Ejemplo CDMA

```
clc
clear all
close all

% Parámetros

n_data = 3;     % nro bits de datos
n_PN = 30;       % nro de bits secuencia pseudo aleatoria
n = 2;          % cantidad de usuarios simultaneos
```

Generación de datos

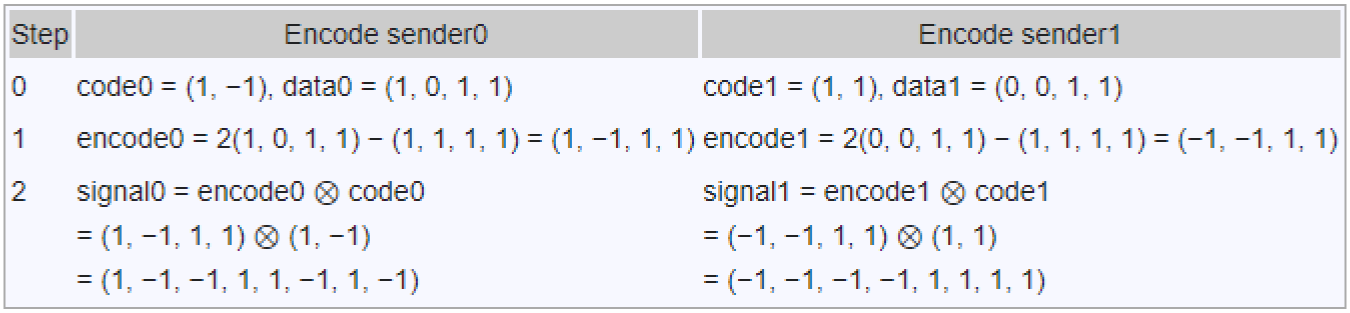
```
% generacion de datos y las secuencias pseudo aleatorias para la
% codificacion
data = randi([0 1],n_data,n);
PN = randi([0 1],n_PN,n);

for i = 1:n
    disp(strcat("Data usuario ",string(i)))
    transpose(data(:,i))
end
```

```
Data usuario 1
ans = 1×3
     1     0     0
Data usuario 2
ans = 1×3
     0     1     0
```

```
for i = 1:n
    disp(strcat("Secuencia pseudoaleatoria usuario ",string(i)))
    transpose(PN(:,i))
end
```

```
Secuencia pseudoaleatoria usuario 1
ans = 1×30
     0     1     1     1     1     0     1     1     0     1     0     0     1···
Secuencia pseudoaleatoria usuario 2
ans = 1×30
     0     0     0     1     0     1     1     1     0     0     0     1     0···
```

