La Modulación M-PSK

Santiago Andrés Ibañez Ramírez
2211258
santiago2211258@correo.uis.edu.co

Nicolás David Martínez Cristancho
2212269
nicolas2212269@correo.uis.edu.co

Jonathan Mauricio Rojas Rodriguez
2212280
jonathan2212280@correo.uis.edu.co

Escuela de Ingeniería Eléctrica, Electrónica y de Telecomunicaciones
Universidad Industrial De Santander
Bucaramanga, Colombia

Repositorio Github

***Resumen*—This laboratory practice focuses on implementing and analyzing M-PSK (M-ary Phase Shift Keying) modulation using GNU Radio, a software-defined radio (SDR) tool. The primary objective was to design an M-PSK transmitter, leveraging theoretical concepts such as complex envelope representation, symbol mapping, and spectral efficiency. Key steps included configuring modulation parameters, generating constellation diagrams, and evaluating the impact of noise on signal integrity. The study also compared bandwidth requirements and symbol rates between M-PSK and Q-PSK schemes. Through flowgraph implementation and signal analysis in time, frequency, and constellation domains, the practice reinforced the advantages of M-PSK in digital communications, particularly its spectral efficiency and robustness in noisy environments. Results highlighted the critical role of parameter optimization, such as samples per symbol (SPS), in minimizing aliasing and ensuring accurate signal representation.**

I. Introducción

La modulación M-PSK es una técnica fundamental en sistemas de comunicación modernos, empleada en aplicaciones como transmisiones satelitales, redes móviles y comunicaciones ópticas. Su operación se basa en variar la fase de una portadora para codificar información digital, lo que permite una alta eficiencia espectral y adaptabilidad a diferentes condiciones de canal. Este laboratorio busca implementar un transmisor M-PSK utilizando GNU Radio, profundizando en conceptos como la envolvente compleja, diagramas de constelación y parámetros críticos como el ancho de banda y la tasa de símbolos.

La práctica se enmarca en el contexto de comunicaciones digitales avanzadas, donde la representación en banda base (envolvente compleja) simplifica el procesamiento de señales al eliminar la dependencia de la frecuencia portadora. Siguiendo metodologías establecidas en laboratorios anteriores (e.g., conversión de RF a envolvente compleja para modulaciones como BPSK y FSK), este trabajo explora las particularidades de M-PSK, incluyendo su configuración mediante tablas de verdad y bloques VCO en GNU Radio. Además, se analiza el efecto del ruido en la integridad de la señal y se comparan métricas de rendimiento con otras técnicas de modulación, como Q-PSK.

II. Metodología

En cuanto al procedimiento realizado para la ejecución de la segunda práctica de laboratorio se procedio como sigue:

II-A. Implementación de M-PSK

La modulación M-PSK se implementó mediante dos enfoques: el método VCO y las tablas de verdad. En el primero, se utilizaron bloques VCO en GNU Radio para generar la señal modulada, ajustando parámetros como frecuencia portadora, fase inicial y tasa de símbolos. Paralelamente, se programó un vector source para mapear símbolos binarios a puntos específicos en la constelación M-PSK, asegurando fases equidistantes en el círculo unitario. Además, se extrajo la envolvente compleja mediante bloques especializados, evitando la necesidad de upconversión a RF y optimizando el procesamiento en banda base.

II-B. Análisis de señales

El análisis abarcó tres dominios: tiempo, frecuencia y constelación. En el dominio temporal, se visualizaron las formas de onda para verificar transiciones entre símbolos. En el espectral, se calcularon los PSD (Espectros de Densidad de Potencia) para determinar el ancho de banda y los cruces por cero, correlacionándolos con la tasa de símbolos. Los diagramas de constelación permitieron evaluar la distribución de puntos en el plano I-Q, identificando distorsiones por ruido y validando la precisión de la modulación.

II-C. Comparación con Q-PSK

Se repitió el proceso para Q-PSK, analizando diferencias en parámetros clave como ancho de banda, eficiencia espectral y robustez frente a interferencias. Se ajustaron variables como la desviación de fase y la frecuencia portadora, observando su impacto directo en la constelación y la integridad de la señal.

II-D. Validación y pruebas adicionales

Para validar la conversión a envolvente compleja. Adicionalmente, cada integrante diseñó una constelación no estándar, explorando configuraciones atípicas y su efecto en la detección de símbolos. Estas pruebas ampliaron la comprensión de los límites prácticos de la modulación y su adaptabilidad a condiciones no ideales.

