

## Prácticas de comunicaciones móviles

Cada practica debe realizarse en lenguaje de programación de Python.

### 1. Implementación de un modulador/demodulador OFDM

El script debe trabajar sobre una imagen cargada en el computador, el cual, se debe pasar a blanco y negro y posteriormente pasarlo a bits. Por ejemplo; una imagen .png de dimensiones 512 x 512 pixeles debería ser transformada a una cadena de bits donde cada pixel tiene 8 bits, entonces se tendría un total de 512 x 512 x 8 bits. Este seria el total de bits de información.

El script debe permitir ingresar mediante teclado:

- Ancho de banda en Hz:  $BW$  (generalmente probamos con 10MHz)
- Espaciamiento entre subportadoras:  $\Delta f$  solo puede ser o 15KHz o 7.5KHz.
- Valor de SNR (dB)
- Escoger Prefijo ciclico:
  - Normal:  $4.6 \mu s$
  - Extendido:  $16.6 \mu s$  para el caso de que  $\Delta f = 15KHz$   
 $33 \mu s$  para el caso de que  $\Delta f = 7.5KHz$
- Porcentaje de portadoras piloto: por ejemplo, del total de portadoras quiero asignar 5% de esas para que sean piloto.
- Seleccione el tipo de modulación:
  - QPSK
  - 16-QAM
  - 64-QAM

Ya dentro del script y en base a los valores ingresados mediante teclado se calculan otras variables necesarias para llevar a cabo la práctica tales como:

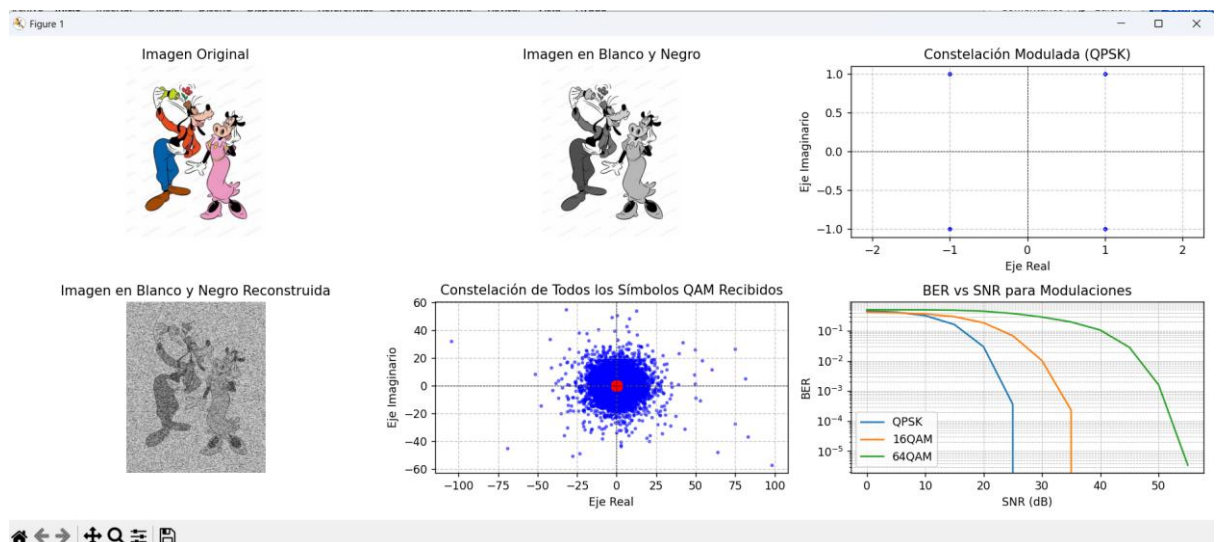
- Calcular el número de subportadoras  $N_c = \frac{BW}{\Delta f}$
- Calcular el tamaño de la IFFT ( $N$ ) como potencia de 2 donde  $N > N_c$
- Realizar el proceso que implica OFDM (agregar prefijo cíclico, aplicar IFFT, realizar estimación de canal, realizar ecualización) (material de las diapositivas).
- Importante para la transmisión simular un canal móvil, puede ser multicamino o retardos.
- En el lado del receptor la imagen reconstruida no debe tener patrones en la imagen, es decir, no debe haber ruido tipo franjas verticales, horizontales o inclinados ya que se supone que el ruido es aleatorio.

