



TALLER

Comunicación MIMO Multi-usuario con Software Defined Radio (SDR)

Guía del Instructor

**TELECO
RENTA**

PLAN DE
PROMOCIÓN DE LOS ESTUDIOS
DE TELECOMUNICACIÓN



Financiado por
la Unión Europea
NextGenerationEU



Plan de Recuperación,
Transformación
y Resiliencia



Disclaimer

Este taller ha sido preparado por el Centre Tecnològic de Telecomunicacions de Catalunya (CTTC), en el contexto del Plan de promoción de los estudios de Telecomunicación (programa UNICO 5G I+D).

El contenido de este taller, incluyendo el código fuente, guía del instructor y guía del estudiante se pone a disposición de uso de cualquier persona o entidad. No se permite sin embargo reutilizar o modificar su contenido sin el consentimiento explícito del CTTC.

Contacto: telecorenta@cttc.cat

Taller preparado por Adrián Agustín, Mónica Navarro.

Historia del documento

Revisión	Fecha	Secciones afectadas	Cambios
V1.0	08/07/2024	Todas	1ª edición

ÍNDICE

1.	Objetivos del taller.....	5
2.	Consideraciones preliminares	5
4.1	SDR ¿Qué es?	5
4.2	Configuración de los equipos	6
4.3	Configuración del USRP Hardware Driver (UHD)	6
4.4	Configuración del Matlab: Communication Toolbox Support Package for USRP radio	7
4.5	Configuración de los dispositivos.....	8
4.6	Puesta en marcha entorno de experimentación	11
4.7	Código Matlab.....	11
4.8	Descripción de las funciones básicas de Matlab en las que se sustenta la práctica.....	12
4.9	Entendiendo el código del transmisor	13
4.10	Entendiendo el código del receptor	14
3.	Modelo de señal.....	15
4.1	Trama transmitida.....	15
4.2	Sincronización en tiempo (inicio de la trama).....	16
4.3	Estimación de canal	16
4.4	OFDM	17
4.5	Precodificación MIMO OFDM	18
4.6	Precodificación MIMO en este Taller	18
4.	Ejercicios prácticos	20
4.1	Experimento 1: Utilización del precodificador Zero Forcing	20
4.2	Experimento 2: Utilización del precodificador Matched Filter	24
4.3	Experimento 3: Utilización de múltiples antenas en recepción	25
5.	Referencias	28

Lista de Tablas y Figuras

Tabla 1. Características de las USRPs X310.	6
Tabla 2. Instalación del driver UHD y conectividad con las SDR X310	7
Figura 1. Componentes UHD. Figura obtenida de [ETT-UHD].....	7
Figura 2. Integración de SDR con Matlab.....	8
Figura 3. Ejemplo de pasos a seguir durante el proceso de instalación del Communication Toolbox Support Package for USRP radio	8
Figura 4. Entorno de laboratorio en el CTTC	9
Figura 5. Interconexión del SDR y Octoclock para compartir el sincronismo.....	10
Figura 6. Modelo de comunicaciones	10
Figura 7. Puesta en marcha de las sesiones de Matlab en el servidor. Comprobación de que las SDRs están disponibles.....	11
Figura 8. Descripción general de las principales funciones para realizar la transmisión	13
Figura 9. Descripción general de las principales funciones para realizar la recepción de la señal en los usuarios.....	14
Figura 10. Trama de señal transmitida.....	15
Figura 11. Asignación de los recursos a los diferentes usuarios en términos de portadoras y tiempo. La transmisión utiliza recursos compartidos (portadoras 4-12) y recursos ortogonales (portadoras 49-100 para UE1 y 149-200 para el UE2). Los recursos en "naranja" son destinados para la transmisión de datos, mientras que los recursos en "verde" transmiten secuencias conocidas (midambles) por parte del receptor, que ayudarán a mejorar la estimación del canal en recepción. En los recursos en blanco no se transmite ninguna señal.....	16
Figura 12. Ejemplo de constelaciones recibidas en el Experimento 1	21
Figura 13. Canal estimado en los diferentes frames en el Experimento 1	22
Figura 14. Error de fase medido para las portadoras 70-170 para el UE1 (izquierda) y 170-200 para el UE2 (derecha).....	22
Figura 13. Ejemplo de constelaciones recibidas en el Experimento 2	25
Figura 16. Ejemplo de constelaciones recibidas en el Experimento 3	26
Figura 17. Autovalores de los canales del UE1 y UE2 en el Experimento 3	27

Acrónimos

DFT	Discrete Fourier Transform
IDFT	Inverse Discrete Fourier Transform
FFT	Fast Fourier Transform
MIMO	Multiple Input Multiple Output
MMCX	Micro-miniature coaxial connector
OFDM	Orthogonal Frequency Division Multiplexing
RF	Radio Frecuencia
SDR	Software Defined Radio
SFP+	Small Form-factor Pluggable plus
SW	Software
UHD	USRP Hardware Driver
USRP	Universal Software Radio Peripheral
...	

1. Objetivos del taller

El objetivo principal de este taller es familiarizarse con la tecnología SDR - Software Defined Radio o radio definida por software - y su aplicación en el contexto de las comunicaciones inalámbricas en redes celulares.

El taller consta pues de dos elementos principales:

- El hardware adaptado a la tecnología SDR (o USRP)
- La herramienta de simulación Matlab (programa SW) y el código desarrollado en Matlab

En particular este taller explora la comunicación multiusuario con múltiples antenas (MIMO, Multiple Input Multiple Output). Al finalizar el taller se espera que el estudiante tenga:

- Motivación por adentrarse en la experimentación con señales transmitidas a través del aire por ondas electromagnéticas, así como haya adquirido familiarización con el procesamiento digital de dichas señales en el receptor.

