

# Modulación de Amplitud de Cuadratura - 32QAM

Universidad Nacional de San Antonio Abad del Cusco

Escuela profesional de Ingeniería Electrónica

Telecomunicaciones II

Alexander Palomino Lopez

Ingeniero Electrónico

Cusco, Perú

alexander.palomino@unsaac.edu.pe

Davis Bremdow Salazar Roa - 200353

Estudiante de Ingeniería Electrónica

Cusco, Perú

200353@unsaac.edu.pe

**Abstract**—La elevada demanda de recursos digitales por la aparición de servicios digitales cuenta con cierta repercusión en los métodos para transmitir la información es en este sentido que los métodos a emplear deben tener en cuenta en el ancho de banda para responder adecuadamente a esta evolución mediante el empleo de técnicas de modulación como la 32QAM que permite mitigar el costo ocupado en frecuencia al transmitir la información en paquetes de 5bits.

**Index Terms**—Modulación digital, 32-QAM, M-QAM, Cuadratura, Reducción del ancho de banda

## I. INTRODUCCIÓN

La modulación en amplitud en cuadratura (QAM, Quadrature Amplitude Modulation) es una de las técnicas de modulación digital más utilizadas en los sistemas modernos de comunicaciones debido a su elevada eficiencia espectral. Al combinar variaciones simultáneas de amplitud y fase sobre dos portadoras ortogonales, QAM permite transmitir múltiples bits por símbolo, optimizando el uso del ancho de banda disponible. Esta característica la hace especialmente atractiva en aplicaciones de alta velocidad de datos, como sistemas de comunicaciones inalámbricas, enlaces de microondas, cable módem y transmisión sobre fibra óptica, donde la demanda de capacidad es cada vez mayor.

No obstante, el incremento en la eficiencia que ofrece QAM viene acompañado de mayores complejidades a nivel de diseño y procesamiento. A medida que aumenta el orden de la modulación, la constelación se vuelve más densa, reduciendo la distancia entre símbolos y haciendo al sistema más sensible al ruido, a la distorsión del canal y a errores de sincronización. Esto exige receptores más sofisticados, con algoritmos avanzados de ecualización, sincronización y corrección de errores, así como una mayor relación señal-ruido para garantizar un desempeño confiable. Desde una perspectiva de investigación, estas limitaciones impulsan el estudio de técnicas de optimización, adaptación de modulación y codificación, y modelado preciso del canal para maximizar el rendimiento de los sistemas basados en QAM.

## II. MARCO TEÓRICO

Para facilitar la comprensión del esquema para una modulación QAM en [1] se emplea a modo gráfico un diagrama de bloques para una modulación 8 QAM mostrada en la figura 1 que brinda la noción fundamental de su funcionamiento destacando el desfase de 90 grados para describir la cuadratura para la división de las señales ejecutada para su modulación.

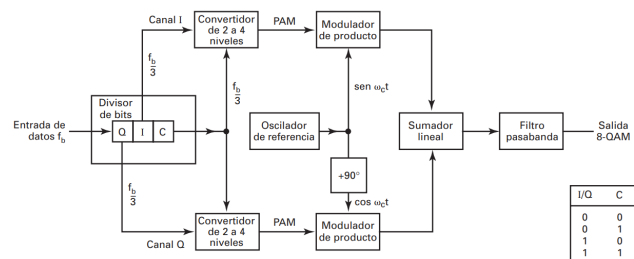


Fig. 1: Modulación 8 - QAM

En donde la señal binaria de entrada a velocidad  $f_b$  se procesa para transmitir información tanto en amplitud como en fase. Los datos ingresan al divisor de bits, que agrupa los bits en símbolos de 3 bits y los distribuye en los canales I (In-phase), Q (Quadrature) y un bit de control C, reduciendo la velocidad de cada rama a  $f_b/3$ . Los canales I y Q alimentan convertidores de 2 a 4 niveles, que generan señales PAM multinivel, permitiendo representar múltiples estados de amplitud asociados a la constelación 8-QAM.

Las señales PAM [2] de los canales I y Q se aplican a moduladores de producto, donde se multiplican por portadoras ortogonales generadas por un oscilador de referencia:  $\sin(\omega_c t)$  y  $\cos(\omega_c t)$ , esta última obtenida mediante un desfase de  $+90^\circ$ . Las salidas moduladas se combinan en un sumador lineal, produciendo una señal que contiene simultáneamente ambas componentes ortogonales. Finalmente, un filtro pasabanda limita el espectro y elimina componentes no deseadas, entregando a la salida la señal 8-QAM, lista para su transmisión por el canal.

