



**udp** UNIVERSIDAD  
DIEGO PORTALES

UNIVERSIDAD DIEGO PORTALES  
ESCUELA DE INFORMÁTICA &  
TELECOMUNICACIONES

## COMUNICACIONES DIGITALES

---

### Laboratorio 4: BER en señalización de banda base

---

*Autores:*

Dante Hortuvia  
Maximiliano Juarez

Profesor: Marcos Fantoal

17 de junio de 2025

---

# Índice

<b>1. I. Introducción</b>	<b>2</b>
<b>2. II. Antecedentes</b>	<b>2</b>
<b>3. III. Metodología</b>	<b>2</b>
3.1. En que hace Byte pack y Byte Unpack . . . . .	2
3.2. Que hacen los bloques a utilizar en Gnu Radio . . . . .	3
3.3. Funciones de la clase gr::digital::constellation . . . . .	3
3.4. Construcción del transmisor . . . . .	4
3.4.1. como se construye el transmisor? . . . . .	4
3.4.2. como se simula el ruido? . . . . .	6
3.4.3. Como funciona el receptor? . . . . .	6
3.4.4. como se evalua la sistema? . . . . .	8
<b>4. IV. Resultados</b>	<b>9</b>
<b>5. VI. Análisis de Resultados</b>	<b>10</b>
5.1. Análisis de BPSK . . . . .	10
5.2. Análisis de QPSK . . . . .	10
5.3. Análisis de 8PSK . . . . .	11
<b>6. VII. conclusión</b>	<b>11</b>
<b>7. VIII. Referencias</b>	<b>12</b>
<b>8. IX.REPOSITORIO GITHUB</b>	<b>12</b>

---

## 1. I. Introducción

En este laboratorio buscamos que los datos viajen de un punto a otro, incluso cuando hay ruido, para lograr esto utilizaremos distintas técnicas de modulación que permiten adaptar los datos binarios a señales que puedan transmitirse por medios físicos, las variantes por desplazamiento de fase (PSK) son muy comunes por su eficiencia y simplicidad.

En este laboratorio nos enfocamos en tres de ellas: BPSK, QPSK y 8PSK, que difieren en cuántos bits transmiten por cada símbolo y en qué tan bien resisten el ruido, El objetivo principal fue ver cómo se comporta cada una de estas modulaciones al variar la relación señal a ruido y cómo esto afecta la tasa de error de bit (BER), mediante Gnu Radio.

## 2. II. Antecedentes

para transmitir información digital es necesario convertir esa información en una forma que pueda viajar a través de un canal físico, como un cable o el aire. Para eso se utilizan las técnicas de modulación digital, y una de las más utilizadas es la modulación por desplazamiento de fase, conocida como PSK.

Dentro de PSK, existen distintas versiones que cambian la cantidad de bits que representan por símbolo:

BPSK (Binary PSK): utiliza solo dos fases, y representa 1 bit por símbolo. Es simple, eficiente y muy resistente al ruido.

QPSK (Quadrature PSK): usa 4 fases diferentes (cada  $90^\circ$ ), representando 2 bits por símbolo. Transmite más información sin necesidad de usar más ancho de banda.

8-PSK: se basa en 8 fases (separadas  $45^\circ$ ), permitiendo enviar 3 bits por símbolo. Es más eficiente, pero más sensible al ruido, ya que las fases están más cerca entre sí.

Para medir qué tan bien funciona una modulación, se suele usar la tasa de error de bit (BER), que indica cuántos bits se recibieron mal respecto al total enviado. A medida que aumentamos la calidad de la señal en relación al ruido, esperamos que el BER disminuya.

## 3. III. Metodología

### 3.1. En que hace Byte pack y Byte Unpack

Estos bloques se utilizan para convertir datos en diferentes representaciones de bits

- 
- Unpacked : los bits son transmitidos individualmente como 0 y 1 , donde cada uno se utiliza para una muestra
  - Packed : Los bits son agrupados por bytes, al utilizar bytes el valor maximo que puede alcanzar es 255.

