

Tarea 9: Diseño de Pulsos Eficientes para la transmisión

Ejercicio 1.

Utilice el script `rcpulse` para diseñar pulsos coseno alzado (RC) que cumplan lo siguiente, y además determine los valores que se piden:

1. $F_s = 8000$, $B = 1000$, $R_b = 2000$, $D = 10$. Determine el valor de β .

$\beta = 0$;

2. $F_s = 8000$, $B = 1000$, $\beta = 0.2$, $D = 10$. Determine el valor de R_b .

$R_b = 0$;

3. $F_s = 4000$, $\beta = 0.8$, $R_b = 2000$, $D = 6$. Determine el valor de B .

$B = 1800$;

Ejercicio 2.

Para cada pulso de la pregunta anterior, determine su ancho de banda teórico y compárelo con su espectro calculado con la herramienta de `wvtool` de Matlab

Ejercicio 3.

1. Para observar, genere una señal basada en pulsos RC para transmitir la secuencia de bits [1 0 1 1 0 0 1 1 1 1].

Supongamos que $\beta = 0$, $F_s = 1000$, $T_p = 1/100$, $D = 10$. Utilice el mapeo polar NRZ con la siguiente regla:

bit 1 \rightarrow amplitud 1

bit 0 \rightarrow amplitud -1.

2. Grafique la señal transmitida, poniendo cuidado de que en el eje horizontal aparezca el tiempo. Identifique los instantes de tiempo donde aparecen las amplitudes que determinan cada bit transmitido.

Ejercicio 4. ETAPA DE TRANSMISIÓN.

a.) Genere el pulso base "Square Root-RC" suponiendo $\beta = 0$, $F_s = 96\text{KHz}$, $R_b = R_s = 9600$, $D = 10$, con energía igual a $T_p = 1/R_s$ Joules.

b.) Genere el tren de pulsos utilizando el código de línea POLAR NRZ y el pulso SRRC para transmitir la secuencia de bits asociada a la imagen de Lena recortada (128x128).

c.) Grafique la señal transmitida (solo la forma de onda asociada a los 16 primeros bits, además de los retardos correspondientes), poniendo cuidado de que en el eje horizontal aparezca el tiempo discreto. Identifique los instantes de tiempo donde aparecen las amplitudes que determinan cada símbolo/bit transmitido.

Ejercicio 5. ETAPA DE RECEPCIÓN

a.) Modele el canal de comunicación como un LPF, de orden 60 y frecuencia de corte igual a $0.6 \times \pi$.

b.) Pase la señal transmitida por el canal de comunicación modelado como un LPF. Aquí tendrá una señal filtrada por el canal, "signal_filtered"

c.) Posteriormente, pase esta señal por el filtro receptor denominado MATCH FILTER (recuerde que el pulso base es "Square Root-Raise Cosine", en este caso).

d.) Obtenga y analice la PSD de la señal a la salida del Match Filter utilizando un estimador

espectral de potencia

- e.) Obtenga los diagramas de ojo correspondientes para 3 UI.
- f.) Muestree la señal considerando los retardos y grafique la constelación digital
- g.) Con la señal muestreada y el umbral de decisión correspondiente, detecte los símbolos y decodifique los bits.
- h.) Calcule el Bit Error Rate
- i.) Reconstruya la imagen de la “lena recortada”
- j.) Realice sus observaciones y comente los resultados de los experimentos.