II-E. Análisis de resultados y documentación

Se recopilaron capturas de pantalla y datos de cada experimento, los cuales fueron consignados en el informe. Además, se analizan los resultados obtenidos y su comportamiento.

III. Análisis de resultados

III-A. Implementación de M-PSK

El diseño del diagrama para M-PSK se basó en el tutorial [1], implementando como ejemplo una modulación 8-PSK. Para ello, se empleó un bloque Random Source que genera bits aleatorios (0 o 1), los cuales se agruparon en símbolos de 3 bits mediante reempaquetado. En la Figura 1, se muestra la gráfica de generación de la fuente y su conversión a símbolos.

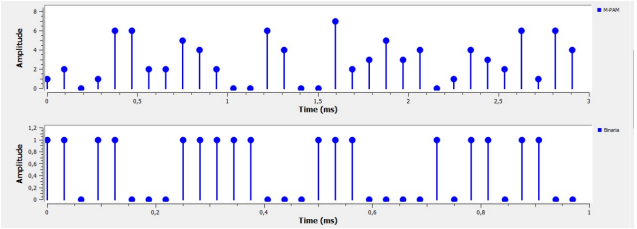

Figura 1. Fuente y Conversión a Simbolos.

Estos simbolos toman valores de 0 a 7, y estos se multiplican con $\frac{2\pi}{M}$ siendo M el numero de simbolos, así me dará una variación de angulo aleatorio dependiendo del simbolo empaquetado. Obteniendo la misma forma que el empaquetado solo que con una escala menor como se muestra en la Figura 2.

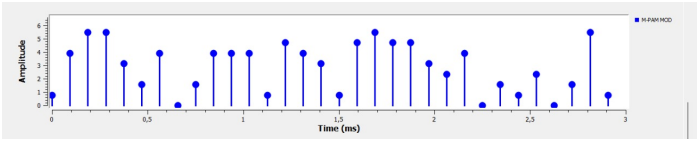

Figura 2. Escalamiento de los simbolos.

Se implementó un bloque en Python Figura 3 que asigna la fase de la señal utilizando valores aleatorios, mientras mantiene una amplitud constante, garantizando una modulación con escala uniforme. Obteniendo como salida la envolvente compleja.

```
import numpy as np
from gnuradio import gr
class blk(gr.sync_block):  # other base classes are basic_block,
decim_block, interp_block
    """Embedded Python Block example - a simple multiply const"""

    def __init__(self, example_param=1.0):  # only default arguments
here
        """arguments to this function show up as parameters in GRC"""
        gr.sync_block.__init__(
            self,
            name='VCO_complex',   # will show up in GRC
            in_sig=[np.float32, np.float32],
            out_sig=[np.complex64]
        )
    def work(self, input_items, output_items):
        Q=input_items[0]
        A=input_items[1]
        Sec=output_items[0]
        Sec[:]=A*np.exp(1.jQ)
        return len(Sec)
```

Figura 3. Bloque de Python.

Para tener una mejor visualizacion de la envolvente compleja se interpola cada dato como se muestra en la Figura 4, adicional se grafica el diagrama de constelación junto con el espectro en como se muestra en la Figura 5.

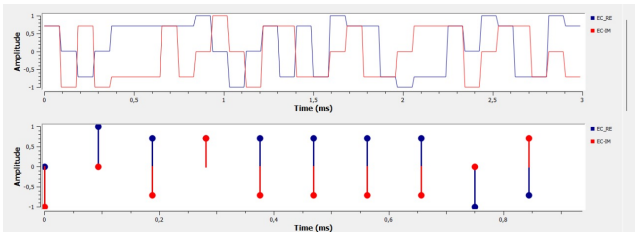

Figura 4. Envolvente compleja.

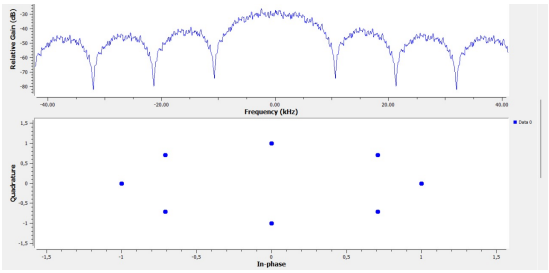

Figura 5. Diagrama de constelación y Espectro.