### 3.2. Que hacen los bloques a utilizar en Gnu Radio

- Random Source : genera una secuencia aleatoria entre numeros enteros, donde tiene parametros claves, como un mínimo , un máximo y un numero de muestras, esto se utiliza para simular una transmisión de datos binarios
- Chunks to Symbols : convierte los bits en símbolos complejos, donde se utilizaran para la modulación digital, estas conversiones se realizan utilizando tablas de símbolos definidos por esquemas de modulación predeterminados.
- Noise Source : este se utiliza para insertar ruido gaussiano, esto se utiliza para simular las condiciones de un canal real donde la señal transmitida se vera afectada por este, la amplitud del ruido es configurable, lo que nos ayuda a varia la energía por bit y densidad espectral de ruido, gracias a esto podemos observar como sera el desempeño del sistema.
- Constellation Decoder : se utiliza para demodulación de la señal recibida, toma los símbolos complejos obtenidos y los compara con los puntos válidos de la señal por el esquema de modulación, gracias a esta se determinan los símbolos más probables donde se reconstruyen los bits originales, gracias a esto se puede hacer el cálculo BER, lo que nos permite decodificar la señal.
- BER : este calcula la tasa de error de bits en la transmisión. Compara la secuencia original de bits anterior a la demodulación con la secuencia de bits recuperada después de la demodulación.
- QT GUI Number Sink : permite mostrar gráficamente el valor del BER calculado.

### 3.3. Funciones de la clase `gr::digital::constellation`

- Funcion `Arity()` : retorna cantidad de símbolos distintos que pueden representar una constelación determinada. Este es fundamental para identificar el tipo de modulación empleado.
- Funcion `Point()` : retorna las coordenadas complejas que representan cada punto de la constelación en el plano IQ. Estos puntos definen cómo se codifican y decodifican los datos dentro del sistema, y determinan la geometría de la modulación.
- Funcion `Base()` : define la base numérica utilizada para representar los símbolos. Normalmente se utiliza la base binaria

- Funcion `Bits_per_symbol()` : Muestra cuántos bits están asociados a cada símbolo de la constelación. Este valor está directamente relacionado con la eficiencia espectral del sistema.

### 3.4. Construcción del transmisor

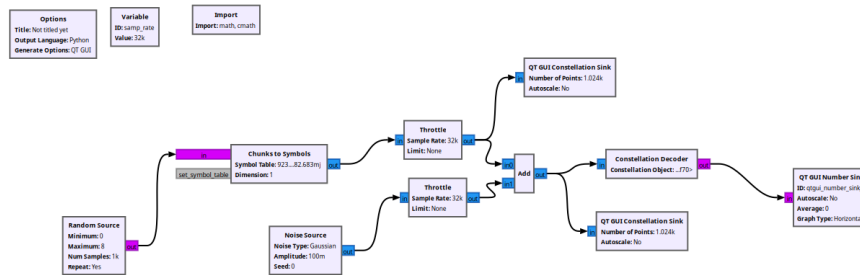


Figura 1: Implementación del circuito en Gnu radio.

#### 3.4.1. como se construye el transmisor?

se necesita implementar una fuente de datos para enviar una señal. Para eso usamos el bloque `Random Source`, que genera números aleatorios entre 0 y 8 (suficientes para representar las tres modulaciones pedidas: BPSK usa 2 símbolos, QPSK usa 4 y 8PSK usa 8). Este bloque genera mil valores continuamente y reinicia cuando llega al final, simulando una transmisión constante de datos.

Después, esos números pasan por el bloque `Chunks to Symbols`, que se encarga de transformar los datos en símbolos complejos que representan puntos en un plano (la constelación). Este bloque es fundamental porque define si estamos transmitiendo en BPSK, QPSK u 8PSK, dependiendo de qué tabla de símbolos usemos. Es decir, con este bloque decidimos qué tipo de modulación aplicamos en cada prueba.

