

Ejercicio 2: Utilizando la imagen de la Lena de la Tarea 6, genere el código de línea polar NRZ usando pulsos base rectangular y “half-sine”, con mp=10 y Fs=96 KHz. Potencia de Tx igual a 1 Watt. Aplique el “Eye pattern” a la señal transmitida:

```
load lena512.mat
lenarec = lena512(252:296,318:362);% Crop the image to a 45 x 45 vector
figure(1);
imshow(uint8(lenarec))
title("Lena's eye");
```

Lena's eye



```
b = de2bi(lenarec,8,'left-msb'); % Convert the samples from uint8 to binary
b = b';
bits = b(:);

bits2send = bits';
Fs= 96000;
mp= 10;
%No_bits= 8712;
Baud_rate= Fs/mp %Symbols per second
```

```
Baud_rate = 9600
```

```
Bit_rate=Baud_rate % bits/s
```

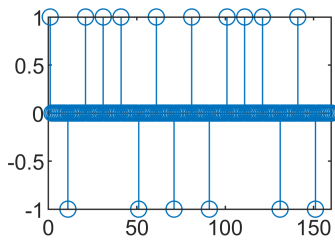
```
Bit_rate = 9600
```

```
Ts=1/Fs;
% The bit rate is Rb= Rs= Fs / mp, because 1 bit= 1 symbol and every symbol has mp
% samples per bit

n= 0:(mp-1);
w = pi/mp;
hs = sin(w*n);

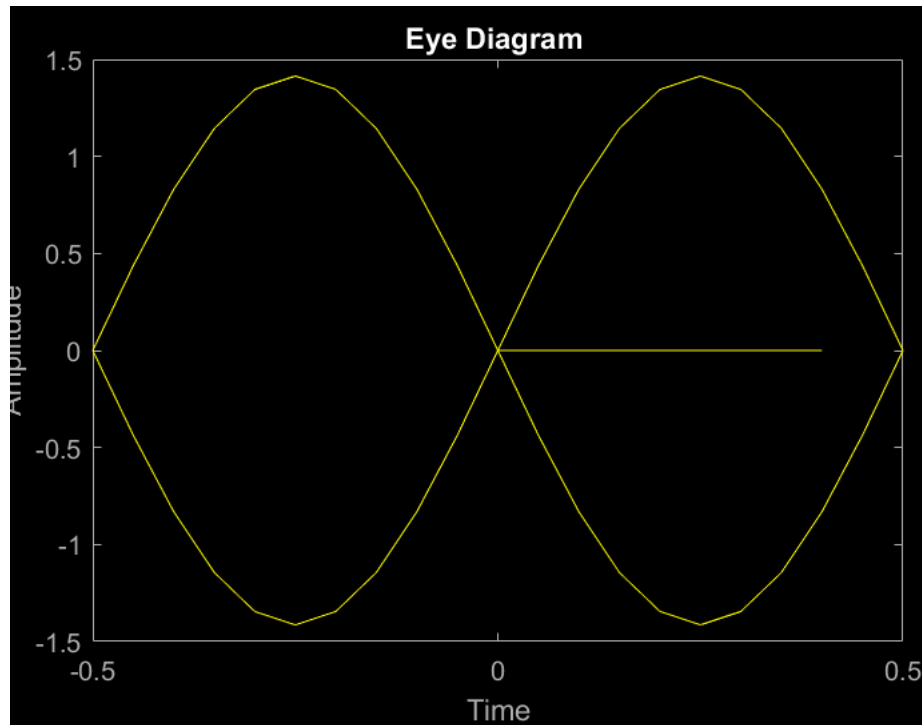
pnrz=ones(1,mp);

s1=bits;
s1(s1==0)=-1;
s=zeros(1,numel(s1)*mp);
s(1:mp:end)=s1; %Impulse train
stem(s(1:mp*16))
```