Para emular un entorno realista, se añadió ruido Gaussiano (con componentes real e imaginaria) al sistema. En la Figura 6 se observa su distribución espectral y su efecto en la forma de la señal, mientras que la Figura 7 muestra la envolvente compleja y la constelación resultante, donde los símbolos mantienen su estructura reconocible pero presentan dispersión debido al ruido.

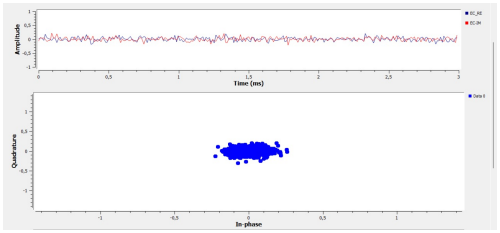

Figura 6. Señal del Ruido y Constelacion.

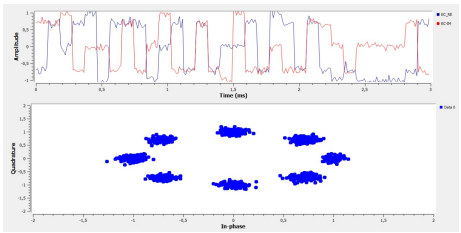

Figura 7. Envolvente Compleja y Constelación con Ruido.

Para implementar la modulación mediante una tabla de verdad, se utilizó una fuente aleatoria y un reempaquetado de bits, seguido del bloque Chunks to Symbols. Este bloque mapea cada símbolo a un valor complejo predefinido en una tabla de verdad (vector de números complejos), donde cada elemento tiene un desfase de 90° respecto al anterior, comenzando en

1+0j. Por ejemplo, el símbolo 000 se asigna a 1+0j, 001 a 0+1j, y así sucesivamente, generando una constelación rectangular. La señal resultante, su envolvente compleja y la constelación se muestran en la Figura 8. Posteriormente, se añadió ruido Gaussiano al sistema, cuyos efectos en la dispersión de los símbolos y la degradación de la constelación se ilustran en la Figura 9.

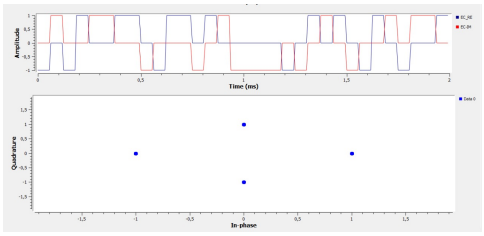

Figura 8. Envolvente compleja y Diagrama de Constelación.

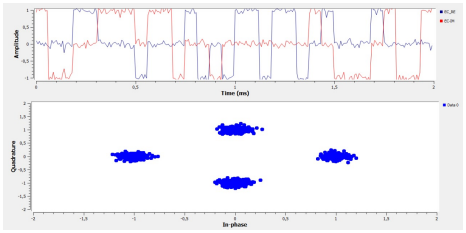

Figura 9. Envolvente compleja y Diagrama de Constelación con Ruido.

III-B. Análisis de Señales

El ancho de banda de la envolvente compleja se puede observar en la Figura 10, siendo $BW = 10{,}7kHz$. Sacado de la formula $\frac{Rb}{Nbps}$, siendo $R_b = 32kHz$ y $Nbps = 3$.

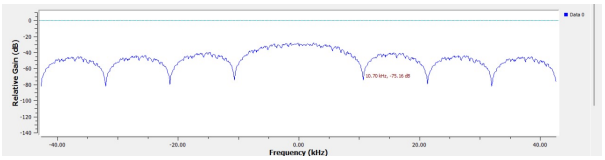

Figura 10. Ancho de Banda de la Envolvente Compleja.

El espectro (en frecuencia) de una señal digital con pulsos rectangulares pasa por cero en múltiplos de la rata de símbolos R_s.Esto se debe a que la transformada de Fourier de un pulso rectangular en el tiempo es una función sinc en frecuencia, así como se puede ver en las Figuras 11 y 12.

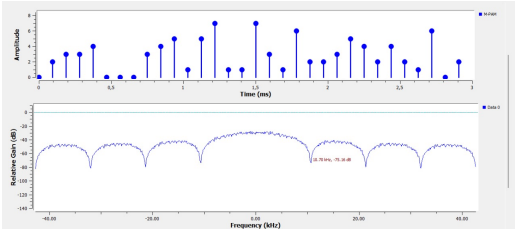

Figura 11. Valor 1 en el que el espectro pasa por cero.