Como resultados mínimos de esta práctica se requiere:

- Obtener imagen en blanco y negro transmitida
- Obtener imagen reconstruida
- Obtener la gráfica de BER vs SNR de cada modulación en un mismo gráfico.

Se debe mostrar algo así para un ejemplo con los siguientes datos de entrada:

- Ancho banda 10MHz
- Espaciado 15khz
- Prefijo cíclico normal
- SNR 10 dB
- Modulación: QPSK

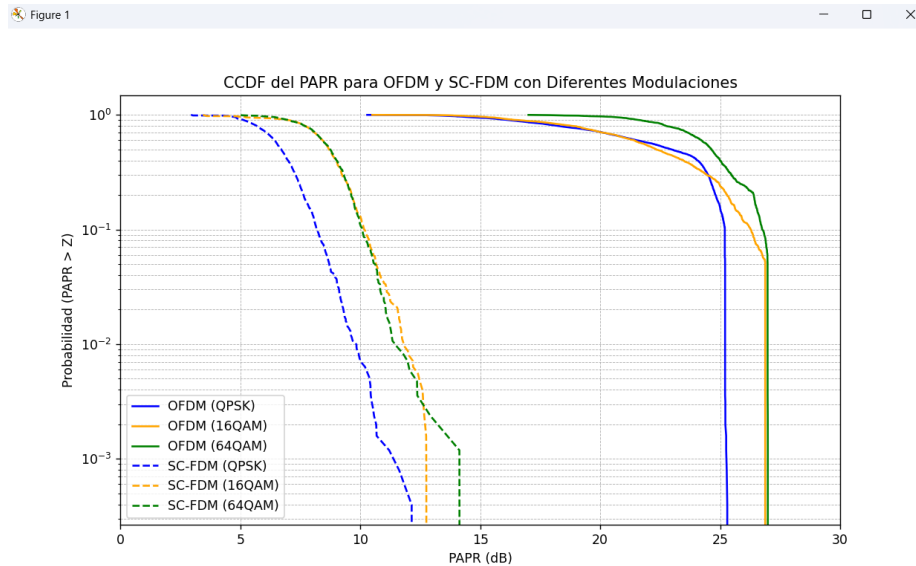


## 2. Implementación de un modulador/demodulador SC-FDM

En base a la práctica 1, aquí solo se debería implementar un bloque extra de pre-codificación que es aplicar DFT antes de enviar al modulador OFDM de la práctica 1.

Como resultados mínimos de esta práctica se requiere:

- Obtener imagen en blanco y negro transmitida
- Obtener imagen reconstruida
- Obtener la gráfica de CCDF de PAPR de cada modulación y para cada caso: OFDM y SC-FDM. Esta grafica es algo como la que se muestra a continuación