Codigo implementado:

```
%ejercicio 1.1
Fs = 8000;
B = 1000;
Rb = 2000;
D = 10; %Determine el valor de ?.
Rs = Rb;
Tp = 1/Rb;
Ts = 1/Fs;
%mp muestras son Tp segundos mp = Tp/Ts
mp = Tp/Ts;
%Rs = 2B/(1+r)
r = (2*B/Rs) - 1;
beta = r;
energy = Tp;
Type = 'rc';
pulso_base = rcpulse(beta, D, Tp, Ts, Type, energy);
%wvtool(pulso_base)

%ancho de banda = frecuencia en samples a los -2db * (Fs/2)
ancho_de_banda = 0.25 * (Fs/2)
%ejercicio 1.2
Fs = 8000;
B = 1000;
beta = 0.2;
D = 10; %determine el valor de Rb
Rs = 2000;
Rb = ((2*B)/Rs)-1;
Tp = 1/Rb;
Ts = 1/Fs;
energy = Tp;
Type = 'rc';
%pulso_base2 = rcpulse(beta, D, Tp, Ts, Type, energy);
%wvtool(pulso_base2)
ancho_de_banda = 0.4 * (Fs/2)

%ejercicio 1.3
Fs = 4000;
Rb = 2000;
beta = 0.8;
D = 6; %determine el valor de B
Tp = 1/Rb;
Ts = 1/Fs;
energy = Tp;
type = 'rc';
B=(Rb*(beta+1))/2;
pulso_base3 = rcpulse(beta, D, Tp, Ts, Type, energy);
%wvtool(pulso_base3)

%ejercicio 2
wvtool(pulso_base)
wvtool(pulso_base2)
wvtool(pulso_base3)

%ejercicio3
beta = 0;
Fs = 1000;
Tp = 1/100;
D = 10;
Rs = 2000;
Ts = 1/Fs;
mp = Tp/Ts; %mp
energy = Tp;
type = 'rc';

pbase = rcpulse(beta,D,Tp,Ts,type,energy);

%3.1
bit_sequence = [1 0 1 1 0 0 1 1 1 1];
PNRZ_map = zeros(1,numel(bit_sequence)*mp);

counter = 0;
for i= 0 : numel(bit_sequence)-1
    if bit_sequence(i+1) == 0
        value = -1;
    else
        value = 1;
    end
    PNRZ_map(counter) = value;
    counter = counter + mp;
end
```

```

else
    value = 1;
end
PNRZ_map(counter*i+1) = value;
counter = mp;
end
Polar_NRZ_signal = conv(pbase ,PNRZ_map);
figure();
plot(Polar_NRZ_signal);
title('Transmitted signal');

%ejercicio 4. Etapa de transmision
load lena512.mat
beta = 0;
Fs = 96000;
Ts = 1/Fs;
Rs = 9600;
D = 10;
Rb = Rs;
Tp = 1/Rb;
energy = sqrt(Tp);
type = 'rc';
mp = round(Tp/Ts); %mp

SRRC = rcpulse(beta,D,Tp,Ts,type,energy);
lenarec=lena512((284-127):284, (350-127):350);
b=de2bi(lenarec,8);
b=b';
bits=b(:);
bit_pixels = b(1:128*128*8);
bit_pixels_PNRZ = zeros(1,numel(bit_pixels)*mp);

counter = 0;
for i= 0 : numel(bit_pixels)-1
    if bit_pixels(i+1) == 0
        value = -1;
    else
        value = 1;
    end
    bit_pixels_PNRZ(counter*i+1) = value;
    counter = mp;
end

SRRC_PNRZ_signal = conv(bit_pixels_PNRZ, SRRC);
plot(SRRC_PNRZ_signal(51:51+mp*16));

start = round(numel(SRRC)/2);
sampled_signal = SRRC_PNRZ_signal(start:mp:end);

scatterplot(sampled_signal);
title('Bits plot');
figure();
pwelch(SRRC_PNRZ_signal,500,300,500,Fs,'power');
title('PSD bit stream');
eyediagram(SRRC_PNRZ_signal, 3*mp);

%Ejercicio 5. ETAPA DE RECEPCIÓN
f=[0 0.6 0.6 1];
m=[1 1 0 0];
ford=60;
filter_1 = fir2(ford,f,m);

signal_filtered = conv(filter_1,SRRC_PNRZ_signal);
match_f = fliplr(SRRC);
recover_signal = conv(match_f,signal_filtered);

start_recovery = ford/2 + start;
recovery_signal = recover_signal(start_recovery:mp:end);
figure();
pwelch(recover_signal,500,300,500,Fs,'power');
title('PSD bit stream');
%eyediagram
eyediagram(recover_signal, 3*mp);
scatterplot(recovery_signal);
treshhold = 0;

PNRZ_recovery_bits_rec = zeros(1,numel(bit_pixels));
PNRZ_recovery_bits_rec((recovery_signal > treshhold)) = 1;

bits_error_PNRZ = xor(bit_pixels, PNRZ_recovery_bits_rec(1:numel(bit_pixels)));
error_PNRZ = sum(bits_error_PNRZ);
Bit error rate PNRZ = (error_PNRZ/numel(bit_pixels)) * 100

```

Simulaciones:

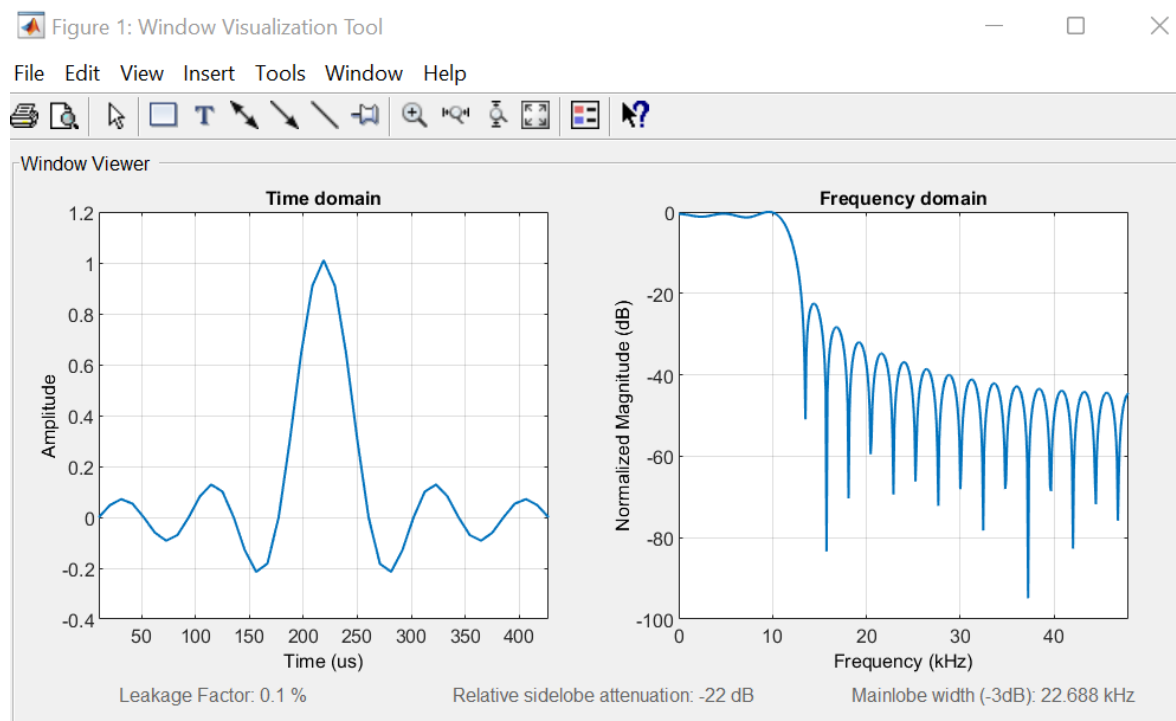


Figure 1 señal simulada

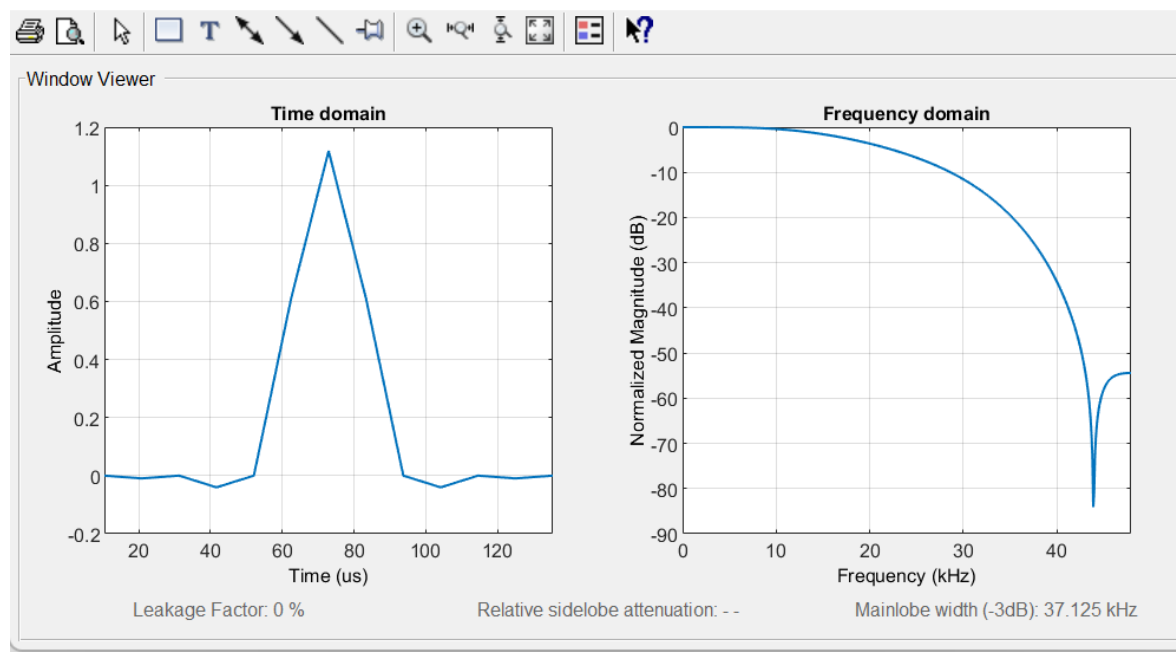


Figure 2 señal simulada [pulso 2]