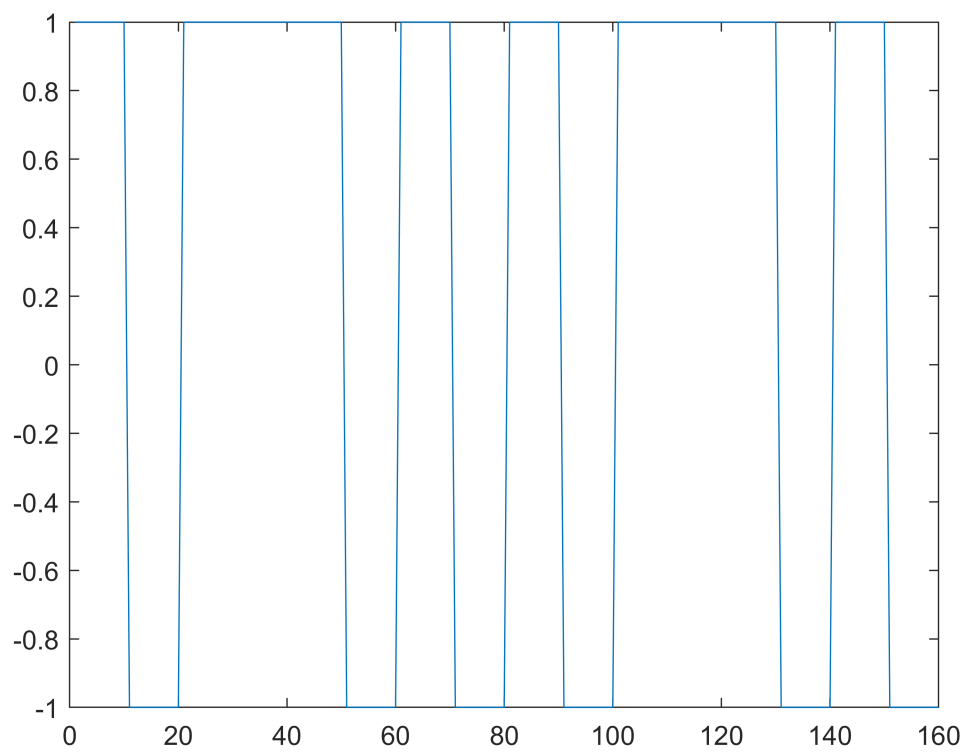


```
xPNRZ_hs=conv(hs,s); %Pulse half sine
xPNRZ=conv(pnrz,s); %Pulse rectangular

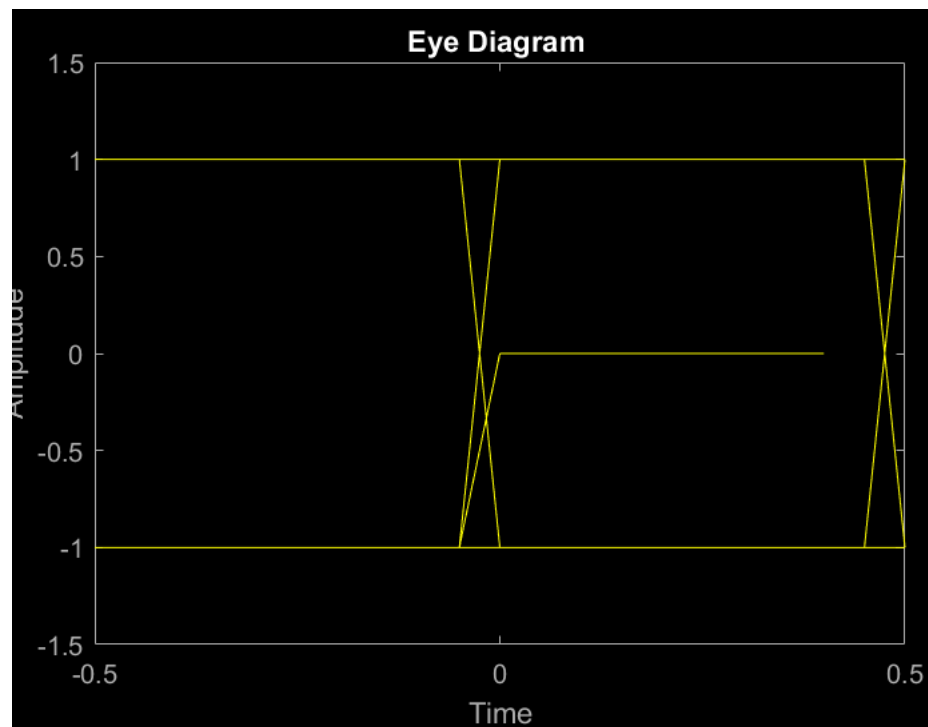
figure();
pow = sum(xPNRZ_hs.^2)/numel(xPNRZ_hs);
pow_Deseada = 1;
xPNRZ_hs = sqrt(pow_Deseada/pow)*xPNRZ_hs;
eyediagram(xPNRZ_hs,2*mp);
```



```
figure();
plot(xPNRZ(1:mp*16))
```



```
pow = sum(xPNRZ.^2)/numel(xPNRZ);
pow_Deseada = 1;
xPNRZ = sqrt(pow_Deseada/pow)*xPNRZ;
eyediagram(xPNRZ,2*mp);
```



