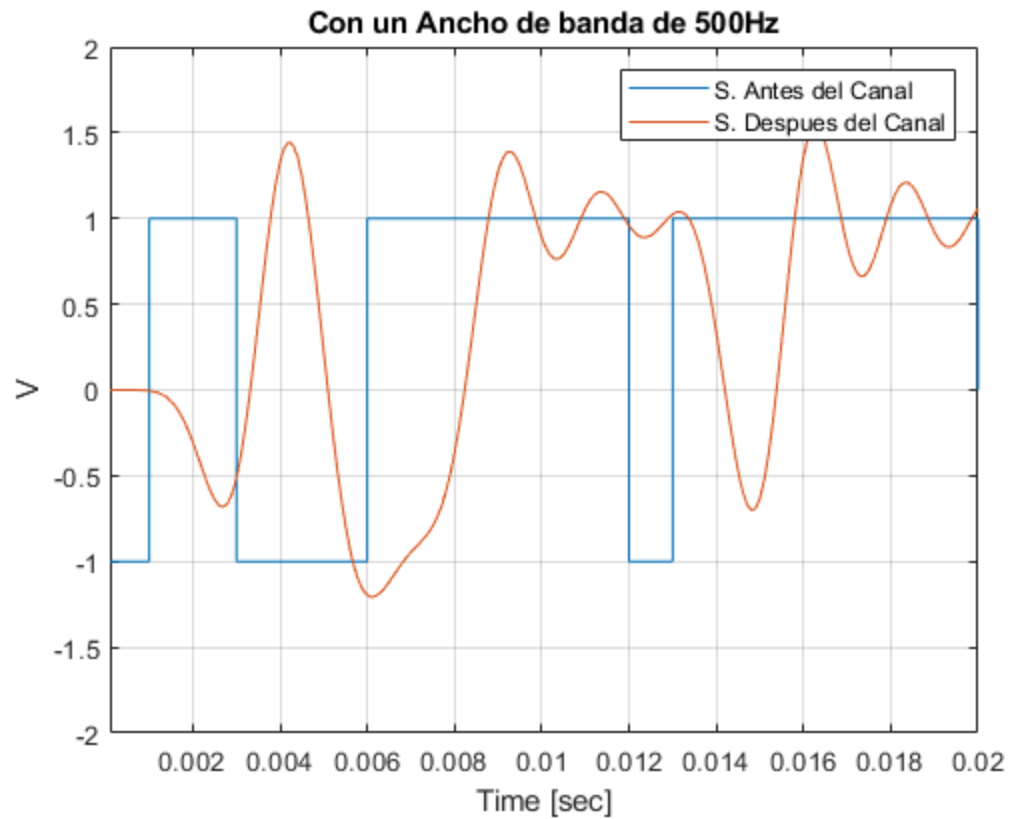
Use GET(GCA,...) instead.

Warning: AXIS('STATE') is obsolete and will be eliminated in future versions.

Use GET(GCA,...) instead.







Ejercicio 1.6

```
%Generamos la secuencia y establecemos Rb
nbits=100; %Tamaño de la secuencia binaria
b=round(rand(1,nbits));
Rb=1000;

%NRZ Polar
x=wave_gen(b, 'polar_nrz', Rb);

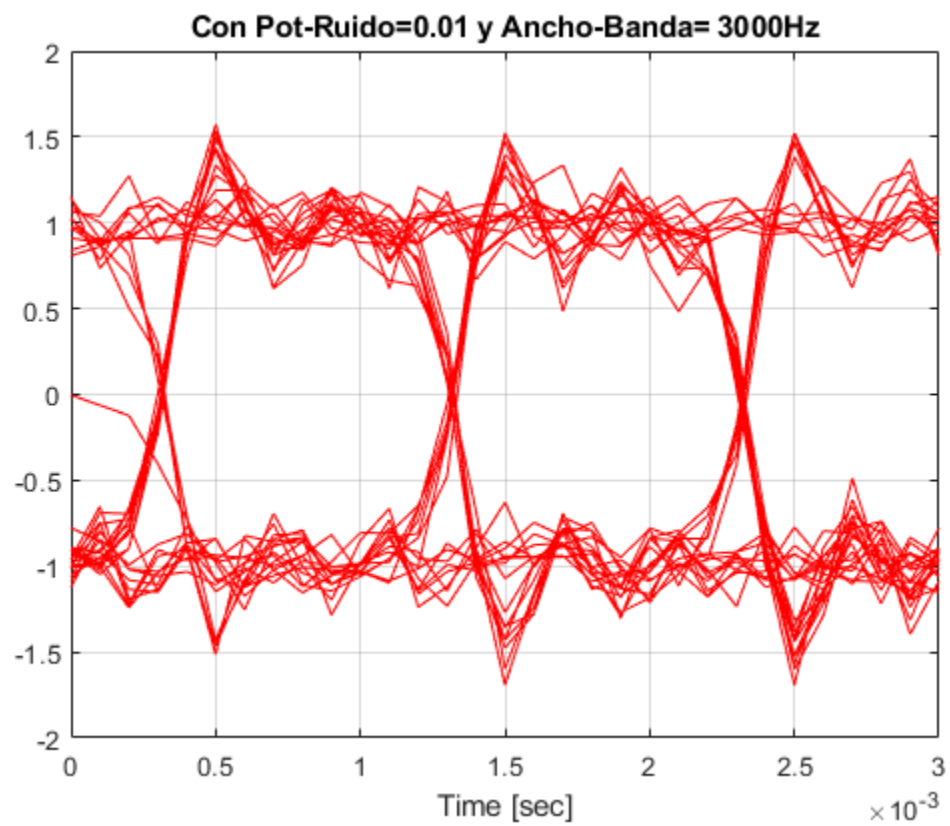