Para la realización del taller se ha partido de unos ejemplos estándar disponibles en Matlab para la comunicación multiusuario. Se proporcionan una serie de ejercicios o pruebas guiadas que permitirán al estudiante experimentar en un entorno con equipos que emulan una comunicación real, trabajando en tiempo real.

El presente documento se estructura en dos secciones principales. La sección 2 (Consideraciones preliminares) donde se hace una introducción a la tecnología SDR y su configuración. Además, se introducen las funciones de Matlab que son importantes para interactuar con SDRs, y se hace una explicación del código que se va a utilizar. El modelo de señal considerado se presenta la sección 3 (Modelo de Señal). La sección 4 (Ejercicios prácticos) propone una serie de ejercicios que los estudiantes deben llevar a cabo. En la siguiente tabla se detalla qué secciones son más importantes para un instructor o un estudiante.

2. Consideraciones preliminares

Esta sección introduce conceptos básicos de la tecnología SDR, así como establece la configuración necesaria previa a la realización del taller.

La sección también proporciona una descripción de la infraestructura experimental necesaria para la realización del taller. Dichos equipos están a disposición para la realización del taller en las instalaciones del CTTC, previo acuerdo por razones organizativas.

La configuración puede reproducirse en otros laboratorios si se dispone del hardware necesario, así como la correspondiente licencia de Matlab.

4.1 SDR ¿Qué es?

Las radios definidas por software [SDR] se refieren a un sistema de radiocomunicaciones con diferentes componentes genéricos que permite la transmisión de una señal generada digitalmente a través de dichos componentes (mezcladores, filtros, detectores, ...). Dichos componentes son los

encargados de general la señal que se transmitirá por el espacio físico a través de un interfaz de radiofrecuencia (RF).

El hecho de ser programable vía software permite una gran flexibilidad para configurar los experimentos que se pueden realizar, y por ello se convierten en una pieza fundamental para realizar experimentos con hardware y en entornos más realistas considerando la transmisión a través del aire.

Las radios definidas por software se componen de tres bloques principales: el **front-end**, que se encarga de recibir la señal y amplificarla de forma adecuada, la **conversión** a una frecuencia intermedia (a veces no es necesario), y la **conversión analógico/digital** para tener la señal en banda base (el proceso es inverso si la señal se debe transmitir en vez de recibir).

4.2 Configuración de los equipos

Los equipos utilizados en este taller consideran los modelos de USRPs existentes en el laboratorio del CTTC. En particular, las URSP responden al modelo X310, cuyas características principales se listan en la Tabla 1.

Placas	Dos slots con ancho de banda de 80 MHz cada uno Placa base cubre las frecuencias: DC a 6 GHz
Procesador de señal digital (DSP)	Xilinx Kintex-7 FP, gran capacidad de configuración para altas prestaciones del DSP (XC7K410T)
Interfaces	Múltiples interfaces de alta velocidad Dual 10 Gigabit Ethernet - 2x RX at 200 MSPs por canal Dual 1 Gigabit Ethernet - 25 MS/s Full Duplex
Arquitectura UHD	La arquitectura UHD permite compatibilidad con GNU Radio C++/Python API Otros entornos y programas (SW) de terceros
Reloj	Arquitectura de distribución de reloj flexible Tiempo de muestreo configurable Operación coherente con OctoClock
Otras	Factor de forma 1U robusto (mitad de ancho) y compacto para escritorio o montaje en rack. Acceso a las entradas y salidas digitales (Digital I/O): desde el panel frontal para control personalizado o vía las interfaces desde la FPGA

Tabla 1. Características de las USRPs X310.

Previamente a la realización del taller se deben configurar los siguientes drivers y toolboxes de Matlab.

4.3 Configuración del USRP Hardware Driver (UHD)

El **USRP Hardware Driver (UHD)** es la pieza clave que permite la comunicación con todos los productos USRP SDR de Ettus / National Instruments. Además, UHD permite la compatibilidad con otros entornos de desarrollo como RFNoC, GNU Radio, LabView y Matlab/Simulink.

Este driver puede ejecutarse en cualquier procesador de propósito general, y permite el control necesario para transmitir la señal desde y hacia la SDR. Además de permitir configurar otros parámetros como la frecuencia de muestreo, la ganancia o la frecuencia RF [ETT-UHD].

La Figura 1 muestra un esquema representativo de la arquitectura UHD.

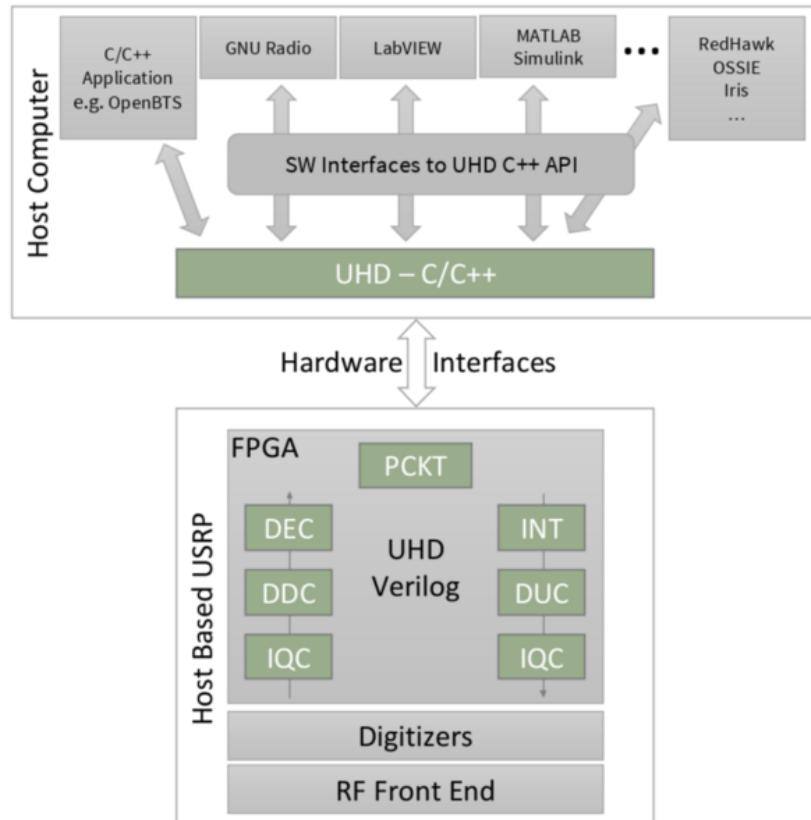


Figura 1. Componentes UHD. Figura obtenida de [ETT-UHD]