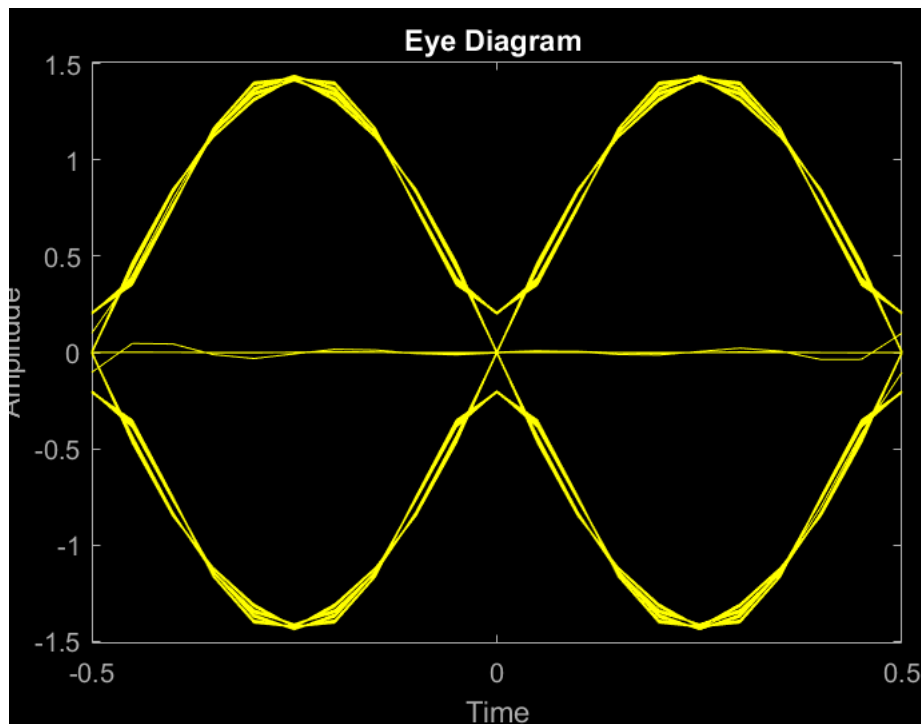
### ejercicio 3

Haga sus comentarios de acuerdo con lo que aprendió en los tutoriales.