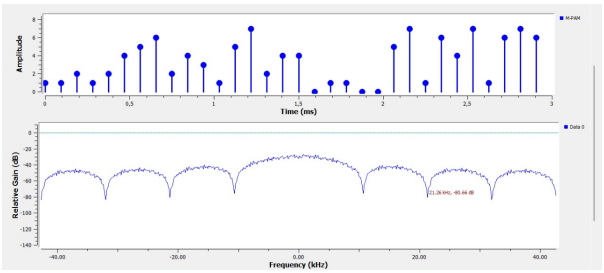

Figura 12. Valor 2 en el que el espectro pasa por cero.

Se reconfiguró el bloque Vector Source para generar una secuencia ordenada de valores del 0 al 7. Estos valores se asociaron a una tabla de verdad, definida como un vector de números complejos (Figura 13), donde cada posición corresponde a un símbolo con fase específica. La señal resultante, validada mediante su envolvente compleja y constelación, se muestra en la Figura 14, confirmando la correcta asignación de símbolos según la tabla definida.

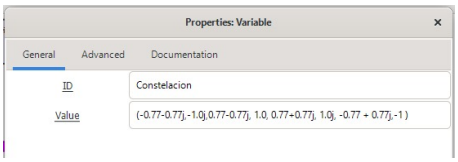

Figura 13. Vector tabla de verdad constelación.

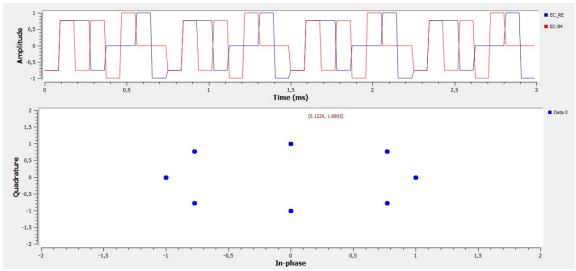

Figura 14. Envolvente compleja y constelación.

III-C. Comparación con Q-PSK

Se desarrolló un flujograma como se muestra en el video [2] que inicia con un Vector Source como fuente de datos, seguido por el bloque Repack Bits para agrupar bits en símbolos de tamaño definido. Estos símbolos se convirtieron a formato no empaquetado mediante Packed to Unpacked y luego se mapearon a valores complejos usando Chunks to Symbols, el cual utiliza una tabla de verdad (vector predefinido) para asignar fases específicas a cada símbolo. Antes de la visualización, la señal se interpoló para mejorar su resolución temporal, obteniendo así la envolvente compleja y el diagrama de constelación mostrados en la Figura 15, que confirman la correcta operación del sistema.

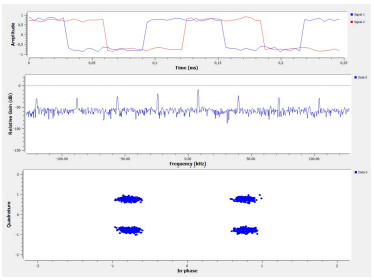

Figura 15. Envolvente Compleja y Constelación con ruido video 2.

III-D. Validación y pruebas adicionales

Cada integrante creó una constelación en especial para hacer las respectivas pruebas y se van mostrando los resultados. Los vectores de los 3 casos se encuentran en la Figura 16. En la primer caso de constelación creada se empieza en un angulo de $\frac{\pi}{3}$ con los resultados mostrados en la Figura 17. En la segundo caso de constelación creada se empieza en un angulo de $\frac{\pi}{9}$ con los resultados mostrados en la Figura 18. En la tercer caso de constelación creada se empieza en un angulo de $\frac{\pi}{8}$ con los resultados mostrados en la Figura 19.

k	Vector 1 (θ_0=π/3)	Vector 2 (θ_0=π/9)	Vector 3 (θ_0=π/8)
0	0.5000 + 0.8660j	0.9397 + 0.3420j	0.9239 + 0.3827j
1	−0.2588 + 0.9659j	0.3420 + 0.9397j	0.3827 + 0.9239j
2	−0.9659 + 0.2588j	−0.3420 + 0.9397j	−0.3827 + 0.9239j
3	−1.0000 + 0.0000j	−0.9397 + 0.3420j	−0.9239 + 0.3827j
4	−0.9659 − 0.2588j	−0.9397 − 0.3420j	−0.9239 − 0.3827j
5	−0.2588 − 0.9659j	−0.3420 − 0.9397j	−0.3827 − 0.9239j
6	0.5000 − 0.8660j	0.3420 − 0.9397j	0.3827 − 0.9239j
7	0.9659 − 0.2588j	0.9397 − 0.3420j	0.9239 − 0.3827j

Figura 16. Vectores de constelación de los 3 casos.