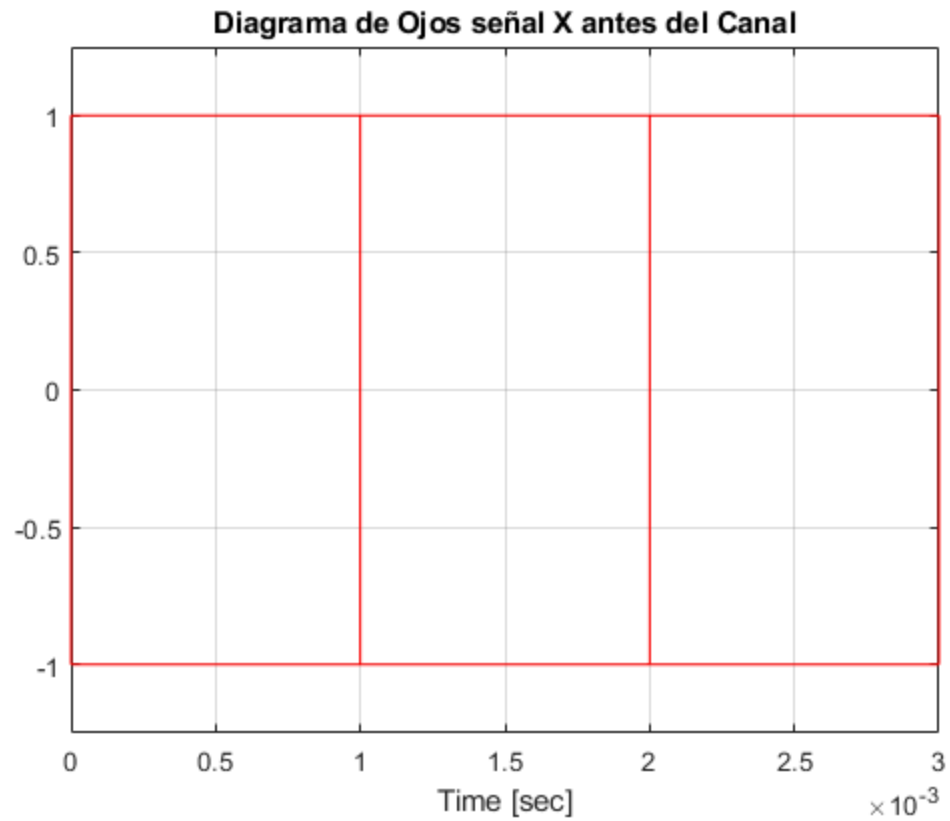
figure, eye_diag(x); title('Diagrama de Ojos señal X antes del Canal');

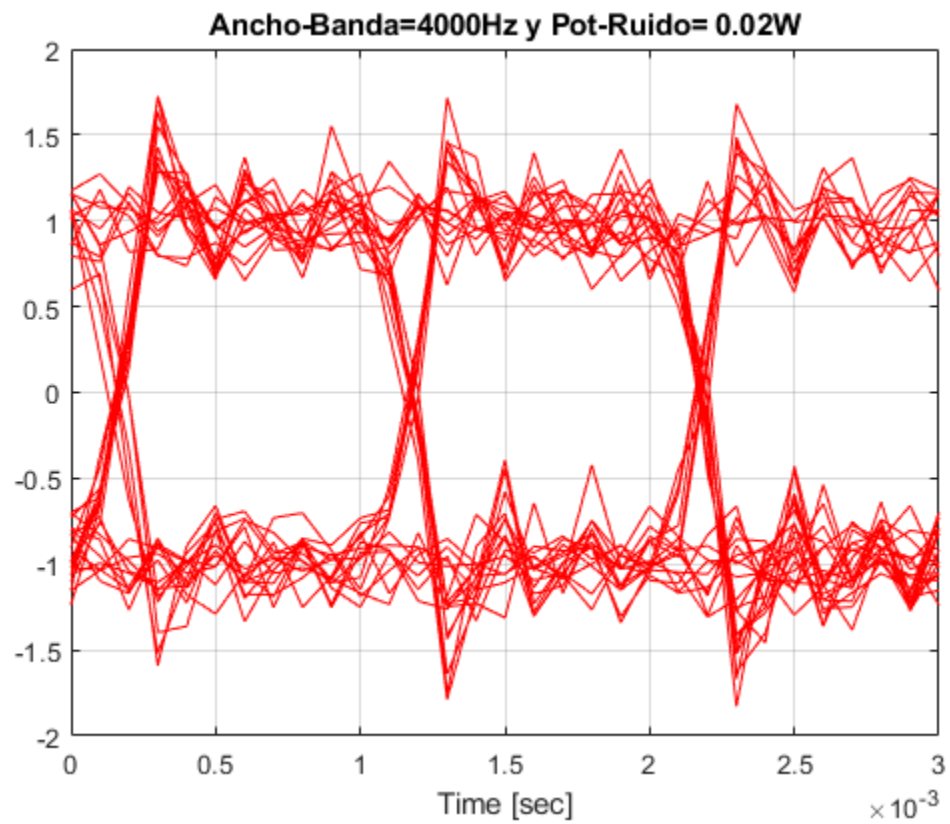
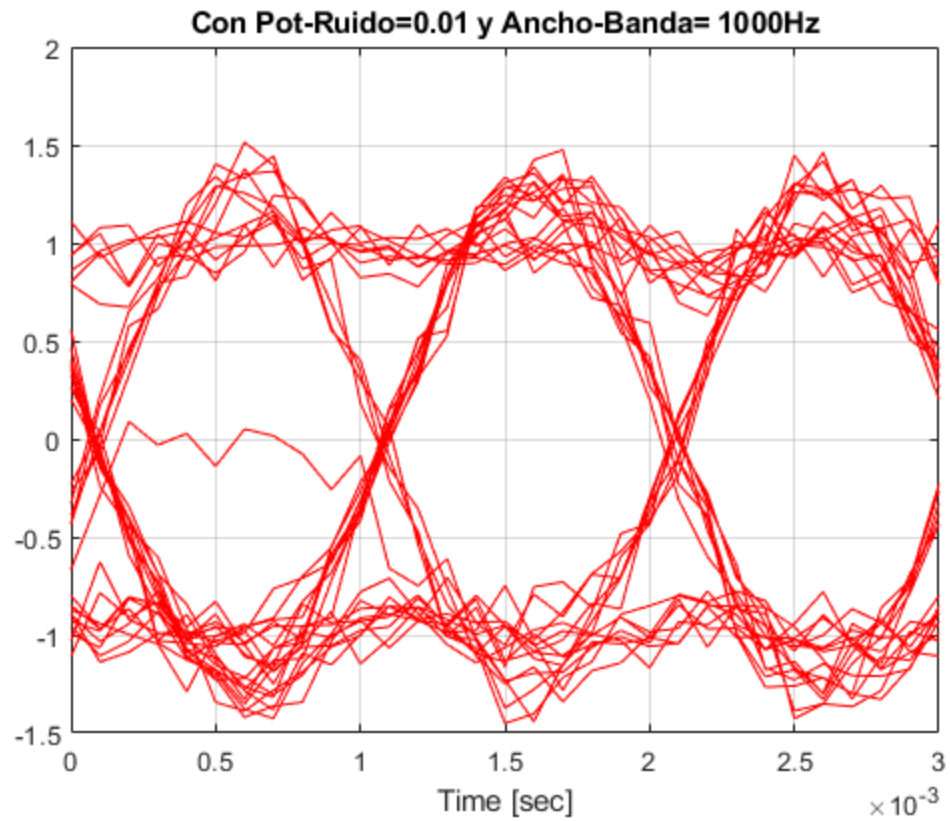
%Generamos el canal
G=1;
pot_ruido=0.01;

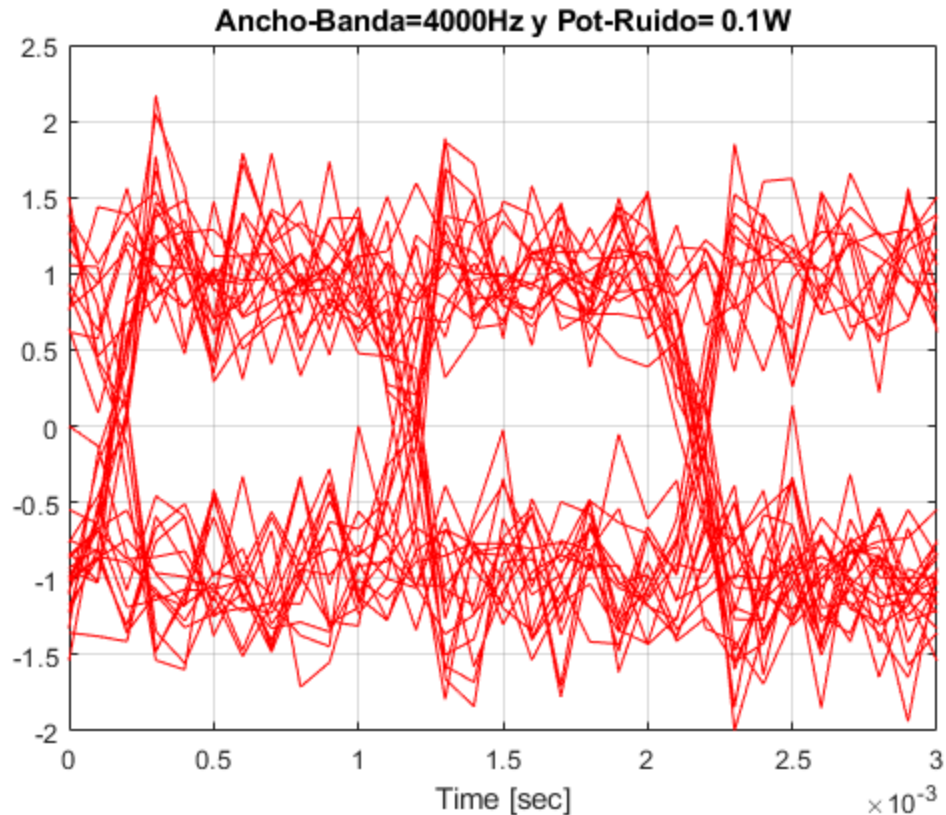
for ancho_banda=[3000 1000]
    y=channel(x,G,pot_ruido,ancho_banda);
    figure, eye_diag(y);
    title(['Con Pot-Ruido=0.01 y Ancho-Banda= '