Codificación

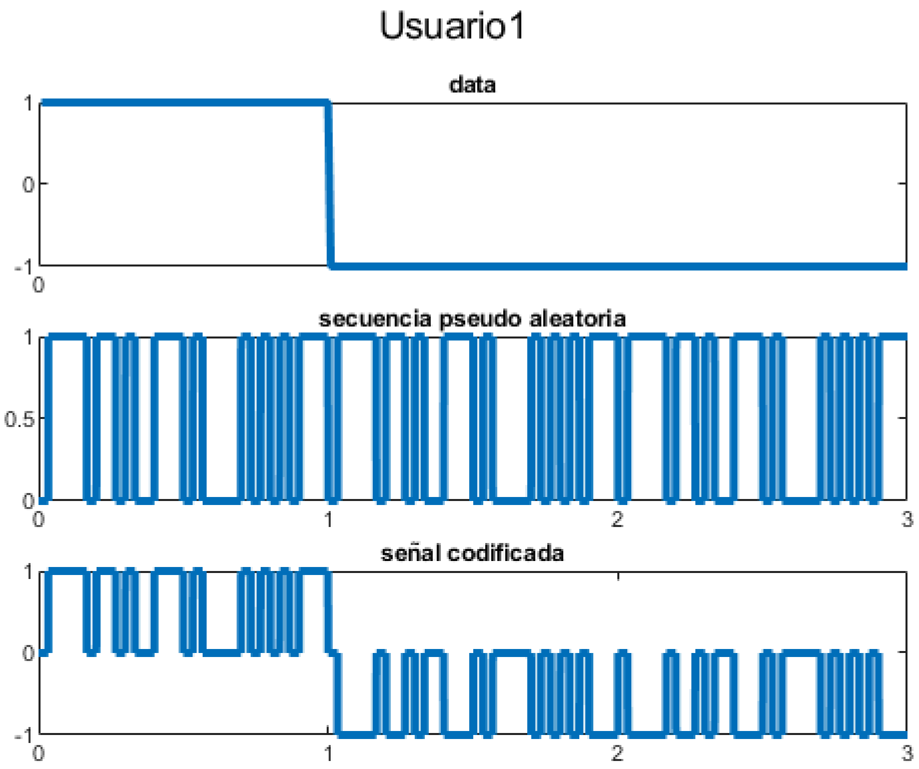


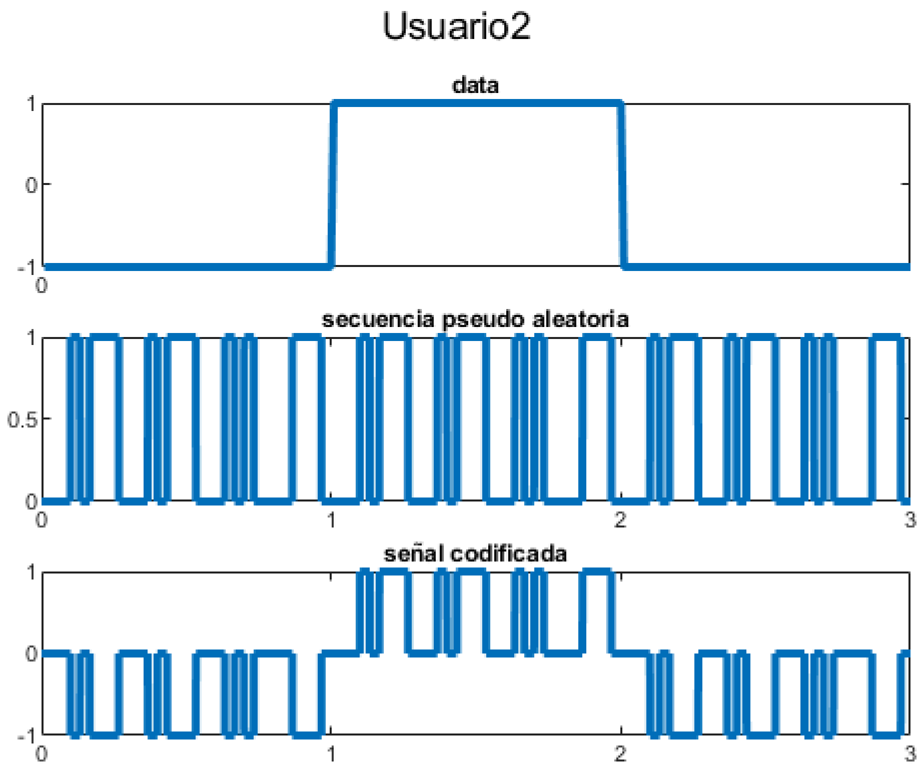
Step	Encode sender0	Encode sender1
0	code0 = (1, −1), data0 = (1, 0, 1, 1)	code1 = (1, 1), data1 = (0, 0, 1, 1)
1	encode0 = 2(1, 0, 1, 1) − (1, 1, 1, 1) = (1, −1, 1, 1)	encode1 = 2(0, 0, 1, 1) − (1, 1, 1, 1) = (−1, −1, 1, 1)
2	signal0 = encode0 ⊗ code0 = (1, −1, 1, 1) ⊗ (1, −1) = (1, −1, −1, 1, 1, −1, 1, −1)	signal1 = encode1 ⊗ code1 = (−1, −1, 1, 1) ⊗ (1, 1) = (−1, −1, −1, −1, 1, 1, 1, 1)

```
% codificacion
data = 2*data-1;

for i = 1:n
    signal(:,i) = kron(data(:,i),PN(:,i));
end
signal
```

```
signal = 90×2
     0     0
     1     0
     1     0
     1    -1
     1     0
     0    -1
     1    -1
     1    -1
     0     0
     1     0
     ⋮
```