Properties: Chunks to Symbols

General

Advanced

Documentation

Input Type

byte

Output Type

complex

Symbol Table

[-1,1]

[complex\_vector]

Dimension

1

[int]

Num Ports

1

[int]

Aceptar

Cancelar

Aplicar

Figura 2: Chunks to Symbols BPSK.

Properties: Chunks to Symbols

General

Advanced

Documentation

Input Type

byte

Output Type

complex

Symbol Table

$$\left[ \frac{(1+j)}{\sqrt{2}}, \frac{(-1+j)}{\sqrt{2}}, \frac{(-1-j)}{\sqrt{2}}, \frac{(1-j)}{\sqrt{2}} \right]$$

[complex\_vector]

Dimension

1

[int]

Num Ports

1

[int]

Aceptar

Cancelar

Aplicar

Figura 3: Chunks to Symbols QSPK.

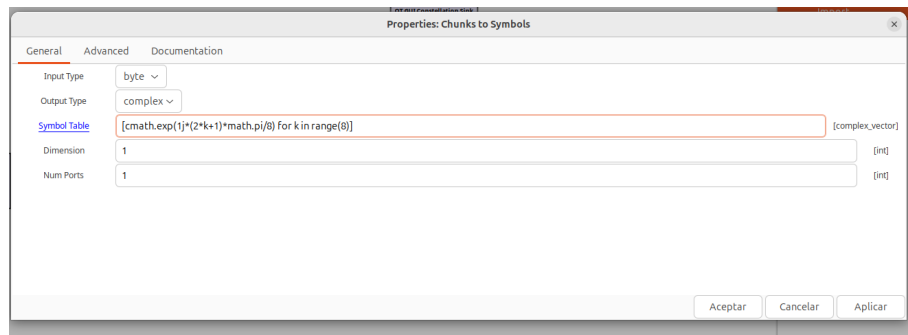


Figura 4: Chunks to Symbols 8SPK.

### 3.4.2. como se simula el ruido?

se implementa el bloque Noise Source, que genera un ruido de tipo gaussiano (como el que afecta las señales en un canal real). Este ruido se suma a la señal transmitida para simular un canal con interferencias.

Antes de sumarlas, tanto la señal como el ruido pasan por un bloque llamado Throttle que regula la velocidad de procesamiento (32 kHz), para que la simulación no sobrecargue el computador. Después de eso, ambas señales llegan al bloque Add, que las combina y genera la señal final que va al receptor.

### 3.4.3. Como funciona el receptor?

Para implementar El receptor se utiliza un bloque de Constellation Decoder, que trata de adivinar qué símbolo fue enviado basándose en la posición de cada punto en el plano de la constelación. Es decir, este bloque toma las decisiones sobre qué datos originales recibió, a pesar del ruido que afectó la señal.