        num2str(ancho_banda) ' Hz'])
end

%Generamos el canal
ancho_banda=4000;
G=1;
for pot_ruido=[0.02 0.1]
    y=channel(x,G,pot_ruido,ancho_banda);
    figure, eye_diag(y);
    title(['Ancho-Banda=4000Hz y Pot-Ruido= ' num2str(pot_ruido) ' W'])
end

%%Tabla
%0.01W-3000Hz --> IOM=0.5 IM=1 MR=0.8
%0.01W-1000Hz --> IOM=0.5 IM=1 MR=0.8
%0.02W-4000Hz --> IOM=1 IM=2 MR=0.7
%0.10W-4000Hz --> IOM=1 IM=2 MR=0.2
```







Ejercicio 1.7

```
%Generamos la secuencia y establecemos Rb
nbits=1000; %Tamaño de la secuencia binaria
b=round(rand(1,nbits));
Rb=1000;

%NRZ Polar
for NYQUIST_ALPHA=[0 0.5 1]
    a=NYQUIST_ALPHA;

    x=wave_gen(b, 'polar_nrz', Rb);
    y=wave_gen(b, 'nyquist', Rb);

    figure, waveplot(x(1:100)); hold on; waveplot(y(1:100));
    legend(['NRZ Polar con alpha=' num2str(a)], ['Nyquist con alpha='
