

# Vídeo y Televisión

## PRÁCTICA II: MODULACIÓN QPSK

Diego Mirón | Óscar David Aponte | 22/01/2024

```

function SymbolStream = CodificaSecuencia(bitstream)
    %Verificar si la longitud del vector de bits es par
    if mod(length(bitstream),2) ~= 0
        error('El número de elementos debe ser par');
    end

    % Inicializar la secuencia de símbolos complejos
    SymbolStream = zeros(1,length(bitstream)/2);

    % Iterar sobre los pares de bits
    for i = 1:2:length(bitstream)
        % Obtener el par de bits actual
        bit1 = bitstream(i);
        bit2 = bitstream(i+1);

        % Convertir el par de bits a un símbolo QPSK
        if bit1 == 0 && bit2 == 0
            SymbolStream((i+1)/2) = 1;
        elseif bit1 == 1 && bit2 == 0
            SymbolStream((i+1)/2) = 1i;
        elseif bit1 == 0 && bit2 == 1
            SymbolStream((i+1)/2) = -1i;
        elseif bit1 == 1 && bit2 == 1
            SymbolStream((i+1)/2) = -1;
        else
            error('Secuencia binaria no válida');
        end
    end
end

function [sr, tr] = SumarRuido(s, t, desviacion)
    % Sumar ruido gaussiano a la señal compleja s(t)
    % s: Señal compleja original
    % t: Base de tiempos de la señal original
    % desviacion: Desviación típica (valor eficaz) del ruido a sumar