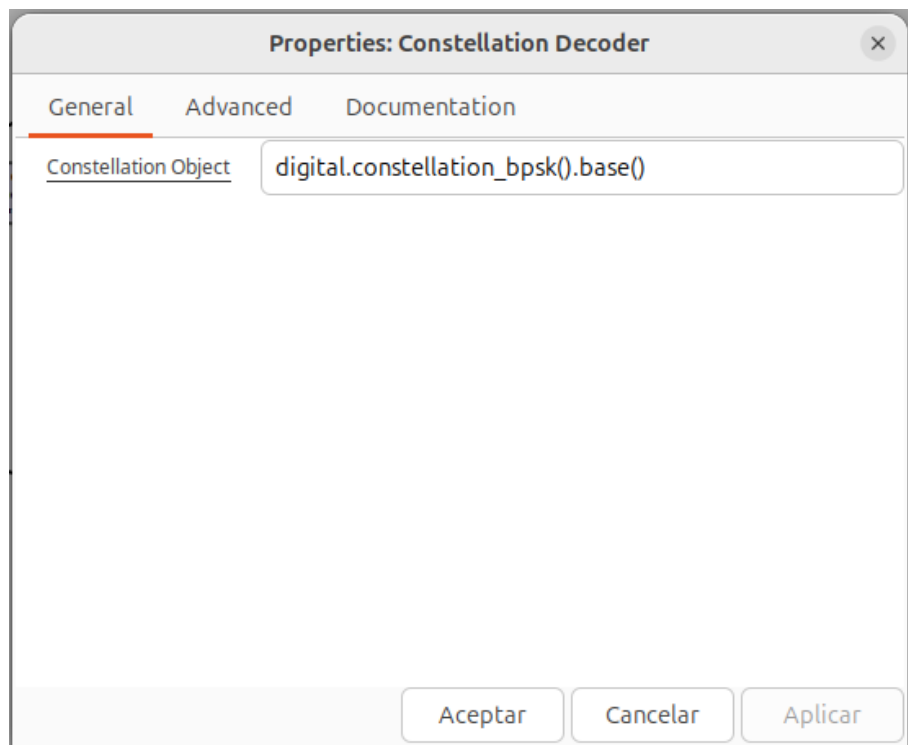


Figura 5: constellationnn decore BPSK.

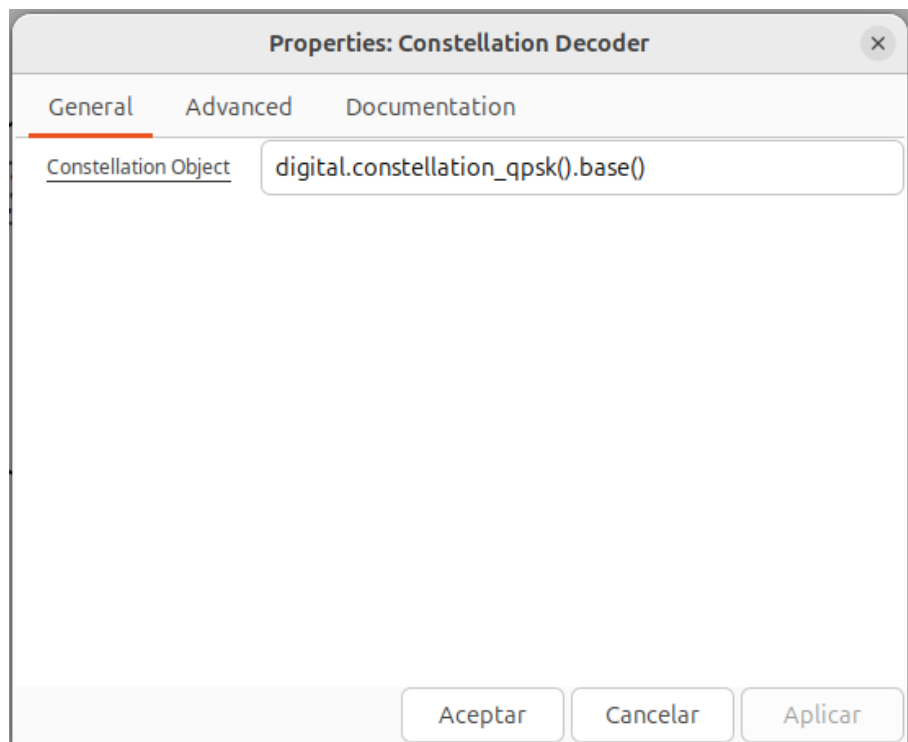


Figura 6: constellationnn decore QSPK.