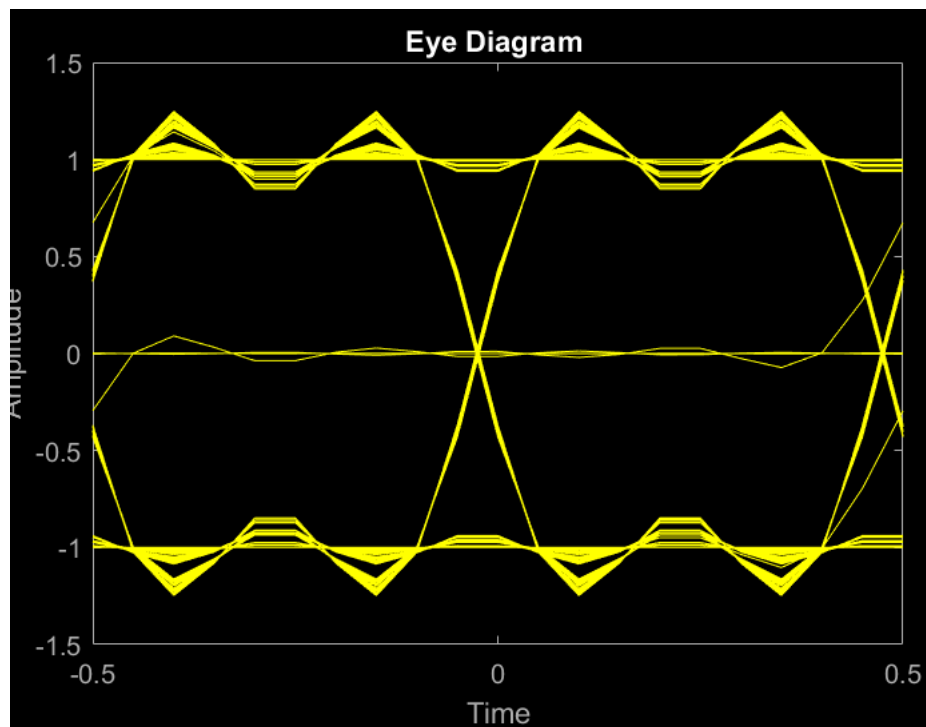
Ejercicio 3: Transmita este código de línea por el canal (usando como Modelo de canal el LPF (orden= 60) con  $F_c$  normalizada igual a  $0.4 \times \pi$ ). Observe el diagrama de ojo recibido y haga sus anotaciones de acuerdo con lo que aprendió en los tutoriales.

```
orden=60;      % Orden del Filtro
f=  [0 0.4 0.4 1]; % Vector de Frecuencias
m=  [1 1 0 0]; % Vector de Magnitudes
f1 = fir2(orden,f,m); % Coeficientes del Filtro usando FIR2( )
%fvtool(f1);

xPNRZ_filtrado_f1 = conv(xPNRZ_hs,f1);
eyediagram(xPNRZ_filtrado_f1,2*mp);
```



```
xPNRZ_filtrado_f2 = conv(xPNRZ,f1);
eyediagram(xPNRZ_filtrado_f2,2*mp);
```



## ejercicio 4

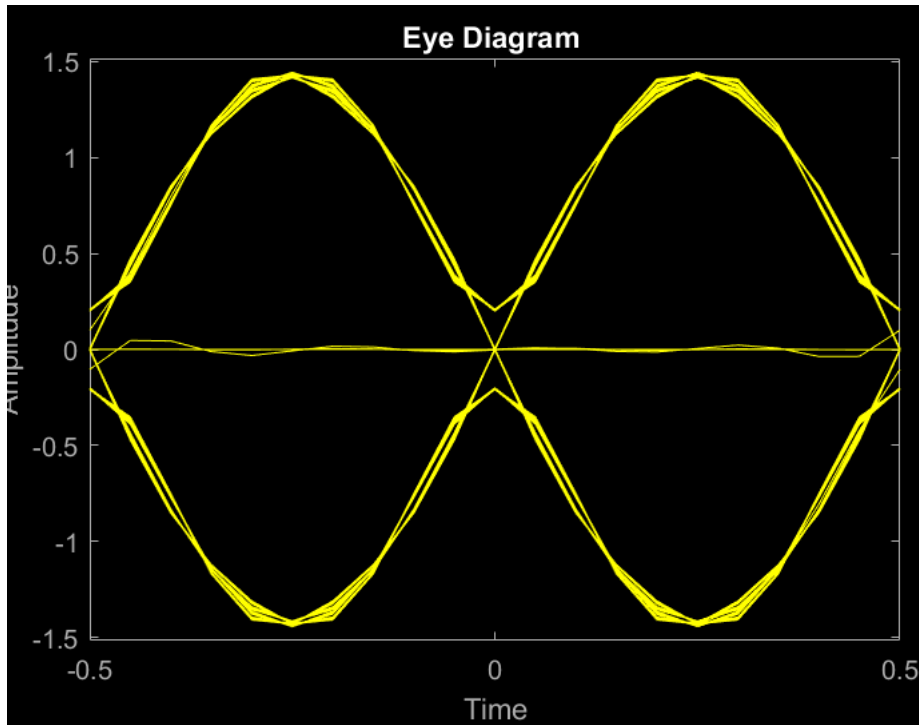
Ejercicio 4: Normalice la señal recibida para tener una potencia de 1 Watt en la señal recibida.