### A. Modelo matemático de la señal QAM

La señal QAM pasabanda puede expresarse como la combinación lineal de dos señales moduladas en amplitud sobre portadoras ortogonales:

$$s(t) = I(t) \cos(2\pi f_c t) - Q(t) \sin(2\pi f_c t) \quad (1)$$

donde  $I(t)$  representa la componente en fase y  $Q(t)$  la componente en cuadratura. Ambas señales corresponden a secuencias de símbolos discretos que toman valores de un conjunto finito de niveles de amplitud. La ortogonalidad entre las funciones seno y coseno garantiza que ambas componentes puedan ser recuperadas de manera independiente en el receptor, aun cuando se transmiten simultáneamente sobre el mismo canal.

### B. Representación en banda base compleja

Para simplificar el análisis matemático, Tomasi introduce la representación en banda base compleja, donde la señal QAM se expresa como:

$$s_{bb}(t) = I(t) + jQ(t) \quad (2)$$

Esta representación permite modelar cada símbolo como un número complejo, facilitando el estudio del sistema en presencia de ruido y distorsiones del canal. Cada punto del plano complejo corresponde a un símbolo específico, y el conjunto de todos los símbolos posibles define la constelación QAM.

### C. Constelación y número de símbolos

En una modulación  $M$ -QAM, el número total de símbolos posibles está dado por:

$$M = 2^k \quad (3)$$

donde  $k$  es el número de bits por símbolo. Cada símbolo puede representarse como:

$$s_m = a_m + jb_m \quad (4)$$

siendo  $a_m$  y  $b_m$  los niveles de amplitud discretos en los ejes I y Q, respectivamente. Tomasi destaca que, a medida que aumenta  $M$ , la constelación se vuelve más densa, incrementando la eficiencia espectral pero reduciendo la distancia mínima entre símbolos.

### D. Energía promedio del símbolo

La energía promedio por símbolo es un parámetro clave en el desempeño del sistema y se define como:

$$E_s = \mathbb{E}[|s_m|^2] = \mathbb{E}[a_m^2 + b_m^2] \quad (5)$$

La energía promedio por bit se obtiene dividiendo la energía por símbolo entre el número de bits transmitidos:

$$E_b = \frac{E_s}{\log_2(M)} \quad (6)$$

Este parámetro es fundamental para comparar el desempeño de distintas modulaciones en términos de eficiencia energética, tal como se analiza en profundidad en el texto de Tomasi.

### E. Modelo de canal AWGN

Considerando un canal ideal con ruido blanco gaussiano aditivo (AWGN), la señal recibida puede expresarse como:

$$r(t) = s(t) + n(t) \quad (7)$$

donde  $n(t)$  es un proceso gaussiano de media cero y densidad espectral de potencia  $N_0/2$ . Este modelo es ampliamente utilizado en investigación por su simplicidad y por permitir obtener expresiones analíticas cerradas para la probabilidad de error.

### F. Distancia euclidiana mínima

Tomasi resalta que el desempeño de una modulación QAM está directamente relacionado con la distancia euclidiana mínima entre símbolos:

$$d_{\min} = \min_{i \neq j} |s_i - s_j| \quad (8)$$

Una mayor distancia mínima implica una mayor robustez frente al ruido, mientras que una constelación más densa reduce  $d_{\min}$  y aumenta la probabilidad de error.

### G. Probabilidad de error de bit

Para un sistema  $M$ -QAM sobre un canal AWGN, la probabilidad de error de bit puede aproximarse por:

$$P_b \approx \frac{4}{\log_2(M)} \left(1 - \frac{1}{\sqrt{M}}\right) Q \left( \sqrt{\frac{3 \log_2(M)}{M-1} \frac{E_b}{N_0}} \right) \quad (9)$$

donde la función  $Q(x)$  está definida como:

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-t^2/2} dt \quad (10)$$

Esta expresión permite evaluar el compromiso entre eficiencia espectral y robustez frente al ruido, uno de los aspectos centrales en el análisis y diseño de sistemas QAM descritos por [1]

## III. SIMULACIÓN

Para la simulación se hizo uso del software de MATLAB que entre todos sus beneficios para el manejo de vectores y matrices, además cuenta con una toolbox para la simulación de comunicaciones (para este caso digitales) y el empleo de funciones implementadas que permiten emular de forma fácil de comprender y de rápida ejecución las diferentes modulaciones digitales como la QAM para este caso asociada a 5 bits **32 - QAM**.