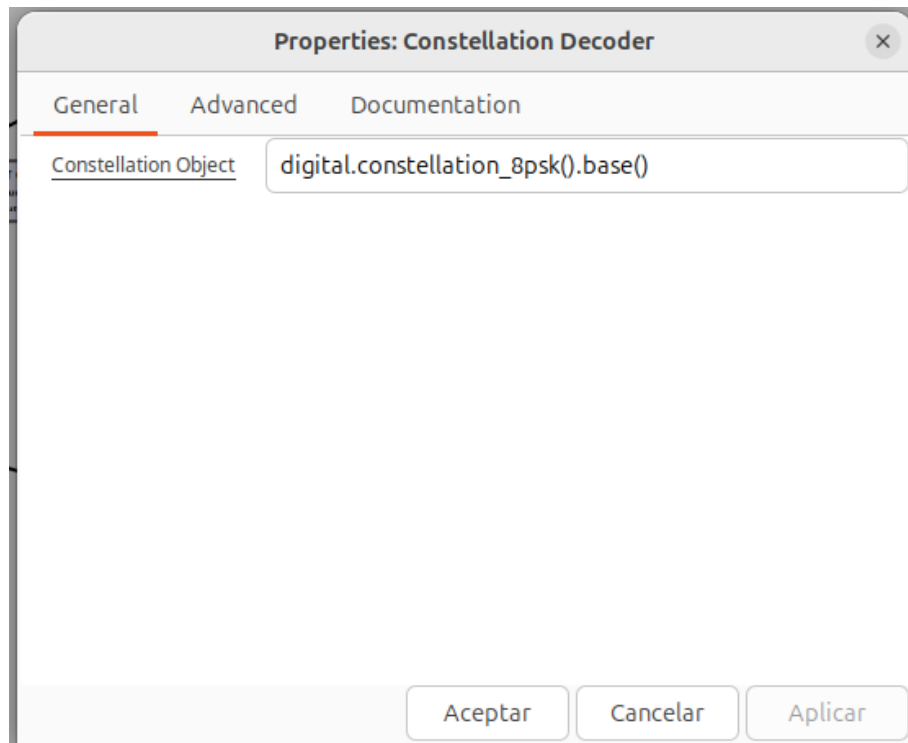


Figura 7: constellationn decoreos 8SPK.

También se agregaran dos bloques de visualización de constelación : uno antes y otro después de sumar el ruido. Así se puede ver cómo se afecta la constelación a causa del ruido, y cuánto afecta la transmisión. Además, con un bloque QT GUI Number Sink se observa el resultado promedio de los símbolos detectados, como un control adicional para saber cómo va el sistema.

#### 3.4.4. como se evalua la sistema?

Para evaluar cuántos errores ocurren, debe incorporarse un bloque que calcule el BER (Bit Error Rate). Este bloque compara bit a bit la señal enviada con la que recibió el receptor, y cuenta cuántos bits fueron mal interpretados.

para variar la amplitud del ruido gaussiano se utilizan valores de 1 a 11 dB. donde se puede observar cómo cambian los errores según el nivel de interferencia del canal. Esta medición debe repetirse para las tres modulaciones (BPSK, QPSK y 8PSK), cambiando la tabla de símbolos en el bloque Chunks to Symbols en cada caso. Finalmente, con esos datos se puede armar una gráfica que muestre cómo varía el BER para cada tipo de modulación.

## 4. IV. Resultados

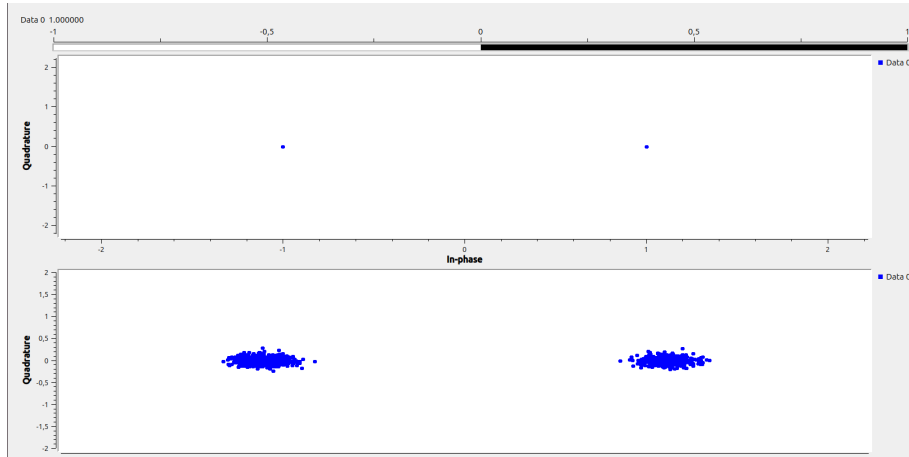


Figura 8: Implementacion del circuito en Gnu radio.