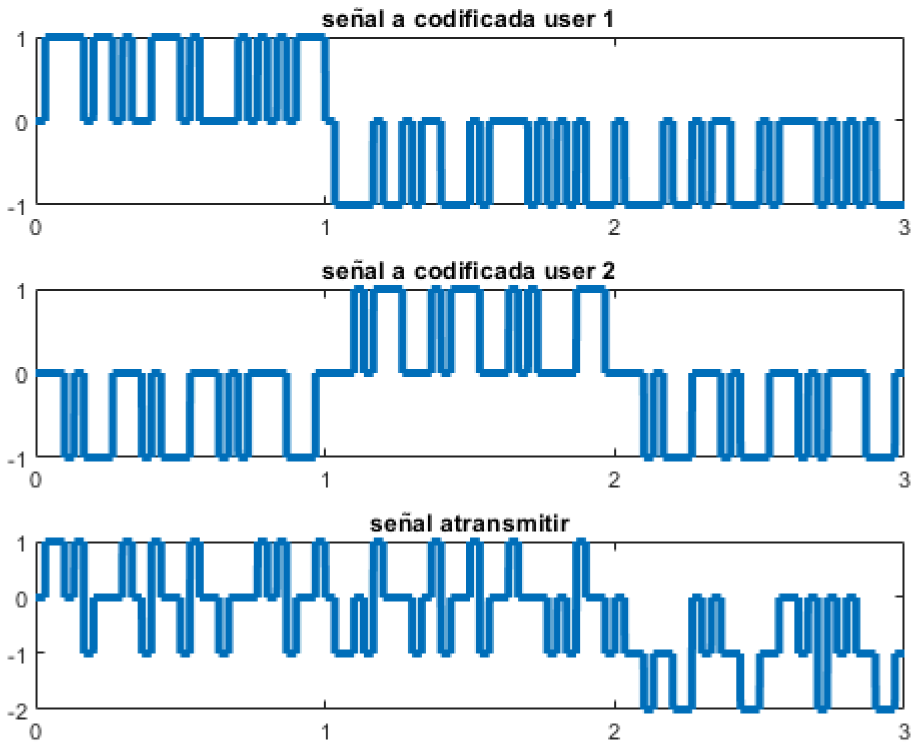
```
% concatena la secuencia PN n_data veces
PN_long = repmat(PN,[n_data 1]);

% repite los datos para graficar
PN_mat_plot = repelem(PN_long,100,1);
data_plot = repelem(data,100,1);
signal_plot = repelem(signal,100,1);

L=length(PN_mat_plot);


for user = 1:n
    figure
    subplot(3,1,1)
    plot(data_plot(:,user),'linewidth',3)
    title('data')
    % configs grafico
    set(gca,'XTick',0:n_PN*100:L);
    set(gca,'XTickLabel',0:n_data);

    subplot(3,1,2)
    plot(PN_mat_plot(:,user),'linewidth',3)
    title('secuencia pseudo aleatoria')
    % configs grafico
    set(gca,'XTick',0:n_PN*100:L);
    set(gca,'XTickLabel',0:n_data);

    subplot(3,1,3)
    plot(signal_plot(:,user),'linewidth',3)
    title('señal codificada')
    % configs grafico
    set(gca,'XTick',0:n_PN*100:L);
    set(gca,'XTickLabel',0:n_data);
    suptitle(strcat('Usuario ', string(user)))
end
```