¿Cuál fue el procedimiento?

```
pow = sum(xPNRZ_filtrado_f1.^2)/numel(xPNRZ_filtrado_f1);
pow_Deseada = 1;
xPNRZ_filtrado_f1 = sqrt(pow_Deseada/pow)*xPNRZ_filtrado_f1;
xPNRZ_filtrado_f1_1W_pot = sum(xPNRZ_filtrado_f1.^2)/numel(xPNRZ_filtrado_f1)
```

```
xPNRZ_filtrado_f1_1W_pot = 1.0000
```

```
eyediagram(xPNRZ_filtrado_f1,2*mp);
```



```
pow = sum(xPNRZ_filtrado_f2.^2)/numel(xPNRZ_filtrado_f2);
pow_Deseada = 1;
xPNRZ_filtrado_f2 = sqrt(pow_Deseada/pow)*xPNRZ_filtrado_f2;
xPNRZ_filtrado_f2_1W_pot = sum(xPNRZ_filtrado_f2.^2)/numel(xPNRZ_filtrado_f2)
```

```
xPNRZ_filtrado_f2_1W_pot = 1.0000
```

```
eyediagram(xPNRZ_filtrado_f2,2*mp);
```

## ejercicio 5

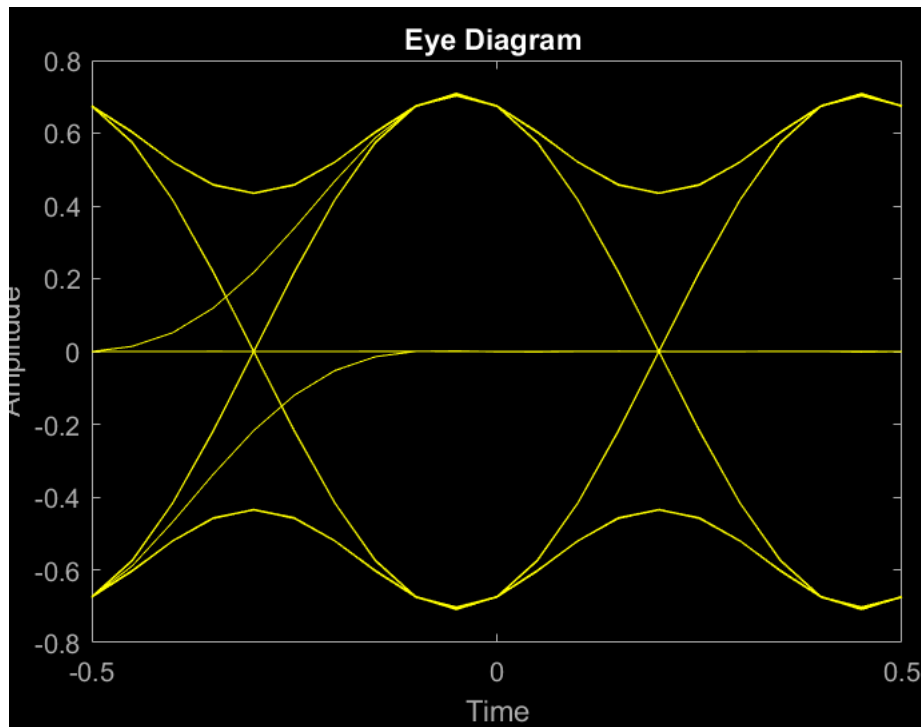
Ejercicio 5: Aplique el Match Filter (MF) para cada caso (pulso base rectangular y half-sine).

Observe el diagrama de ojo a la salida del MF y haga sus anotaciones al respecto.