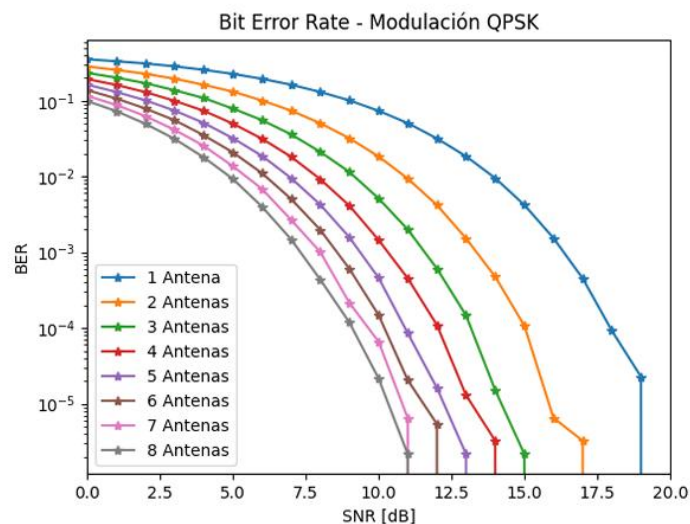


### 3. Implementar un sistema de comunicación con Diversidad en RX

También en base a la práctica 1. Se debe realizar la diversidad en RX para al menos 4 antenas en RX. Se debe utilizar la lógica de Maximum-Ratio Combination MRC.

Como resultados mínimos de esta práctica se requiere:

- Obtener imagen en blanco y negro transmitida
- Obtener imagen reconstruida normal sin diversidad en RX (es decir la imagen que se reconstruye en la Práctica 1)
- Obtener la imagen reconstruida usando 2 antenas y obtener la imagen reconstruida usando 4 antenas.
- Obtener la gráfica de BER vs SNR de cada antena en una misma gráfica. Se debería ver algo así (imagen para caso de 8 antenas)



#### **4. Implementación de un esquema de diversidad en TX SFBC**

También en base a la practica 1. Se debe implementar diversidad en TX con 2 antenas en TX usando la lógica de SFBC Space–Frequency Block Coding / Space– Frequency TX Diversity.

Como resultados mínimos de esta práctica se requiere:

- Obtener imagen en blanco y negro transmitida
- Obtener imagen reconstruida normal sin diversidad en TX (es decir la imagen que se reconstruye en la Práctica 1)
- Obtener la imagen reconstruida usando 2 antenas (con diversidad)
- Obtener la gráfica de BER vs SNR de cada antena en una misma gráfica. Grafica igual a la de la práctica 3.
- Gráfica de BER vs SNR al usar OFDM simple, OFDM con diversidad en RX y OFDM con diversidad en TX. En un mismo grafico todo esto.

#### **5. Implementación de un sistema de TX con beamforming**

También es en base a la práctica 1. Implementar beamforming en TX con 4 a 8 antenas.

Como resultados mínimos de esta práctica se requiere:

- Obtener imagen en blanco y negro transmitida
- Obtener imagen reconstruida normal sin beamforming (es decir la imagen que se reconstruye en la Práctica 1)
- Obtener la imagen reconstruida usando beamforming (4 u 8 antenas)
- Obtener la gráfica de BER vs SNR de cada antena en una misma gráfica. Grafica igual a la de la práctica 3.
- Gráfica de BER vs SNR al usar OFDM simple, OFDM con diversidad en RX y OFDM con diversidad en TX y Beamforming. En un mismo grafico todo esto

#### **6. Implementación de un sistema MIMO — Multiplexación Espacial en Python**

También es en base a la práctica 1. Implementar un sistema puede ser cualquiera de estos:

- 8x2
- 8x4
- MU-MIMO

- SU-MIMO

por ejemplo: 8X2 :8 antenas en transmisor y 2 antenas en receptor. Por ejemplo: En un sistema 8x2, el número máximo de flujos independientes es  $\min(8,2)=2$ . Por lo tanto, es posible transmitir dos flujos de datos separados.

Se requiere:

- Enviar los flujos en paralelo NO SECUENCIAL
- Evaluar los tiempos de ejecución de transmisión usando sistema MIMO vs tiempo de ejecución de transmisión de práctica 1. El tiempo de ejecución del sistema MIMO debe ser menor al tiempo de la práctica 1.
- Obtener imagen reconstruida normal sin MIMO (es decir la imagen que se reconstruye en la Práctica 1)
- Obtener la imagen reconstruida usando MIMO

## **7. Implementar un codificador y un decodificador de canal especificado en el estándar LTE.**

Solo nos dijo que se puede realizar usando la librería de Python llamada CommPy.