Por otro lado este software también permite agregar ruido AWGN de forma controlada (niveles configurables) para realizar un análisis más exhaustivo sobre el comportamiento de una 32 QAM frente al ruido, por tanto dentro de esta simulación se contará con las siguientes etapas.

- Modulación
- Añadido de ruido AWGN
- Transmisión
- Demodulación
- Gráficas y curvas relacionadas

## A. Modulación

La modulación 32QAM transforma secuencias binarias en símbolos complejos distribuidos en el plano I-Q como se describe en [3] asignando para esto grupos de 5 bits se asigna a una de las 32 posiciones de constelación minimizando errores al asignar bits adyacentes a símbolos vecinos y utilizando para la implementación la función `qammod()` de MATLAB, aplicando al mismo tiempo portadoras ortogonales en coseno (I) y seno (Q) para modular la señal en banda base a frecuencia intermedia (conocidas técnicamente como señal en banda media).

Por otro lado el código 1 genera la constelación teórica como referencia y visualiza la asignación símbolo-bit, mostrando la distribución uniforme de los 32 puntos en una cuadrícula 6x6 en la cuadrícula dibujada, destacando que tal representación mostradas en las figuras 2 y 3



Fig. 2: Diagrama de constelación modulación 32QAM

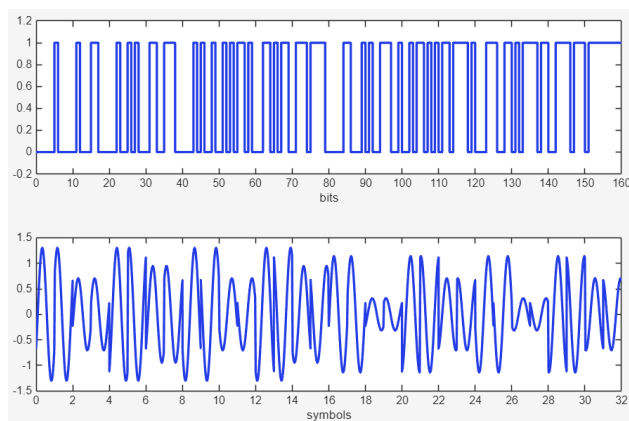


Fig. 3: Modulación 32QAM

Permiten constatar los conceptos vistos frente al esquema en el cual la amplitud y fase de cada punto codifican la información digital.

## B. Añadido de ruido AWGN

El ruido AWGN (Additive White Gaussian Noise) se añade modelando condiciones reales de canal mediante muestras gaussianas complejas independientes. La potencia de ruido se calcula a partir de la SNR especificada en dB, convirtiéndola a escala lineal mediante la relación  $A = 10^{SNR_dB/10}$  para determinar la varianza y nivel de ruido definida en el bloque de código 2 implementado, permite analizar el degradamiento respecto al SNR.

Por otro lado es importante destacar que el ruido afecta tanto a los símbolos en banda base como a la señal modulada, pudiendo analizar este comportamiento mediante la relación entre el BER respecto al SNR como se muestra en la figura 4.

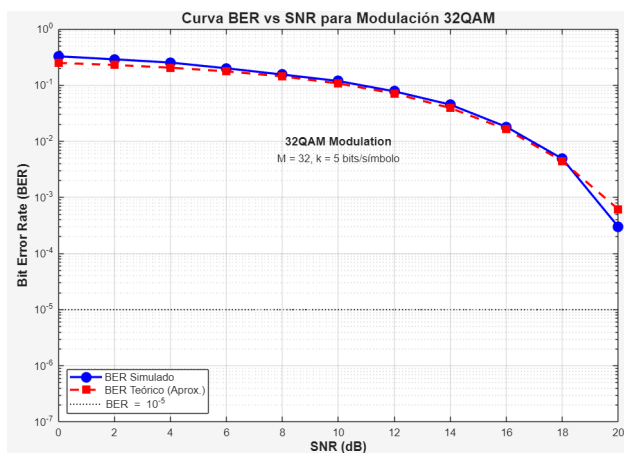


Fig. 4: Probabilidad de Error de bit vs SNR - Modulación 32QAM

## C. Transmisión

La transmisión simula el paso de la señal por el canal mediante la combinación de la señal modulada con el ruido generado, equivaliendo esto en simulación al uso de variables empleadas para almacenar las señales moduladas en las etapas posteriores, sin embargo también es posible modelar un canal para emular las pérdidas o interferencia sujetas a la experimentación física real.