Debe instalarse previamente a la ejecución de los ejercicios del taller. Para el proceso de instalación se pueden seguir los diferentes tutoriales que hay disponibles:

Driver UHD	https://kb.ettus.com/Building_and_Installing_the_USRP_Open-Source_Toolchain_(UHD_and_GNU_Radio)_on_Linux
Conectividad X310	https://files.ettus.com/manual/page_usrp_x3x0.html#x3x0_setup_network https://files.ettus.com/manual/page_usrp_x3x0_config.html#x3x0cfg_hostpc_netcfg_ip

Tabla 2. Instalación del driver UHD y conectividad con las SDR X310

4.4 Configuración del Matlab: Communication Toolbox Support Package for USRP radio

Una vez instalado el controlador UHD, se debe instalar un paquete adicional disponible en Matlab Communication Toolbox Support Package for USRP radio [M-CUSRP]. La instalación es necesaria para poder interactuar con las SDRs a través de la interfaz de Matlab (ventana de comandos, etc). La instalación se puede llevar a cabo desde el menú de herramientas tal y como se indica en la Figura 2.

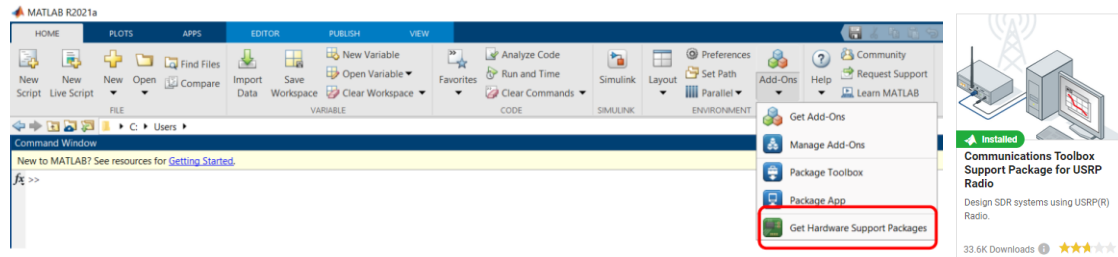


Figura 2. Integración de SDR con Matlab

Una vez seleccionado el paquete para la SDR [M-CUSRP] se seguirá un proceso guiado por Matlab donde se irán identificando las SDR instaladas en el sistema y comprobará su conectividad. A modo de ejemplo, en la Figura 3 se presentan algunas pantallas que aparecerán durante el proceso guiado de Matlab.

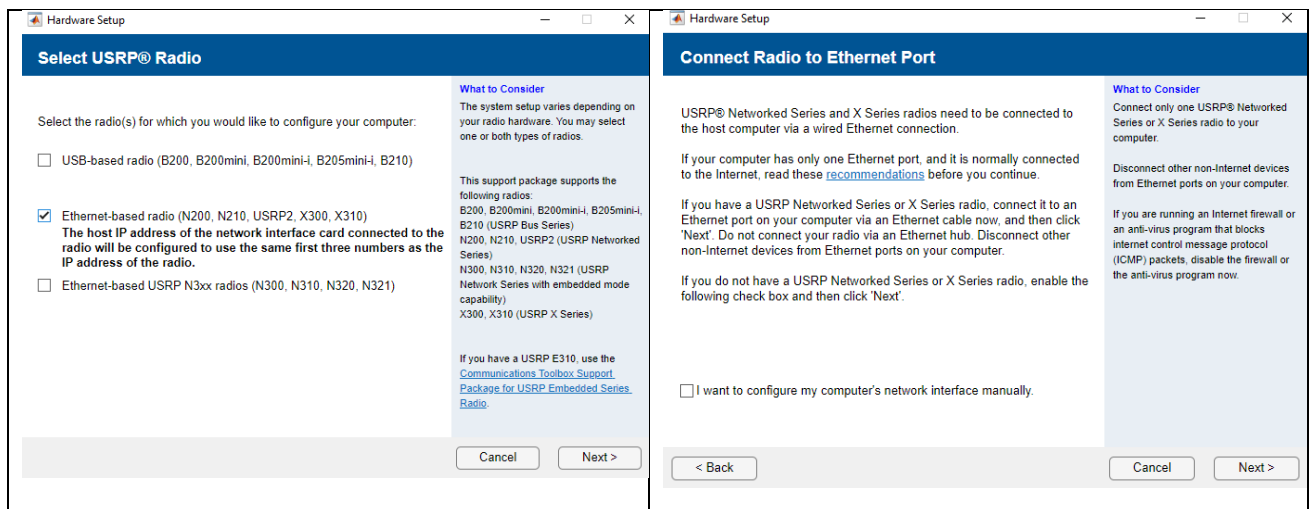


Figura 3. Ejemplo de pasos a seguir durante el proceso de instalación del Communication Toolbox Support Package for USRP radio

4.5 Configuración de los dispositivos

El entorno de experimentación se ha configurado con un mínimo de 4 SDRs, pero el taller puede extenderse a dispositivos adicionales. La Figura 4 muestra la configuración preparada para la realización del taller en las instalaciones del CTTC, dentro de uno de los laboratorios.