    % Generar ruido gaussiano independiente para la parte real e imaginaria
    ruido_real = desviacion * randn(size(s));
    ruido_imaginario = desviacion * randn(size(s));

    % Sumar ruido a la parte real e imaginaria de la señal
    sr = real(s) + ruido_real;
    si = imag(s) + ruido_imaginario;

    % Construir la señal compleja con ruido
    sr = sr + 1i * si;

    % Devolver la base de tiempos original
    tr = t;
end

function [s,t] = PasarAnalogico(SymbolStream,T0,rate)

Ts = T0*rate;
% Crear un vector de tiempo
t = 0:T0:(length(SymbolStream)-1)*rate;

% Generar la señal en deltas
delta_train = zeros(size(t/T0));

% Inicializar índice de símbolos
k = 1;

% Bucle para generar la señal en deltas
for n = 0:(length(SymbolStream)-1)*rate
    % Índice para evitar errores
    i = n+1;

    % Verificar si n*T0 es múltiplo de Ts
    if mod(n, rate) == 0
        % Asignar el símbolo actual al tren de deltas
        delta_train(i) = SymbolStream(k);
        k = k+1;
    end
end

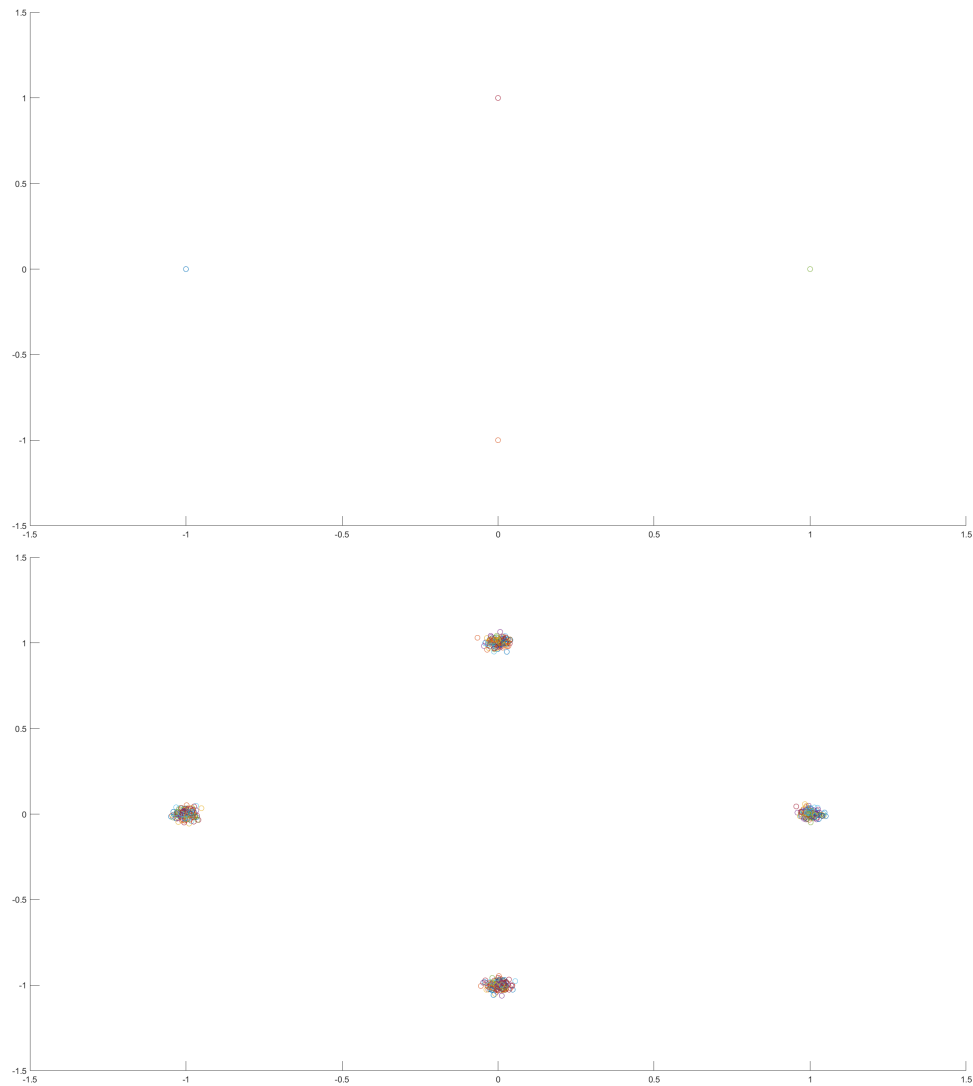
% Obtener el pulso conformador (coseno alzado)
[p,tp] = Pulsob(rate, Ts);

% Convolución de la señal en deltas con el pulso conformador
s = Convolucion(delta_train,t,p,tp);

% Normalizar la señal
s = s*rate;

end

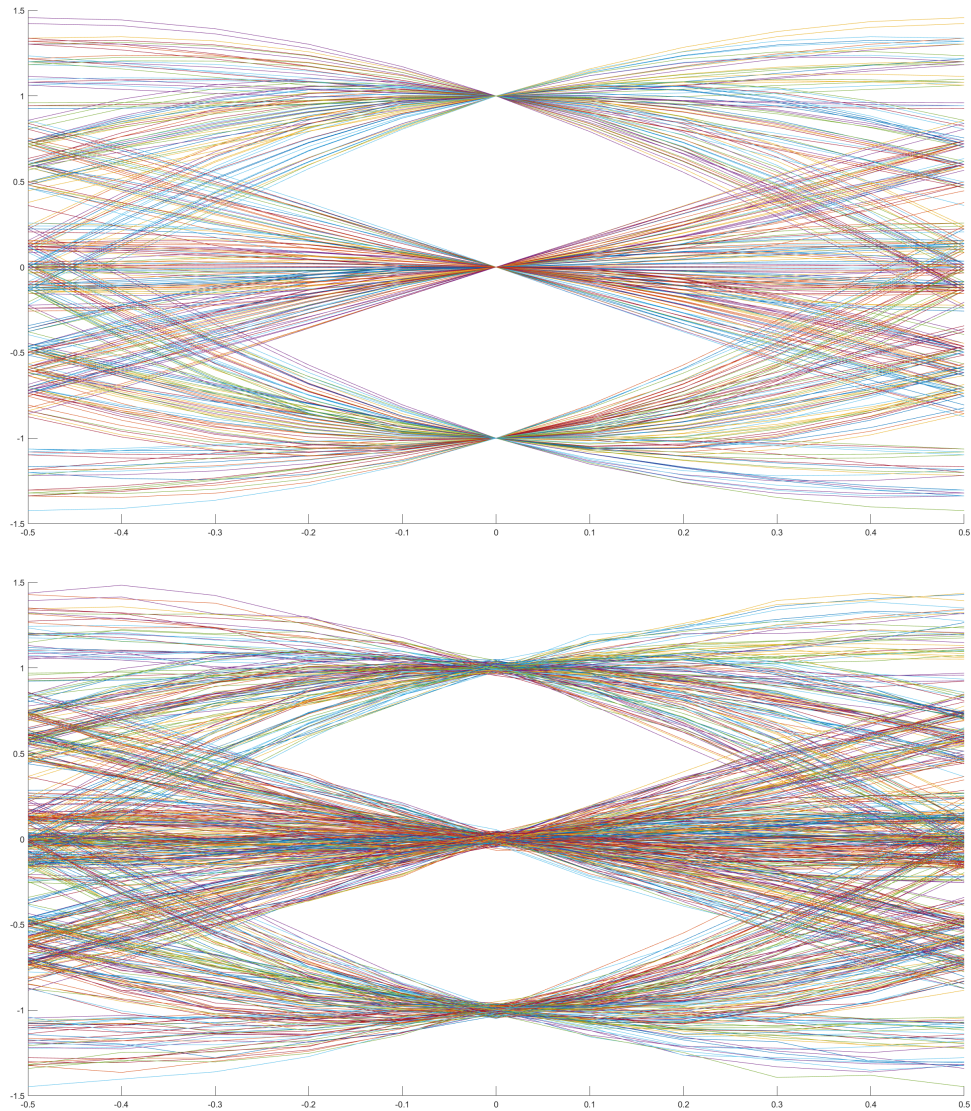
```