## D. Demodulación

Finalmente en el proceso inverso o demodulación se emplea la función `qamdemod()` de MATLAB para decidir, a partir de los símbolos ruidosos recibidos, qué símbolo de constelación fue transmitido utilizando para este proceso la detección por mínima distancia euclidiana a los puntos de constelación ideales, seguido de la conversión de símbolos a bits.

Por ello dado el proceso inverso es que también en este bloque es que se puede realizar un análisis preciso entre los bits modulados y los demodulados verificando el error y cual es su relación con el SNR de la señal, la implementación muestra el diagrama de constelación para diferenciar gráficamente como afecta el ruido a las figuras 5 y 6

El sistema demuestra cómo, incluso con ruido significativo, la mayoría de decisiones son correctas cuando

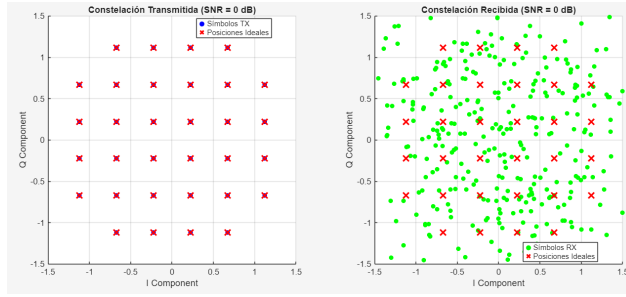


Fig. 5: Señal 32QAM demodulada - SNR 0 dB

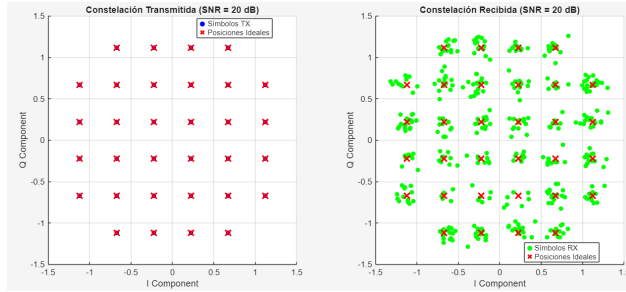


Fig. 6: Señal 32QAM demodulada - SNR 20 dB

los símbolos recibidos caen dentro de las regiones de decisión apropiadas. Se incluye análisis de errores por posición de bit dentro del símbolo, mostrando que los bits mapeados a decisiones de signo (MSBs) son más robustos que aquellos mapeados a decisiones de magnitud (LSBs) en QAM.

#### IV. CONCLUSIÓN

El método de simulación implementado en MATLAB, basado en la modulación y demodulación mediante funciones estándar como `qammod()` y `qamdemod()`, permitió modelar de manera precisa el comportamiento de un sistema 32-QAM sobre un canal AWGN con una correcta parametrización de la potencia unitaria del símbolo y la inclusión de mapeo Gray garantizaron que el análisis del BER reflejara fielmente las condiciones teóricas, validando el uso de este enfoque como una herramienta confiable para el estudio del desempeño de modulaciones digitales de alto orden.

El análisis del BER en función del SNR evidenció claramente la relación inversa entre la probabilidad de error y la relación señal-ruido en sistemas 32-QAM. A bajos valores de SNR, la elevada densidad de la constelación incrementa significativamente la probabilidad de error debido a la reducción de la distancia euclidiana mínima entre símbolos. Sin embargo, a medida que el SNR aumenta, el BER disminuye de forma exponencial, mostrando una fuerte concordancia entre los resultados simulados y la expresión teórica, lo que confirma la validez del modelo AWGN utilizado.

Los resultados obtenidos demuestran que, si bien la modulación 32-QAM ofrece una alta eficiencia espectral al transmitir 5 bits por símbolo, esta ventaja se logra a costa de una mayor sensibilidad al ruido, reflejada directamente en el comportamiento del BER. El método

de análisis empleado permite cuantificar este compromiso entre eficiencia espectral y robustez, proporcionando criterios claros para la selección de esquemas de modulación en función de los requerimientos de calidad de servicio y las condiciones del canal.