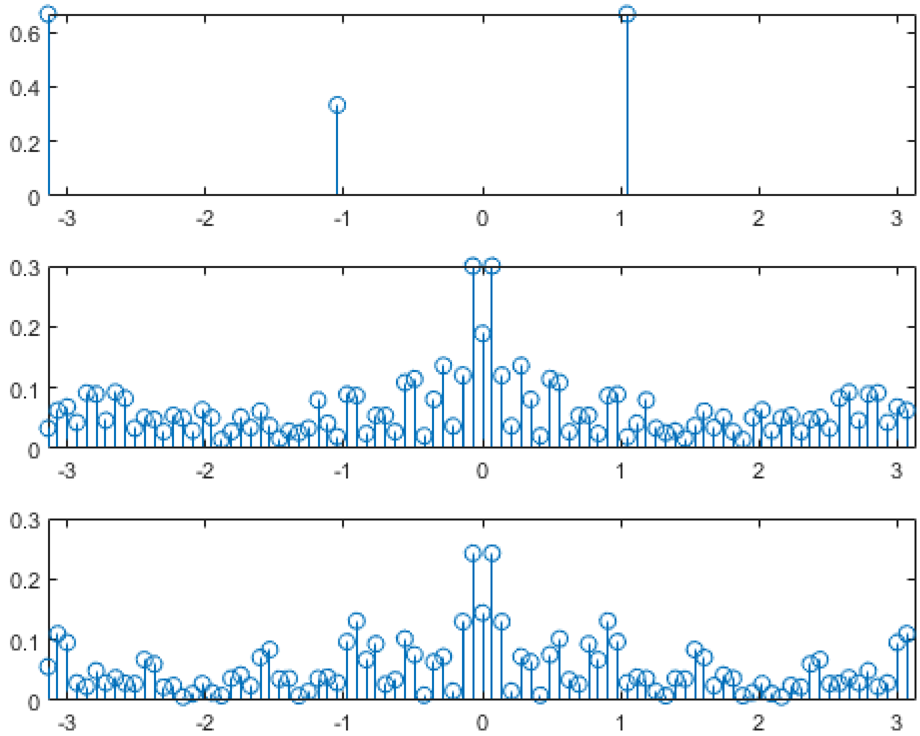


Combinación de señales

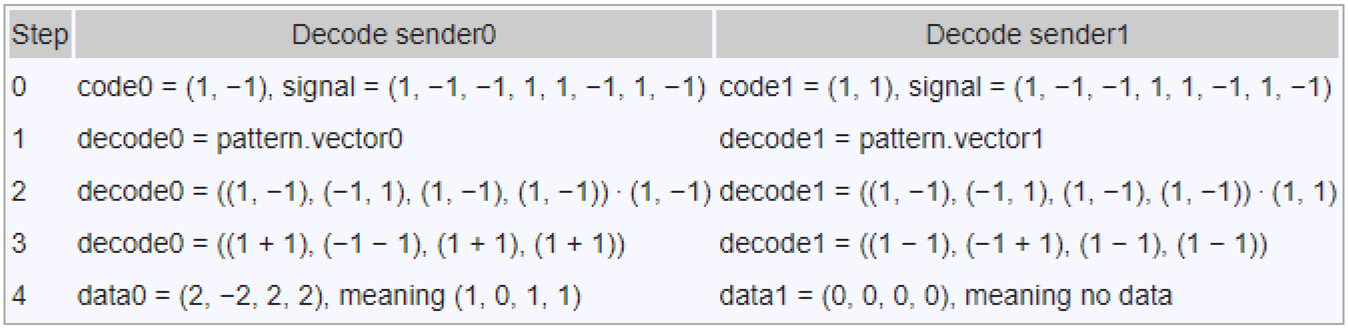
```
transmittesr_signal = sum(signal,2);
transmitter_signal_plot = sum(signal_plot,2);

figure
subplot(3,1,1)
plot(signal_plot(:,1),'linewidth',3)
title('señal a codificada user 1')
% configs grafico
set(gca,'XTick',0:n_PN*100:L);
set(gca,'XTickLabel',0:n_data);

subplot(3,1,2)
plot(signal_plot(:,2),'linewidth',3)
title('señal a codificada user 2')
% configs grafico
set(gca,'XTick',0:n_PN*100:L);
set(gca,'XTickLabel',0:n_data);

subplot(3,1,3)
plot(transmitter_signal_plot,'linewidth',3)
title('señal atransmitir')
% configs grafico
set(gca,'XTick',0:n_PN*100:L);
set(gca,'XTickLabel',0:n_data);
```



```
% espectros
f_data = -pi:2*pi/length(data):pi-2*pi/length(data);
f_signal = -pi:2*pi/length(signal):pi-2*pi/length(signal);
figure
subplot(3,1,1)
stem(f_data,abs(fftshift(fft(data(:,1))))/length(data))
xlim([-pi pi])
subplot(3,1,2)
stem(f_signal,abs(fftshift(fft(signal(:,1))))/length(signal))
xlim([-pi pi])
subplot(3,1,3)
stem(f_signal,abs(fftshift(fft(signal(:,2))))/length(signal))
xlim([-pi pi])
```