num2str(a)]);

    figure, psd(x); hold on; psd(y);
    legend(['NRZ Polar con alpha=' num2str(a)], ['Nyquist con alpha='
num2str(a)]);

end

%%A
```

```
%alpha=0    --> ancho de banda= 500Hz  
%alpha=0.5  --> ancho de banda= 750Hz  
%alpha=1    --> ancho de banda= 1000Hz
```

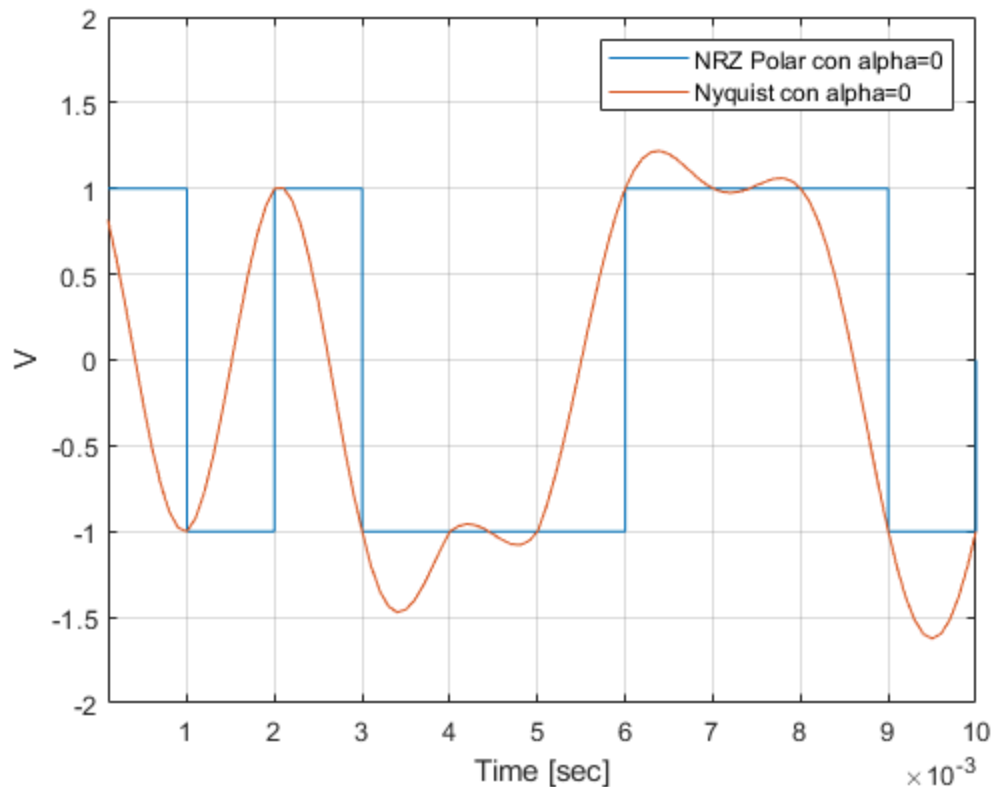
*Warning: AXIS('STATE') is obsolete and will be eliminated in future versions.