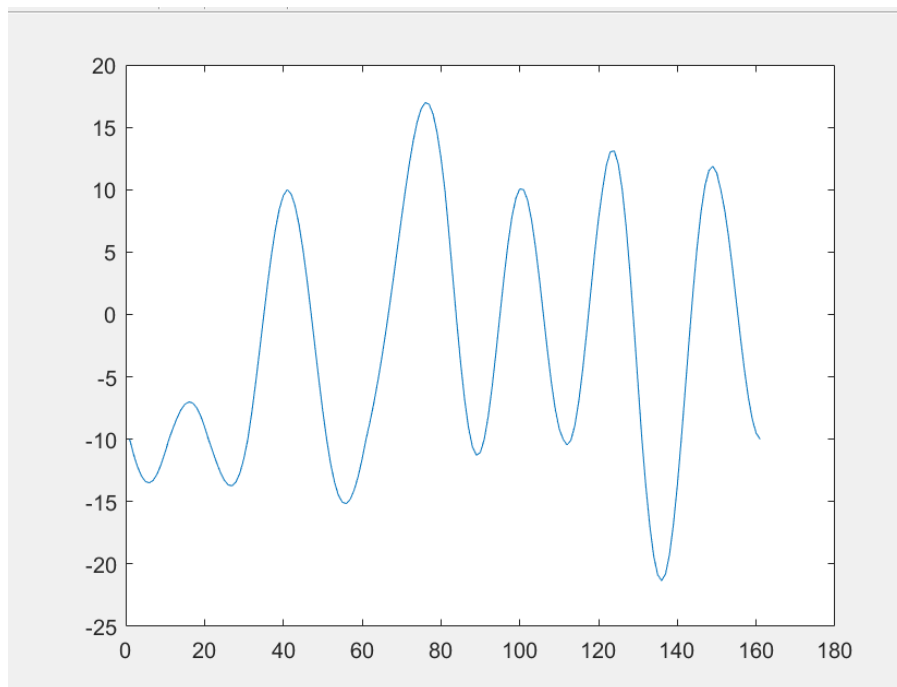


Figure 3 pulso a 16 muestras

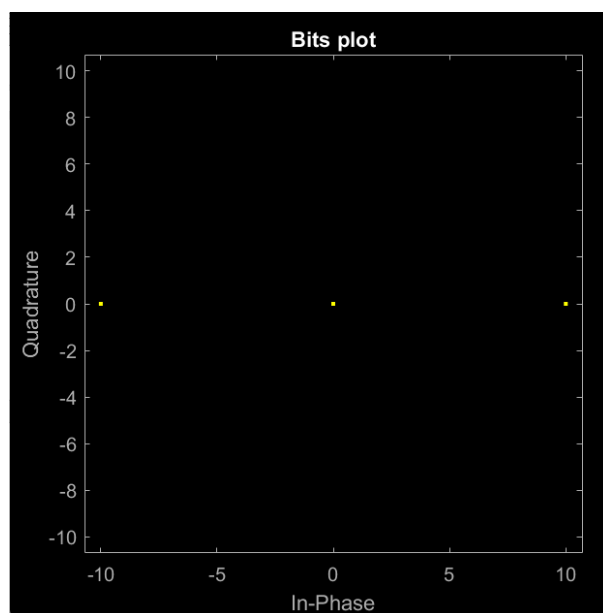


Figure 4 constelación

Neidys Marioly Vargas Montilva

Tarea #9