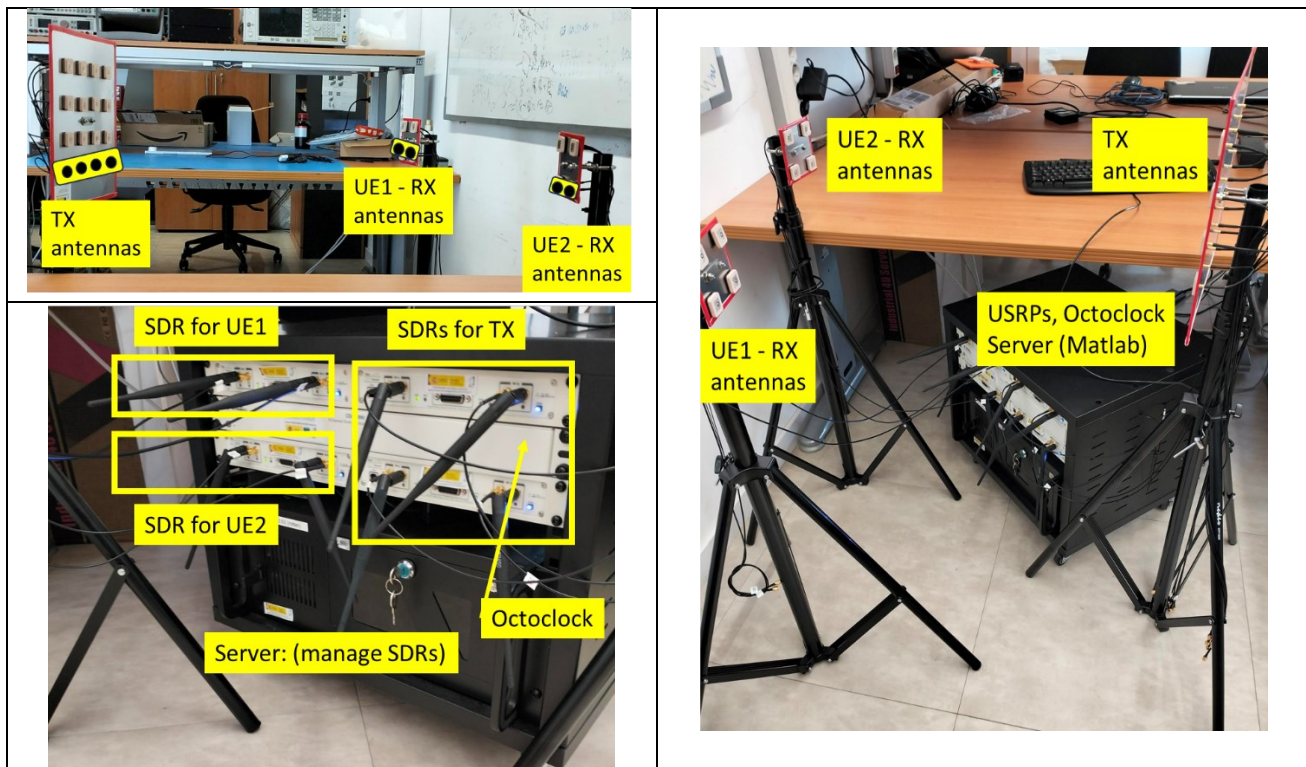


Figura 4. Entorno de laboratorio en el CTTC

La infraestructura se compone de 4 partes:

- Servidor:** Servidor Intel(R) Core (TM) i7-7800X CPU @ 3.50GHz con sistema operativo Ubuntu 22.04.2 LTS donde se ha instalado todo el software necesario: Matlab y driver UHD para comunicarse con las SDR. La conexión de datos entre servidor y SDRs es a través de conexiones SFP+, las cuales requieren del cableado y tarjeta de red correspondiente. Para la instalación se han seguido las instrucciones detalladas en [ETT-X310].
- SDR:** Como se aprecia en la figura, hay 4 SDRs (USRP X310) que tienen 2 tarjetas de RF cada una de ellas. Cada SDR está conectada directamente con el servidor y conexiones con el elemento que distribuye el sincronismo (que no se observan en la Figura 4).
- Elemento de sincronismo.** Este dispositivo es el Octoclock 2990, elemento que separa las SDRs superiores e inferiores en la Figura 4. Para la instalación se ha seguido los pasos descritos en [ETT-Octoclock]. Por la parte trasera del dispositivo se distribuye la señal de reloj de 10MHz y 1 PPS a las 4 SDRs. En el caso del laboratorio del CTTC, se utiliza una SDR como referencia para distribuir el reloj y la señal de PPS al resto de SDRs. En la Figura 5 se presenta el diagrama de conexión en el caso de 2 SDRs. La SDR denominada como SDR#1 es la fuente del reloj, y debemos estar seguros de que esté funcionando en los momentos de las pruebas que se necesiten que las SDR estén sincronizadas. La extensión a más SDRs se realiza conectando las salidas de "10 MHz out" y "PPS out" a la SDR correspondiente.
- Antenas:** Para cada tarjeta de RF hay una salida de transmisión (TX) y otra de recepción (RX), que se conectarán a diferentes paneles de antenas. Las antenas de cada panel tienen dimensiones 25*25*5.3mm y son antenas de parche (WPC.25B.35 2.4GHz Ceramic Patch Antenna) con un on-board MMCX(F) (Micro-miniature coaxial connector). Como se observa en la figura, los transmisores (SDR#1, SDR#2) están conectados en el panel de forma que

utilizan 4 antenas, mientras que los usuarios están (#SDR3 y #SDR4) están conectados a 2 antenas de cada panel.