Use GET(GCA,...) instead.*

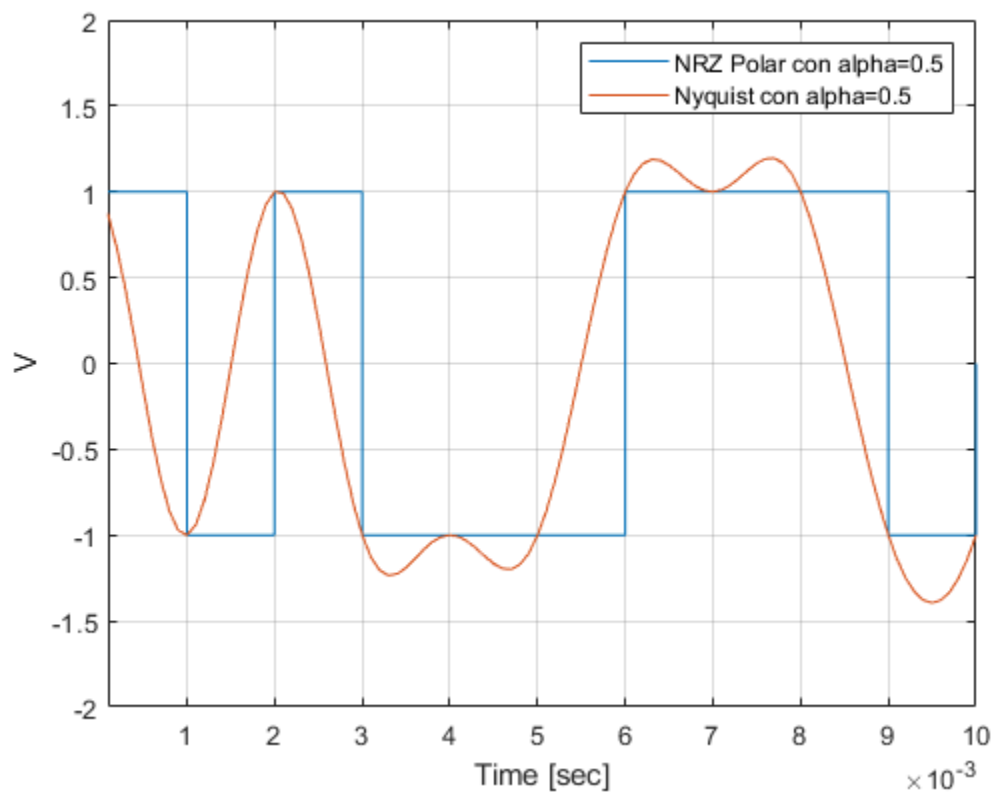
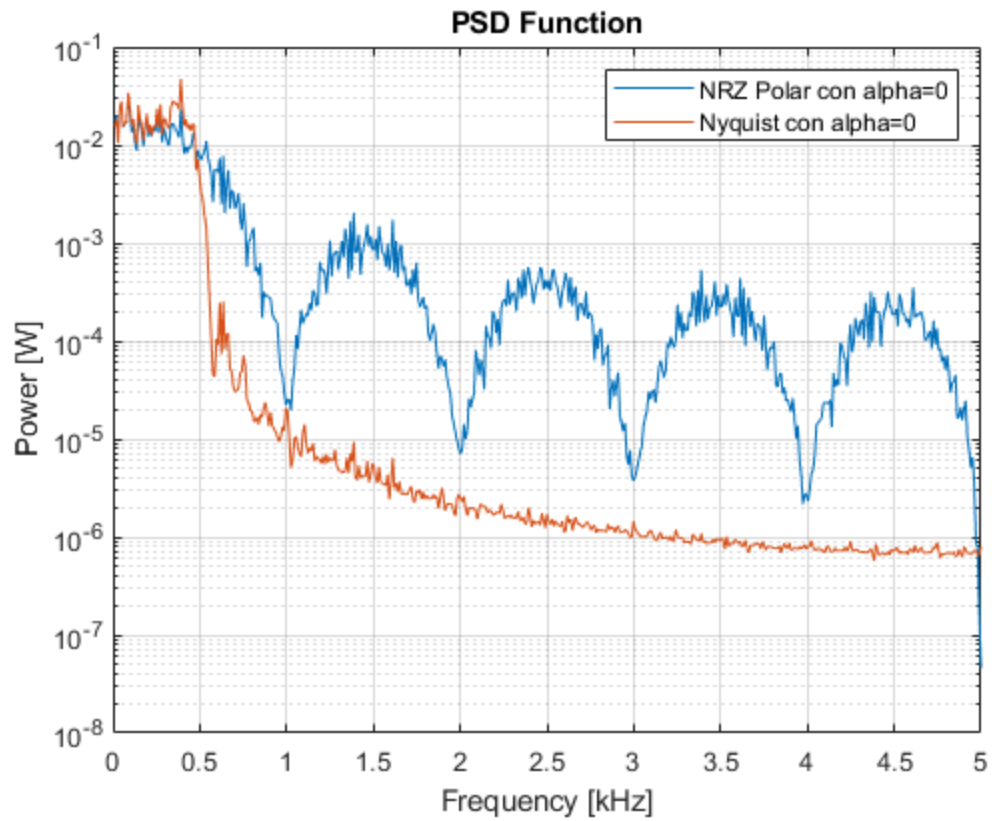
*Warning: AXIS('STATE') is obsolete and will be eliminated in future versions.