### ¿CUÁL ES EL EFECTO DEL RUIDO SOBRE LA CONSTELACIÓN? ¿EN QUÉ MOMENTO EMPEZARÁ A HABER ERRORES DE RECEPCIÓN?

El ruido suele mover de lugar los símbolos en la constelación porque agrega variabilidad a la amplitud y a la fase de la señal, afectando a la calidad de la misma y aumentando la probabilidad de errores de recepción.

Los errores de recepción pueden comenzar a ocurrir cuando la SNR cae por debajo de un cierto umbral crítico. Dicho de otra forma, cuando un símbolo se puede confundir con otro distinto debido al desplazamiento del ruido, ahí se producen errores.



### ¿QUÉ EFECTO TIENE EL RUIDO SOBRE EL DIAGRAMA DE OJO? ¿EN QUÉ MOMENTO EMPEZARÁ A HABER ERRORES DE RECEPCIÓN?

Puede reducir la apertura del ojo, dificultando la interpretación de los niveles de voltaje en los instantes de muestreo, lo que aumenta la probabilidad de errores.

Puede deformar también los bordes del ojo, haciendo que las transiciones entre los niveles de voltaje sean menos nítidas y se superpongan los símbolos adyacentes, lo que provoca errores de decisión del receptor.

Mayor jitter y desplazamiento temporal, lo que afecta a la precisión en la ubicación de los instantes de muestreo.

Los errores pueden producirse si los instantes de muestreo se vuelven demasiado difusos o si los distintos niveles de voltaje no son lo suficientemente claros.

SI OS PIDO SIMULAR LA MODULACIÓN QAM, ¿QUÉ CAMBIOS HARÍAIS? EN PARTICULAR, LA FÓRMULA QUE PROPORCIONA EL FINAL DEL INTERVALO DE INTERÉS ( $T_{MAX} = (L/2-1) \cdot T_s$ ), ¿SEGUIRÁ VALIENDO O HABRÁ QUE MODIFICARLA?, ¿SI HAY QUE CAMBIARLA CÓMO?

En una modulación QAM, cada símbolo representa 2 bits, igual que la QPSK. Otra cosa sería si nos dijese una 16-QAM, donde cada símbolo representa 4 bits.

Como estamos en QAM, la fórmula de  $T_{max}$  sigue siendo válida. La única diferencia que presenta la QAM frente a la QPSK es la diferencia de fase en sus símbolos, por lo que habría que codificar los símbolos de manera distinta a la QPSK, añadiendo una parte imaginaria o real según falte en cada símbolo.