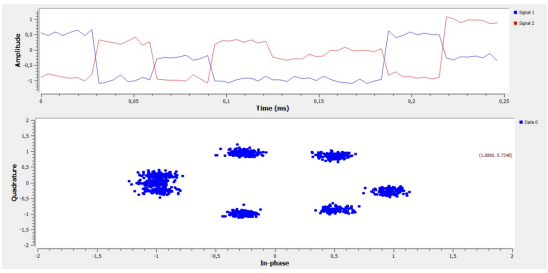

Figura 17. Caso 1.

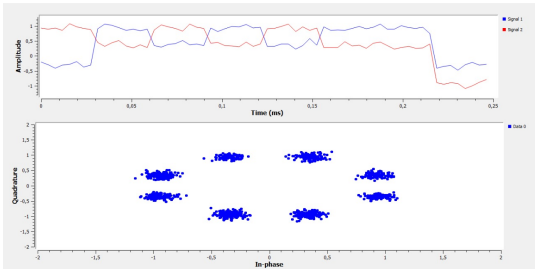

Figura 18. Caso 2.

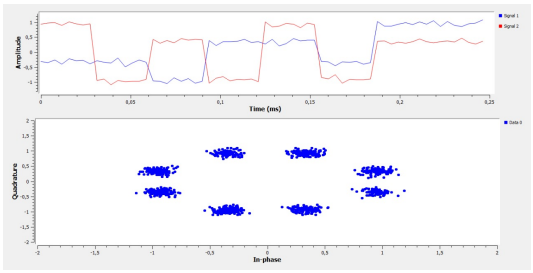

Figura 19. Caso 3.

Se puede observar que los angulos se van variando conforme a envolvente compleja y su componente real e imaginaria dadas por la tabla, adicional en algunos casos el ruido no dejar percibir la variación del angulo como lo es en el caso 2 y 3.

IV. Conclusiones

- La modulación M-PSK demostró alta eficiencia espectral, especialmente en su versión 8-PSK, al codificar múltiples bits por símbolo. Su adaptabilidad a entornos ruidosos se validó mediante la adición de ruido Gaussiano, donde, aunque se observó dispersión en la constelación Figuras 7 y 9, los símbolos mantuvieron su estructura reconocible, evidenciando su robustez en condiciones realistas.
- El uso de la envolvente compleja simplificó el procesamiento de señales al operar en banda base, eliminando la dependencia de la frecuencia portadora. Esto permitió visualizar claramente los símbolos en el plano I-Q Figuras 5 y 8) y analizar el ancho de banda ocupado, que cumplió con las expectativas teóricas Figura 10.
- Al implementar Q-PSK Figura 15, se confirmó que M-PSK ofrece mayor eficiencia espectral a expensas de una mayor complejidad en la detección de símbolos, especialmente en presencia de ruido. La elección entre esquemas dependerá de las condiciones del canal y los requisitos de ancho de banda..
- Las pruebas con constelaciones no estándar (Figuras 17-18-19) demostraron la adaptabilidad de M-PSK a configuraciones específicas. Sin embargo, se evidenció que desfases iniciales inadecuados ($\frac{\pi}{9}$) pueden aumentar la probabilidad de error, subrayando la importancia de un diseño coherente con las condiciones del sistema.
- Los resultados respaldan el uso de M-PSK en aplicaciones como comunicaciones satelitales y redes móviles, donde la eficiencia espectral y la tolerancia al ruido son prioritarias. La implementación en GNU Radio validó su viabilidad en sistemas de radio definida por software (SDR).

Referencias

[1] Video Tutorial ”Python Module - GNU Radio”Modulación 8PSK en GNU Radio usando dos métodos: VCO y Tabla de verdad”. Disponible en: https://www.youtube.com/watch?v=47FUTpV7y4A

[2] Video Tutorial Ïmplementacion de un modulador M-PSK en GNU Radio. En particular QPSK”. Disponible en: https://www.youtube.com/watch?v=2rsu-c26Tqo