#### REFERENCES

- [1] W. Tomasi, *Sistemas de Comunicaciones Electrónicas*, 4th ed. Prentice Hall, 2001.
- [2] L. W. Couch, *Sistemas de comunicación digitales y analógicos*, 7th ed. México: Pearson Educación, 2012.
- [3] W. Stallings, *Comunicaciones y Redes de Computadores*, 7th ed. Madrid, España: Pearson Prentice Hall, 2004, traducción de la 7ª edición en inglés.

## V. ANEXOS

### Modulación 32-QAM

```
%% Parámetros de la simulación
order = 32;
symorder = 'gray';

symbols = 0 : order - 1;
bits = reshape(de2bi(symbols).', [],
    ↪ 1);

%% Modulación QAM
qam_symbols = qammod(bits, order,
    ↪ symorder, "InputType", "bit",
    ↪ "UnitAveragePower", true);
n_symbols = length(qam_symbols);
```

Listing 1: Modulación 32-QAM

### Análisis de ruido AWGN

```
signal_power =
    ↪ mean(abs(tx_symbols).^2);

SNR_linear = 10^(SNR_dB/10);

noise_power = signal_power /
    ↪ SNR_linear;

noise_symbols = sqrt(noise_power/2) *
    ↪ ...
    (randn(size(tx_symbols)) +
    ↪ 1i*randn(size(tx_symbols)));

rx_symbols = tx_symbols +
    ↪ noise_symbols;

noise_signal = sqrt(noise_power/2) *
    ↪ randn(size(modulated_signal));
noisy_signal = modulated_signal +
    ↪ noise_signal;
```

Listing 2: Añadido de ruido AWGN

### Demodulación 32-QAM

```
rx_bits_matrix = qamdemod(rx_symbols,
    ↪ M, symorder, ...
    ↪ "OutputType", "bit", "UnitAveragePower", true);

rx_bits = reshape(rx_bits_matrix.', [],
    ↪ 1);

bit_errors = data_bits ~= rx_bits;
BER = mean(bit_errors);

symbol_errors = any(reshape(bit_errors,
    ↪ k, [], 1);
SER = mean(symbol_errors);

EVM_rms = sqrt(mean(abs(rx_symbols -
    ↪ tx_symbols).^2)) / ...
    ↪ sqrt(mean(abs(tx_symbols).^2)) * 100;
```

Listing 3: Demodulación QAM

```

SNR_dB_range = 0:2:20;
num_snr_points = length(SNR_dB_range);
BER_simulated = zeros(1,
    ↪ num_snr_points);
BER_theoretical = zeros(1,
    ↪ num_snr_points);
bits_per_snr_test = 50000;

Q = @(x) 0.5 * erfc(x/sqrt(2));

for snr_idx = 1:num_snr_points
    SNR_linear =
        ↪ 10^(SNR_dB_range(snr_idx)/10);

    test_bits = randi([0 1],
        ↪ bits_per_snr_test, 1);
    test_symbols =
        ↪ qammod(reshape(test_bits, k, []).',
        ↪ M, symorder, ...
        "InputType", "bit", "UnitAveragePower", true);

    noise_power =
        ↪ mean(abs(test_symbols).^2) /
        ↪ SNR_linear;
    noise = sqrt(noise_power/2) * ...
        (randn(size(test_symbols)) +
        ↪ 1i*randn(size(test_symbols)));
    rx_symbols = test_symbols + noise;

    rx_bits = qamdemod(rx_symbols, M,
        ↪ symorder, ...
        "OutputType", "bit", "UnitAveragePower", true);
    rx_bits = reshape(rx_bits.', [], 1);

    BER_simulated(snr_idx) = mean(test_bits
        ↪ ~ rx_bits);

    EbN0_linear = SNR_linear / k;
    BER_theoretical(snr_idx) = (4/k) * (1 -
        ↪ 1/sqrt(M)) * ...
    Q(sqrt(3*k*EbN0_linear/(M-1)));
end

figure;
semilogy(SNR_dB_range, BER_simulated,
    ↪ 'o-'); hold on;
semilogy(SNR_dB_range, BER_theoretical,
    ↪ '--');
grid on;
xlabel('SNR (dB)');
ylabel('BER');
legend('BER Simulado', 'BER
    ↪ Teórico', 'Location', 'southwest');
ylim([1e-7 1]);
xlim([min(SNR_dB_range)
    ↪ max(SNR_dB_range)]);

```

Listing 4: BER - SNR