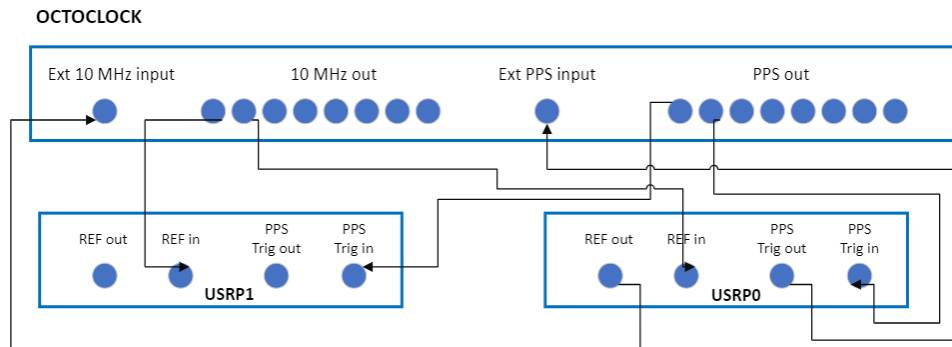


Figura 5. Interconexión del SDR y Octoclock para compartir el sincronismo

La configuración propuesta responde a evaluar un modelo de comunicación multiusuario como el que se ilustra en la Figura 6. Este corresponde a un sistema de comunicación multipunto a multipunto over-the-air", a través de la interfaz radio (medio físico). Es decir la señal se trasmite y recibe "over-the-air", pero el procesado digital se realiza en los servidores. El diseño del filtro transmisor necesita de un conocimiento del estado de canal, para ello, los diferentes terminales estimarán el canal en recepción, y lo reportarán al transmisor. Esta parte de la comunicación no es realizada "over-the-air", a través de intercambios de ficheros.

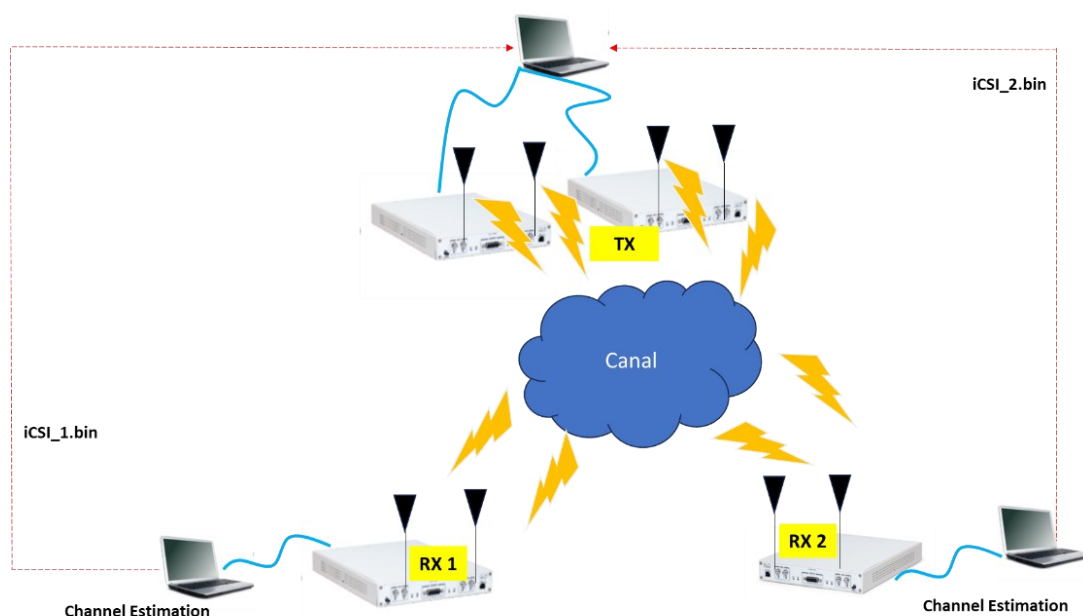


Figura 6. Modelo de comunicaciones

4.6 Puesta en marcha entorno de experimentación

Una vez se ha instalado el driver UHD, se ha configurado la conectividad de las USPs X310 e instalado el paquete de Matlab Communication Toolbox Support Package for USRP radio, ya se puede realizar la puesta en marcha del entorno de experimentación. Cada SDR estará controlada/gestionada por un Matlab, de forma que puedan ejecutar operaciones de forma independiente. Podemos comprobar que todo está bien configurado realizando los siguientes pasos:

- Abrir 3 ventanas de Terminal en Ubuntu para poder ejecutar una sesión de Matlab en cada una de ellas.
- Comprobar que todas las SDR configuradas estén disponibles, a través de la función de Matlab `findsdr`. En la Figura 7 se puede observar que la variable `current_USRPs` (`=findsdr`) contiene una estructura de datos. Ejecutando la instrucción, `current_USRPs(:).IPAddress` podremos comprobar las direcciones IP de cada SDR que se han encontrado, y que estarán accesibles desde Matlab. Además, con `current_USRPs(:).Status` veremos el estado de cada una de ellas. Si la SDR se ha encontrado correctamente (identificada), pero no está disponible (está ocupada) no se podrá utilizar en el experimento.