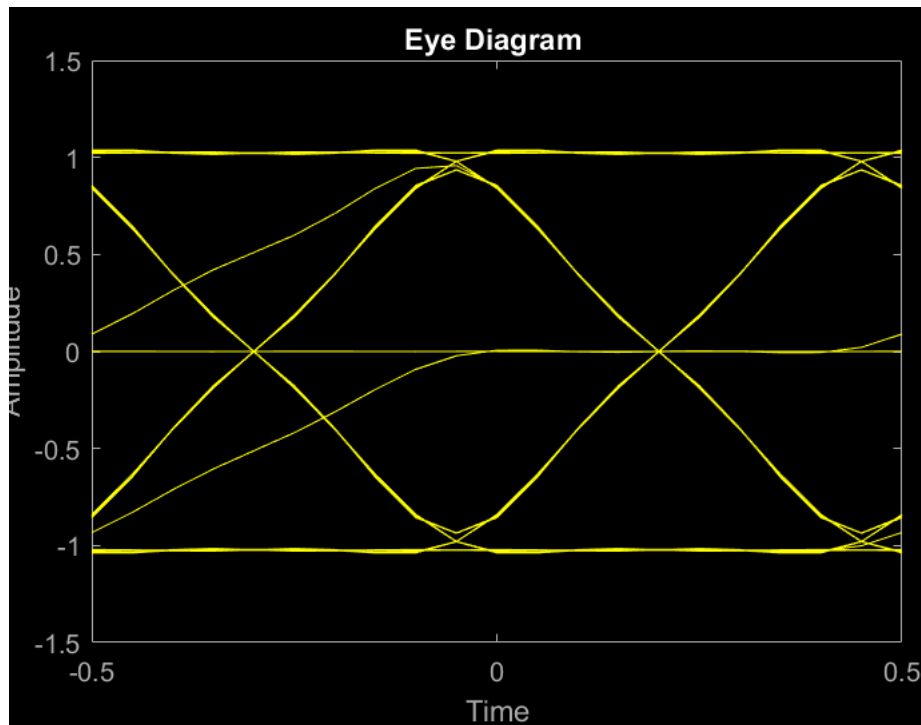
```
Match_filter1 = fliplr(hs);
Match_filter2 = fliplr(pnrz);

Const11 =conv(xPNRZ_filtrado_f1,Match_filter1)/mp;
Const22 =conv(xPNRZ_filtrado_f2,Match_filter2)/mp;

eyediagram(Const11,2*mp);
```



```
eyediagram(Const22,2*mp);
```



## ejercicio 6

Ejercicio 6: Muestree la señal recibida escogiendo el umbral (threshold) y el punto de muestreo para cada transmisión (pulso rectangular y pulso seno).

```

delay_signal = orden/2 + numel(hs)/2; %calculate the signal delay, channel delay + match filter
start_recovery_count = delay_signal + mp/2;

PNRZ_recovery_hs = Const11(start_recovery_count:mp:end);
PNRZ_recovery_rec = Const22(start_recovery_count:mp:end);

PNRZ_recovery_hs = PNRZ_recovery_hs(1:numel(bits2send));
PNRZ_recovery_rec = PNRZ_recovery_rec(1:numel(bits2send));
%Decition_treshold_PNRZ = 0;

plot(PNRZ_recovery_hs);
title('hs pulse sampled');

```

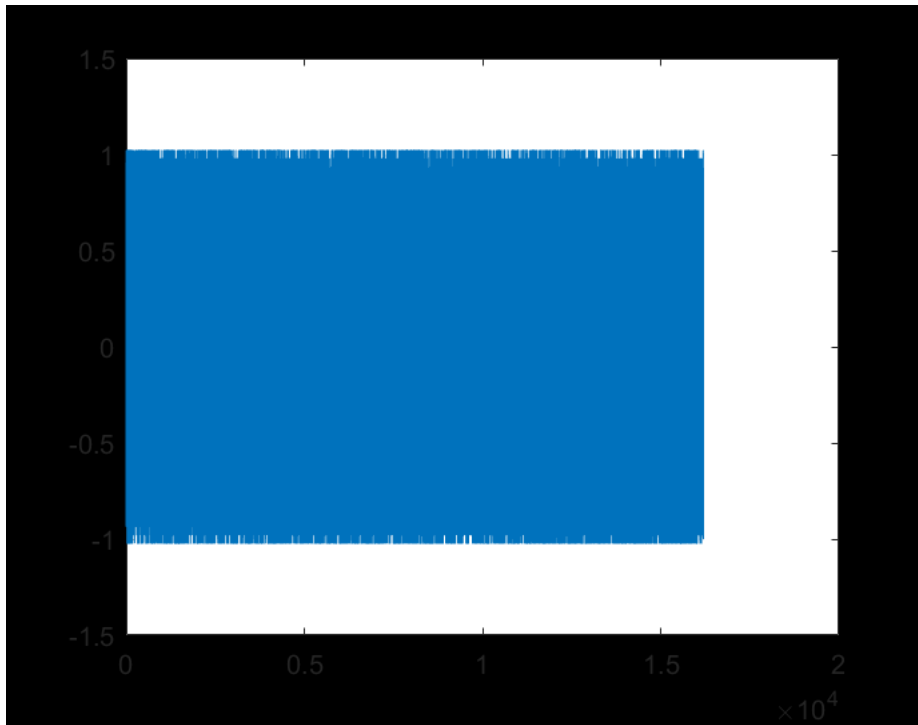


```

plot(PNRZ_recovery_rec);
title('rec pulse sampled');