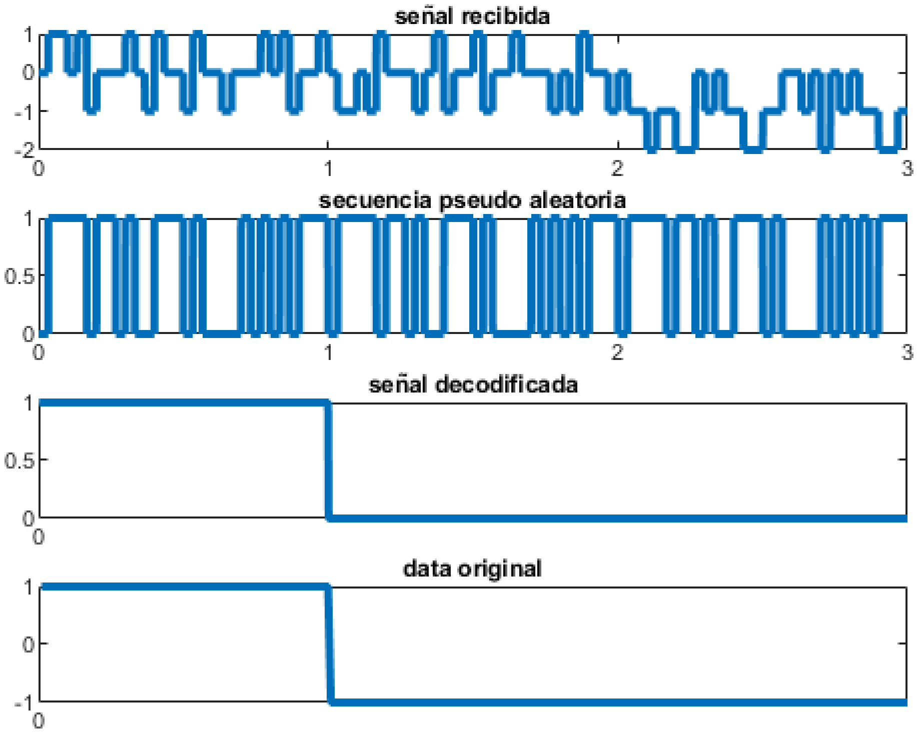


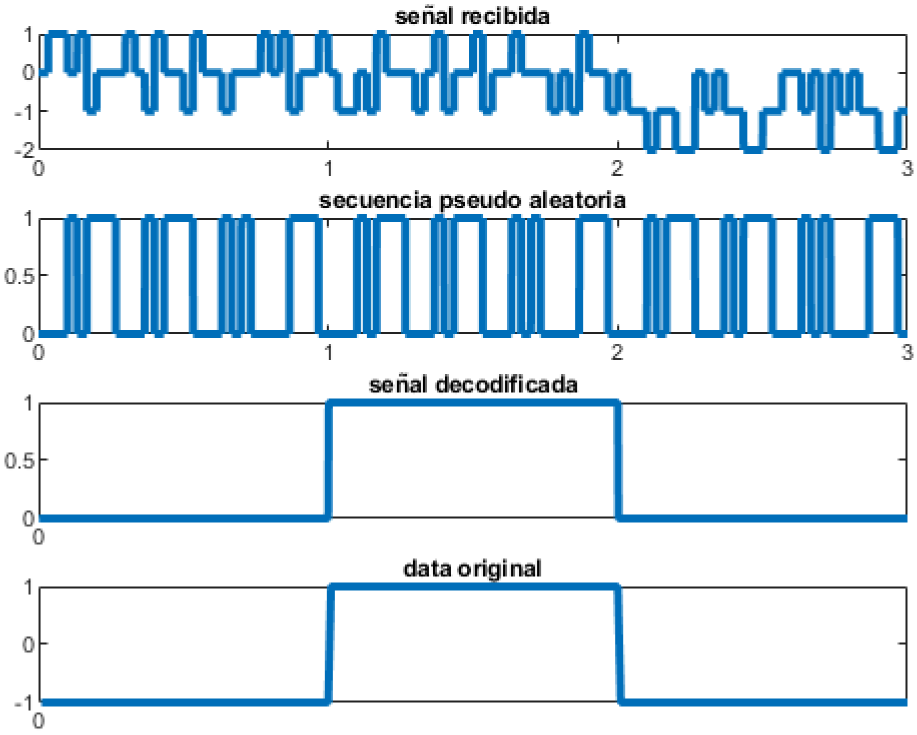
Decodificación



Step	Decode sender0	Decode sender1
0	code0 = (1, −1), signal = (1, −1, −1, 1, 1, −1, 1, −1)	code1 = (1, 1), signal = (1, −1, −1, 1, 1, −1, 1, −1)
1	decode0 = pattern.vector0	decode1 = pattern.vector1
2	decode0 = ((1, −1), (−1, 1), (1, −1), (1, −1)) · (1, −1)	decode1 = ((1, −1), (−1, 1), (1, −1), (1, −1)) · (1, 1)
3	decode0 = ((1 + 1), (−1 − 1), (1 + 1), (1 + 1))	decode1 = ((1 − 1), (−1 + 1), (1 − 1), (1 − 1))
4	data0 = (2, −2, 2, 2), meaning (1, 0, 1, 1)	data1 = (0, 0, 0, 0), meaning no data

```
for user = 1:n

    decoded_signal(:,user) = (reshape(transmittesr_signal,[n_PN n_data])'*PN(:,user));
    recovered_signal(decoded_signal(:,user) > 0) = 1;
    recovered_signal(decoded_signal(:,user) <= 0) = 0;


    %plots
    decoded_signal_plot = repelem(decoded_signal(:,user),500,1);
    recovered_signal_plot = repelem(recovered_signal',500,1);

    figure
    subplot(4,1,1)
    plot(transmitter_signal_plot,'linewidth',3)
    title('señal recibida')
    % configs grafico
    set(gca,'XTick',0:n_PN*100:L);
    set(gca,'XTickLabel',0:n_data);

    subplot(4,1,2)
    plot(PN_mat_plot(:,user),'linewidth',3)
    title('secuencia pseudo aleatoria')
    % configs grafico
    set(gca,'XTick',0:n_PN*100:L);
    set(gca,'XTickLabel',0:n_data);

    subplot(4,1,3)
    plot(recovered_signal_plot,'linewidth',3)
    title('señal decodificada')
    % configs grafico
    set(gca,'XTick',0:n_PN*100:L);
    set(gca,'XTickLabel',0:n_data);

    subplot(4,1,4)
    plot(data_plot(:,user),'linewidth',3)
    title('data original')
    % configs grafico
    set(gca,'XTick',0:n_PN*100:L);
    set(gca,'XTickLabel',0:n_data);
end
```




señal recibida
1
0
-1
-2
0
1
2
3
secuencia pseudo aleatoria
1
0.5
0
0
1
2
3
señal decodificada
1
0.5
0
0
data original
1
0
-1
0