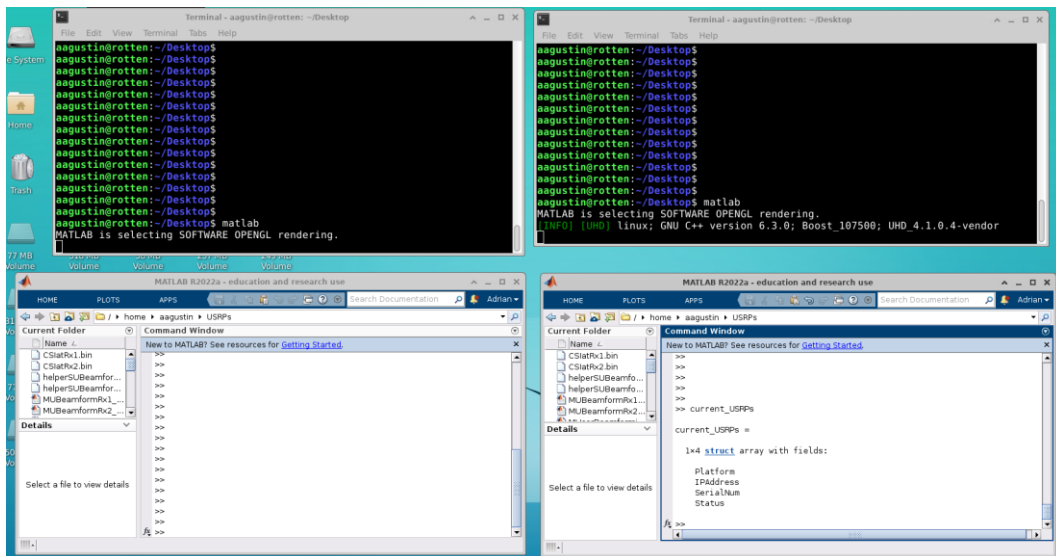


Figura 7. Puesta en marcha de las sesiones de Matlab en el servidor. Comprobación de que las SDRs están disponibles.

4.7 Código Matlab

El código se puede descargar desde: <https://github.com/TelecoRenta>

El paquete de software del taller contiene los siguientes ficheros en código Matlab:

Scripts:

1. Experimento 0

- Ex0_script_BS.m

- b. Ex0_script_UE_1.m / Ex0_script_Offline_UE_1.m
- c. Ex0_script_UE_2.m / Ex0_script_Offline_UE_1.m

2. Experimento 1

- a. Ex1_script_BS.m
- b. Ex1_script_UE_1.m / Ex1_script_Offline_UE_1.m
- c. Ex1_script_UE_2.m / Ex1_script_Offline_UE_2.m

3. Experimento 2

- a. Ex2_script_BS.m
- b. Ex2_script_UE_1.m / Ex2_script_Offline_UE_1.m
- c. Ex2_script_UE_2.m / Ex2_script_Offline_UE_2.m

4. Experimento 3

- a. Ex3_script_BS.m
- b. Ex3_script_UE_1.m / Ex3_script_Offline_UE_1.m
- c. Ex3_script_UE_2.m / Ex3_script_Offline_UE_2.m

Funciones:

1. MU_MIMO_Tx_BS_USRP.m
2. PilotSequenceGen.m
3. MU_MIMO_Rx_UE_n_USRP.m
4. Ex1_MU_MIMO_Rx_UE_n_USRP.m
5. ChannelEstimationMIMO.m
6. Offline_MU_MIMO_Rx_UE_n_USRP.m
7. Ex1_Offline_MU_MIMO_Rx_UE_n_USRP.m

4.8 Descripción de las funciones básicas de Matlab en las que se sustenta la práctica

A continuación, se detallan una serie de funciones disponibles en Matlab que se utilizarán para interactuar con las SDR a través de Matlab, y para la generación o almacenamiento de señales.

- **findsdru.** Esta función devuelve una estructura de datos que permite buscar el estado de todas las SDR encontradas y conectadas en la red del servidor donde se ejecuta Matlab. Si se incluye la dirección IP específica de una SDR, la búsqueda se centra en revisar si esa SDR está disponible ('Success'), No compatible con el firmware ('No compatible'), no responde ('No responding'), ocupada por otro proceso ('busy') o error desconocido ('Unknown error'). Más detalles a través de la ayuda de Matlab o en [M-FSDR]
- **comm.SDRuTransmitter.** Este objeto es que permite enviar los datos generados en Matlab a la SDR [M-SDRuTx]. Este objeto puede soportar la conexión con múltiples SDR de forma simultánea, y las entradas pueden de ser de forma matricial, por ejemplo, si solo hay un canal

de la SDR la entrada será un vector columna, y si hay múltiples, la entrada será una matriz con tantas columnas como canales activos en diferentes SDRs.

- **dsp.SpectrumAnalyzer.** Este objeto de Matlab permite representar señales en el dominio de la frecuencia y su objetivo poder observar el espectro en la banda que se esté trabajando [M-SPECA1]. Para más detalles de su configuración revisar [M-SPECA2].
- **comm.ConstellationDiagram.** Este objeto de Matlab permite representar la constelación de la señal [M-CONST].
- **step(obj,x).** Esta función es la que ejecuta el algoritmo implementando en el objeto de sistema `obj` con la entrada proporcionada con `x`. [M-STEP]
- **release(obj).** Esta instrucción permite liberar los recursos de sistema asignados al objeto (memoria, conexiones hardware, enlaces a ficheros). Generalmente, una vez liberado es cuando se permiten cambiar las propiedades del objeto.
- **comm.BasebandFileWriter.** Este objeto de sistema permite almacenar en un fichero binario la señal en banda base, donde se debe indicar el muestreo y la frecuencia central [M-FileWriter]. El objeto `comm.BasebandFileReader` permitirá la lectura del fichero [M-FileReader]. Estos objetos se utilizarán para poder almacenar señal y post-procesarla cuando en los casos que las operaciones en tiempo real no sean posibles.

4.9 Entendiendo el código del transmisor

La ejecución de la parte de transmisión se realiza a través del script `script_BS.m`, que a su vez llama a la función `MU_MIMO_Tx_BS_USRP.m`. En la *Figura 8* se realiza una breve descripción de las principales tareas de cada función. En el `script_BS.m` se configuran los principales parámetros de la comunicación como el número de antenas, número de usuarios, direcciones IP de las USRPs a utilizar para la transmisión, la frecuencia portadora y el tipo de precodificación espacial que se aplica en los recursos compartidos. Esta última puede ser de 3 tipos: 0) Siempre Matched filter (MF) o filtro adaptado, 1) Siempre Zero Forcing (ZF) o forzador de ceros, 2) Alternando un intervalo de tiempo MF y ZF.