Use GET(GCA,...) instead.*

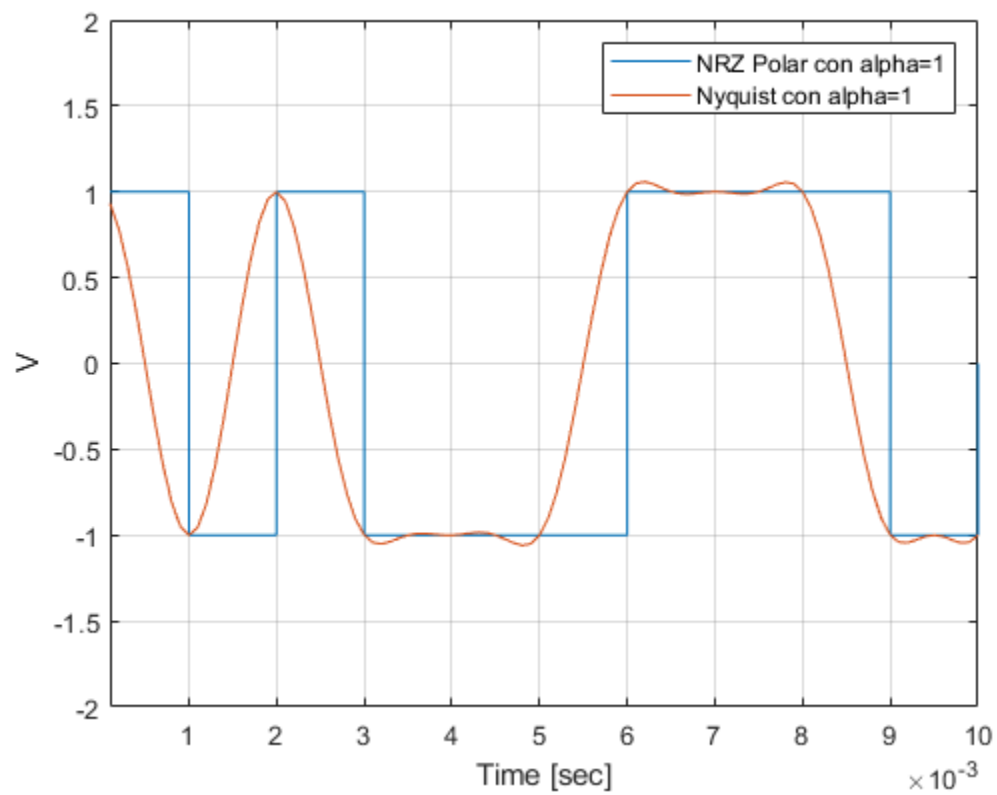
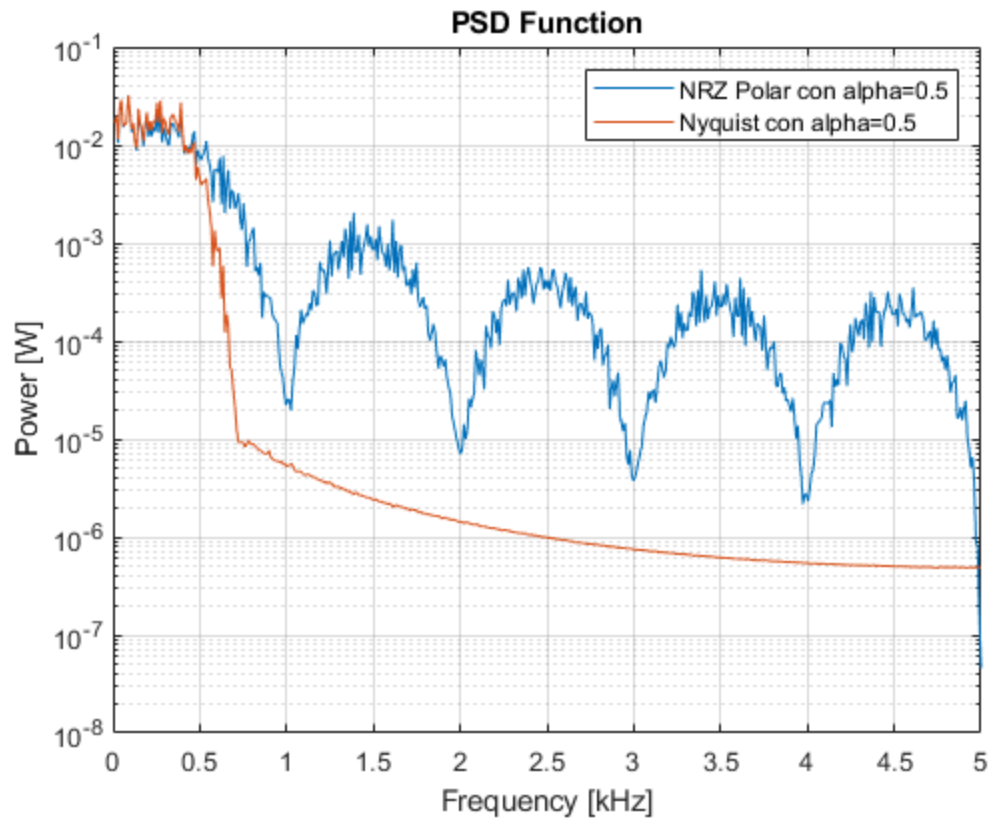
*Warning: AXIS('STATE') is obsolete and will be eliminated in future versions.