```

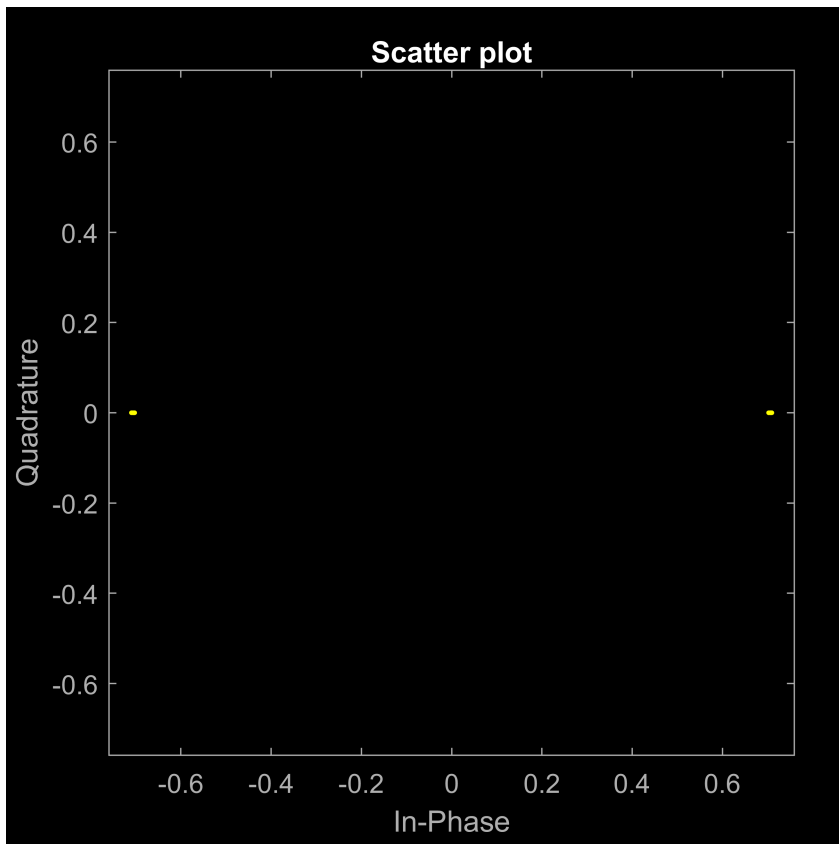




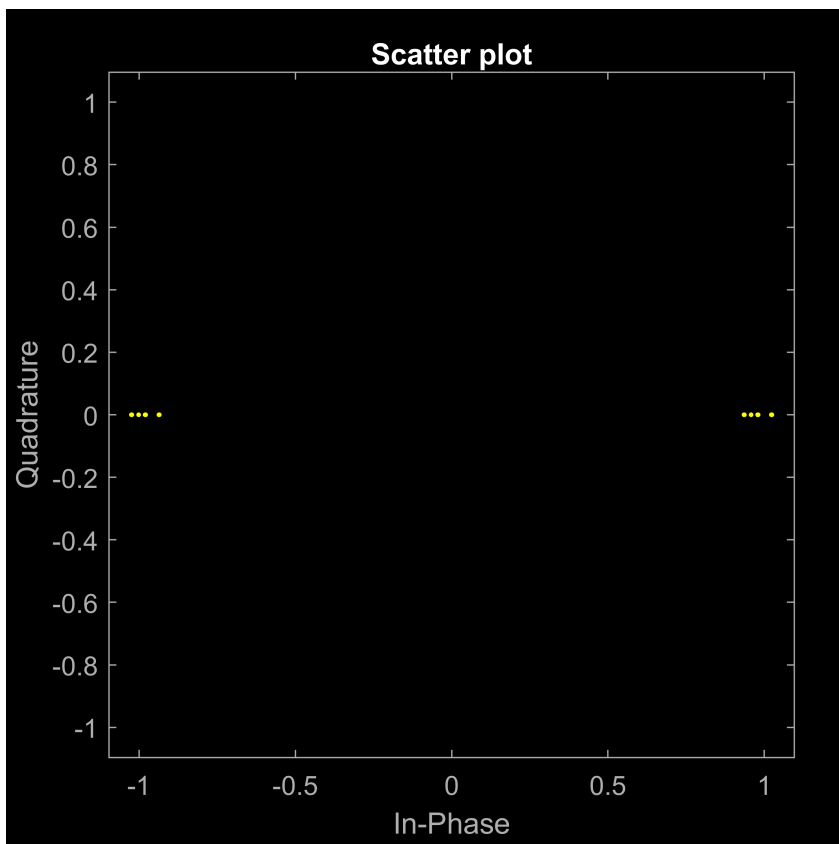
## ejercicio 7

Ejercicio 7: Grafique la constelación utilizando la función de scatterplot () de Matlab y relacione los resultados obtenidos con el diagrama de ojo obtenido a la salida del MF (Ejercicio 5).

```
scatterplot(PNRZ_recovery_hs);
```



```
scatterplot(PNRZ_recovery_rec);
```



## ejercicio 8

Ejercicio 8: Recupere la imagen recibida y calcule el BER correspondiente.

```
bits_RX_PNRZ_1 = zeros(1,numel(bits2send));  
bits_RX_PNRZ_2 = zeros(1,numel(bits2send));
```

```
xPNRZ_f_m1 = PNRZ_recovery_hs
```

```
xPNRZ_f_m1 = 1×16200  
    0.7071    -0.7093    0.7059    0.7026    0.7061   -0.7094    0.7087   -0.7092 ...
```

```
xPNRZ_f_m2 = PNRZ_recovery_rec
```

```
xPNRZ_f_m2 = 1×16200  