Por su parte, la función `MU_MIMO_Tx_BS_USRP.m` tiene como misión inicializar las USRPs a los valores apropiados, y generar la señal a transmitir. Para ello, esta función irá leyendo la información de canal a través de unos ficheros que envía cada usuario y utilizará esa información para calcular el precodificador espacial.

script_BS.m

Configuration:

- M (tx antenas)
- K (UEs)
- N (rx antenas)
- IP addresses
- Carrier Frequency
- Type precoding

MU_MIMO_Tx_BS_USRP.m

- USRP configuration
- Generation Pilots in single-carrier
- Payload in OFDM
- Files to exchange the CSI (reading only)
- File to store all the received CSI
- Read the CSI
- Calculate the precoders
- Generate the data to be transmitted
- Send the data to the USRP

Figura 8. Descripción general de las principales funciones para realizar la transmisión

4.10 Entendiendo el código del receptor

Cada usuario debe ejecutar su script correspondiente, `script_UE_1.m` o `script_UE_2.m`, que se encarga de detallar la configuración de cada dispositivo. Hay que destacar que la USRP asociada a uno de los usuarios (usuario #1) es la encargada de proporcionar la señal de reloj para sincronizar a las restantes USRPs. Esta función se configura con la variable `isClockReference=1`.

Una vez se configuran los principales parámetros, cada script llama a la función `MU_MIMO_Rx_UE_n_USRP.m` que llevará a cabo la recepción y el procesamiento de la señal. Después de configurar el objeto de Matlab `comm.SDRUReceiver` que comunicará el espacio de trabajo de Matlab con la USRP se realizarán los diferentes pasos:

- Obtener la señal recibida por la USRP
- Estimar el canal a través de las secuencias piloto presentes en el preámbulo y obtención de las muestras que hacen referencia a los datos (payload).
- El canal estimado se escribe en un fichero (que será leído por el transmisor cuando necesite realizar la precodificación)
- Para estimar los datos se realiza la recepción de la señal OFDM y se seleccionan las portadoras y símbolos donde cada UE tiene su información
- Se explota la transmisión de secuencias conocidas (*midamble*) para tener una mejor estimación del canal en la recepción, y se obtiene la constelación recibida.

script_UE_1.m

Configuration:
 - N (rx antennas)
 - IP address
 - Carrier Frequency
 - IsClockReference (1)

script_UE_2.m

Configuration:
 - N (rx antennas)
 - IP address
 - Carrier Frequency
 - IsClockReference (0)

MU_MIMO_Rx_UE_n_USRP.m

- USRP configuration
- Initialize Matlab objects to visualize results
- Generation Pilots in single-carrier
- Receive data from the USRP
- Channel Estimation at the i-th frame
- Write the estimated channel in a file (to be exchanged with the tx)
- Extract Payload in OFDM
- Use midambles to estimate the channel on subcarriers
- Show the constellation of received symbols

Figura 9. Descripción general de las principales funciones para realizar la recepción de la señal en los usuarios

3. Modelo de señal

Esta sección describirá de forma resumida los diferentes aspectos a considerar para la transmisión y recepción de las señales.

4.1 Trama transmitida

La Figura 10 presenta la trama que se transmite que tiene una duración aproximada de unos 83.35 ms. Está compuesta por 2 bloques principales: Preámbulo y datos precodificados. Entre transmisiones de diferentes bloques hay un intervalo de tiempo sin ninguna transmisión que permitirá absorber la dispersión del canal.

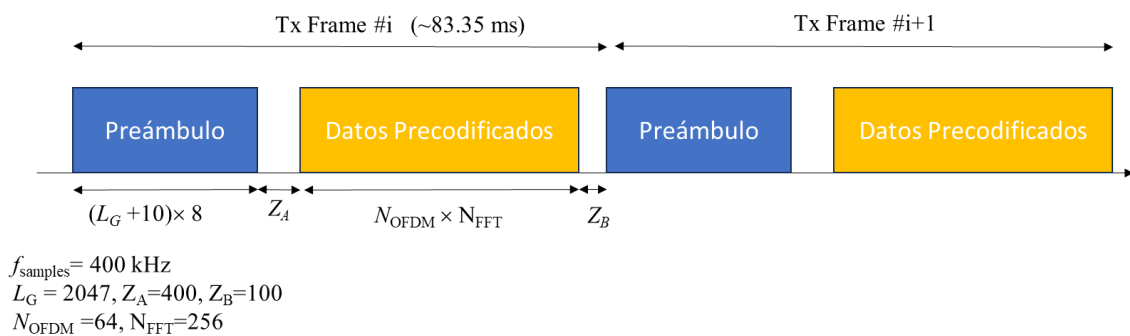


Figura 10. Trama de señal transmitida.

El Preámbulo consiste en transmitir una secuencia pseudoaleatoria independiente por cada antena sin ninguna precodificación. Se puede modelar como una matriz de dimensiones $(M \times (L_G + 10)8)$, con M el número de antenas transmisoras y L_G la longitud de la secuencia piloto, que en nuestro caso viene de las secuencias de Gold de longitud $L_G = 2047$. Para la transmisión de estas secuencias se considera la utilización de un filtro raíz de coseno alzado con 8 muestras por símbolo. La recepción de este preámbulo permitirá tener una estimación del canal inalámbrico en ese momento para toda la banda.

Los datos precodificados para cada usuario se van a transmitir como 64 símbolos OFDM (Orthogonal Frequency Division Multiplexing) [BAR-05] y la asignación de recursos se ilustra en la Figura 11, donde existen una serie de recursos compartidos por ambos usuarios (portadoras 4-12) que demandará de una precodificación espacial para eliminar las interferencias, y otros recursos ortogonales para cada usuario (portadoras 49-100 y 149-200). En los recursos asignados se transmitirán datos (recursos en naranja en la Figura 11) y secuencias piloto (o *midambles*) para refinar la estimación del canal (recursos en verde en la Figura 11). Los datos utilizan diferentes modulaciones por portadora (desde 64 QAM hasta 4-QAM), mientras que los *midambles* se transmiten con BPSK. Para este experimento no se ha considerado la utilización de prefijo cíclico (Cyclic Prefix, CP).