Use GET(GCA,...) instead.*

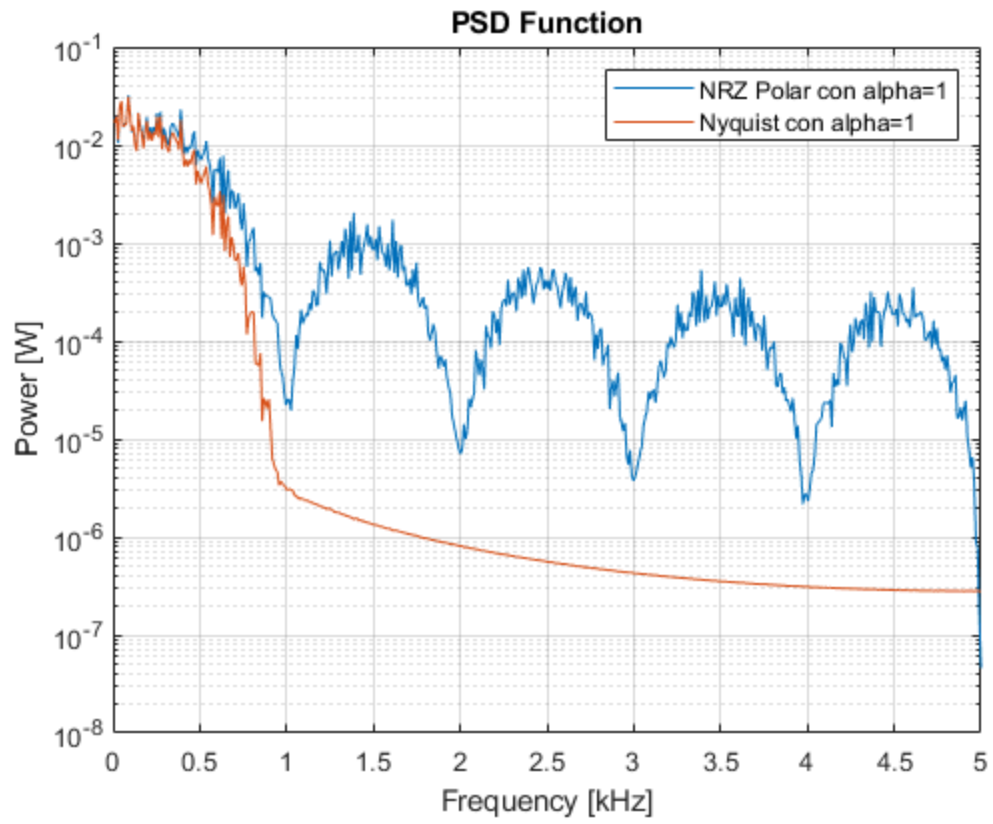
*Warning: AXIS('STATE') is obsolete and will be eliminated in future versions.
Use GET(GCA,...) instead.*

*Warning: AXIS('STATE') is obsolete and will be eliminated in future versions.
Use GET(GCA,...) instead.*









Published with MATLAB® R2018b