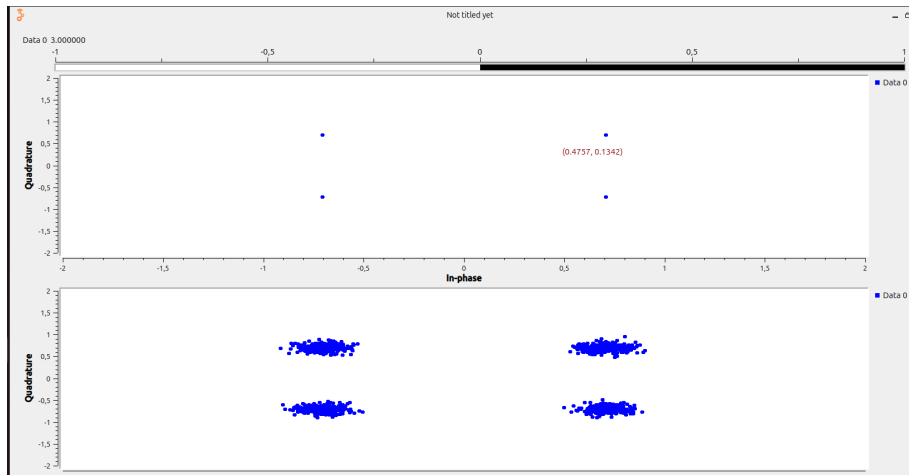


Figura 9: Implementacion del circuito en Gnu radio.

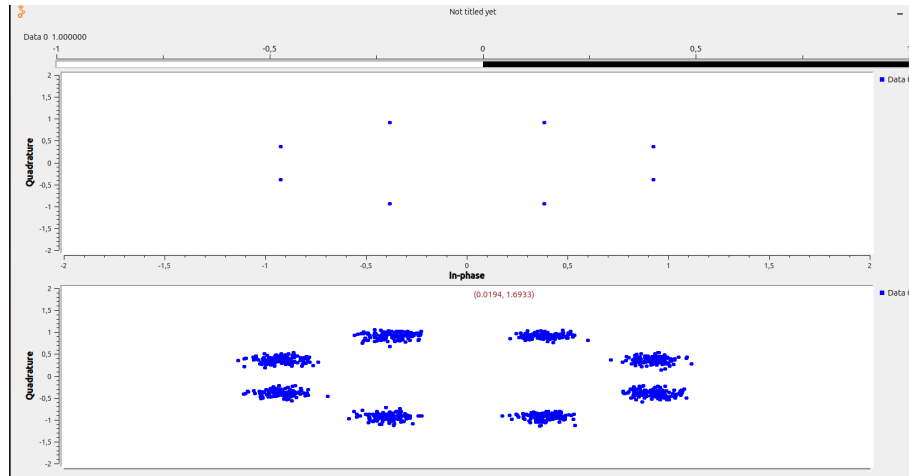


Figura 10: Implementacion del circuito en Gnu radio.

## 5. VI. Análisis de Resultados

### 5.1. Análisis de BPSK

En la Figura 8 , la constelación ideal solo contiene dos puntos, uno en -1 y otro en +1 sobre el eje In-phase (en fase). Esta simplicidad es gracias al funcionamiento de BPSK. Lo que ocurre aquí es que cada símbolo transmite únicamente 1 bit, y la única diferencia entre ellos es un cambio de  $180^\circ$  en la fase de la señal. Es decir, o bien se envía la señal "tal cual" o invertida.

Un detalle interesante es que no hay componente en el eje Quadrature (cuadratura), lo que significa que toda la modulación se realiza exclusivamente en el eje de fase. Esto hace que el sistema sea más fácil de implementar y analizar.

la gráfica inferior, se ve que estos dos puntos ideales se convierten en dos nubes de puntos alargadas, se observa que claramente están separadas. Esta separación amplia entre símbolos es lo que le da a BPSK una gran robustez frente al ruido. Es bastante fácil distinguir si se trataba de un 0 o un 1. Por eso, BPSK es una excelente opción para comunicaciones en entornos difíciles o cuando la prioridad es minimizar los errores de transmisión, aunque eso signifique sacrificar velocidad de datos.