    0.9578   -0.9360    0.9804    1.0226    0.9806   -0.9364    0.9364   -0.9368 ...
```

```
bits_RX_PNRZ_1(xPNRZ_f_m1 > (max(PNRZ_recovery_hs)/2)) = 1;    %Umbral +  
bits_RX_PNRZ_2(xPNRZ_f_m2 > (max(PNRZ_recovery_rec)/2)) = 1;    %Umbral +
```

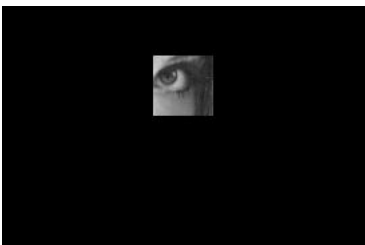
```
xPNRZ_BER_1 = (sum(xor(bits2send,bits_RX_PNRZ_1))/numel(bits2send)) * 100
```

```
xPNRZ_BER_1 = 0
```

```
xPNRZ_BER_2 = (sum(xor(bits2send,bits_RX_PNRZ_2))/numel(bits2send)) * 100
```

```
xPNRZ_BER_2 = 0
```

```
bitsM1 = reshape(bits_RX_PNRZ_1,8,45*45);  
bitsM1 = bitsM1';  
decval1 = bi2de(bitsM1,'left-msb');  
lenaRS1 = reshape(decval1, size(lenarec));  
imshow(uint8(lenaRS1))
```



```
bitsM1 = reshape(bits_RX_PNRZ_2,8,45*45);  
bitsM1 = bitsM1';  
decval1 = bi2de(bitsM1,'left-msb');  
lenaRS1 = reshape(decval1, size(lenarec));  
imshow(uint8(lenaRS1))
```