### 5.2. Análisis de QPSK

En la Figura 9 muestra una constelación ideal con cuatro puntos, distribuidos en forma rectangular y ubicados cerca de las coordenadas  $(\pm 0.7, \pm 0.7)$  en el plano In-phase-Quadrature. A diferencia de BPSK, aquí cada símbolo representa 2 bits, lo que permite transmitir el doble de información en el mismo tiempo. Para lograrlo, se utilizan cuatro fases diferentes.

---

En este caso, los símbolos tienen componentes tanto en el eje In-phase como en el de Quadrature, lo cual da como resultado una modulación más compleja, pero también más eficiente desde el punto de vista espectral. Es decir, se puede enviar más información sin necesitar más ancho de banda.

Al observar la gráfica inferior con ruido, se aprecia que cada punto ideal se convierte en una nube dispersa, pero las cuatro nubes siguen estando razonablemente separadas. Esto permite que el receptor identifique correctamente la mayoría de los símbolos, aunque haya algo de distorsión. QPSK ofrece un equilibrio entre cantidad de información enviada y robustez; transmite más datos que BPSK, sin ser demasiado vulnerable al ruido.

### 5.3. Análisis de 8PSK

en la Figura 10, la constelación se compone de ocho puntos distribuidos uniformemente. Esto se debe a que cada símbolo codifica 3 bits, lo que representa un gran salto en eficiencia con respecto a QPSK y BPSK. Sin embargo, ese incremento de eficiencia viene con un costo.

Como hay más puntos en el mismo espacio, la distancia entre ellos se reduce. Al añadir ruido, como vemos en la gráfica inferior, estas ocho nubes de puntos se acercan entre sí. Esto nos indica que, con una pequeña perturbación, un punto puede afectar la zona de decisión de un símbolo incorrecto. Esto nos da una mayor probabilidad de error.

8PSK permite transmitir más datos por símbolo, pero es más frágil ante el ruido. Esta modulación es útil cuando el canal es relativamente limpio o cuando se pueden aplicar técnicas de corrección de errores que compensen su vulnerabilidad.

## 6. VII. conclusión

El laboratorio permitió comprender de manera práctica cómo distintas modulaciones por desplazamiento de fase (PSK) ofrecen diferentes niveles de eficiencia y resistencia frente al ruido. La modulación BPSK se destacó por su gran robustez, manteniendo una clara separación entre los símbolos incluso en presencia de interferencias, lo que la hace ideal para entornos ruidosos. Por otro lado, 8PSK, si bien entrega una mayor eficiencia al transmitir más bits por símbolo, resultó ser más vulnerable al ruido debido a la menor distancia entre los puntos de su constelación. En medio de estos dos extremos, QPSK demostró ser un buen compromiso, equilibrando eficiencia espectral y confiabilidad.

La utilización de GNU Radio fue clave para implementar y analizar estos esquemas, ya que permitió visualizar de forma directa cómo el ruido distorsiona

---

las constelaciones y facilitó la medición de la tasa de error de bits (BER) en cada caso.

## **7. VIII. Referencias**

- Tomasi, W., Sistemas de comunicaciones electrónicas, Pearson, 2003.
- Couch, L., Sistemas de comunicaciones digitales y analógicos, Pearson, 2008.
- Guía de Laboratorio CIT2111 — BER en señalización de banda base, UDP.
- Documentación de GNU Radio Companion: [https://www.gnuradio.org/doc/doxygen/classgr\\_1\\_1digital\\_1\\_1constellation.html](https://www.gnuradio.org/doc/doxygen/classgr_1_1digital_1_1constellation.html)

## **8. IX.REPOSITORIO GITHUB**

REPOSITORIO GITHUB