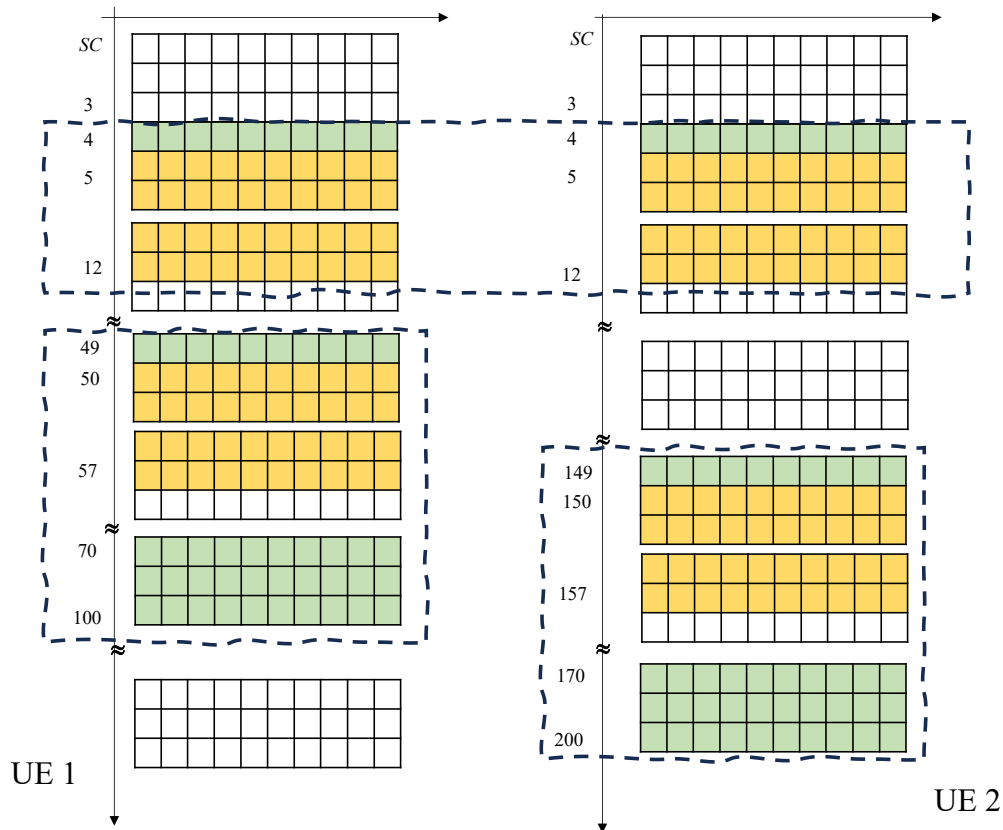


Figura 11. Asignación de los recursos a los diferentes usuarios en términos de portadoras y tiempo. La transmisión utiliza recursos compartidos (portadoras 4-12) y recursos ortogonales (portadoras 49-100 para UE1 y 149-200 para el UE2). Los recursos en “naranja” son destinados para la transmisión de datos, mientras que los recursos en “verde” transmiten secuencias conocidas (midambles) por parte del receptor, que ayudarán a mejorar la estimación del canal en recepción. En los recursos en blanco no se transmite ninguna señal.

4.2 Sincronización en tiempo (inicio de la trama)

Los usuarios necesitan detectar cuando empieza la trama transmisora, esto se puede realizar con la detección de las secuencias piloto presentes en el preámbulo. Al estar formadas por secuencias de Gold de gran longitud, correlando en recepción por las mismas señales obtendremos picos en los instantes que empiezan las secuencias, y por tanto la trama. Con esta información, podremos extraer las muestras que nos permitirán: i) estimar el canal y ii) detectar los datos.

4.3 Estimación de canal

La estimación de canal también se realiza a través de la decodificación del preámbulo presentado en la Figura 10. La señal recibida por el usuario i equipado con N antenas receptoras puede modelarse como,

$$\mathbf{Y} = \mathbf{H}\mathbf{P} + \mathbf{V} \quad (1)$$

con \mathbf{P} con dimensiones (M, L_p) una matriz con M el número de antenas transmisoras y L_p el tamaño de la secuencia piloto transmitida, que de acuerdo Figura 10 sería $L_p=(L_G+10)8$, \mathbf{H} es una matriz de dimensiones (N, M) y representa la matriz de canal, \mathbf{Y} modela la señal recibida y es una matriz de dimensiones (N, L_p) y finalmente la matriz \mathbf{V} representa el ruido presente en la comunicación. Las diferentes secuencias piloto son conocidas por los usuarios, por lo que la estimación puede realizarse

$$\hat{\mathbf{H}} = \mathbf{Y} \text{pinv}(\mathbf{P}) \quad (2)$$

donde $\text{pinv}()$ representa la función pseudo-inversa.

4.4 OFDM

La transmisión de los datos se realiza a través de OFDM. Seguidamente se presentan las principales características de este tipo de transmisión. Asumiendo una comunicación punto a punto (una antena en transmisión y una antena en recepción) la señal se puede modelar como,

$$\tilde{\mathbf{y}}(n) = \tilde{\mathbf{H}} \mathbf{G} \mathbf{x}(n) + \mathbf{v}(n) \quad (3)$$

donde $\tilde{\mathbf{y}}$, \mathbf{v} son vectores de dimensión $(N_c, 1)$, $\tilde{\mathbf{H}}$, \mathbf{G} son matrices de dimensiones (N_c, N_c) , \mathbf{x} es el vector de símbolos a transmitir de dimensión $(N_c, 1)$. La matriz de canal, $\tilde{\mathbf{H}}$ presenta una estructura Toeplitz y circulante [BAR05], que tiene la siguiente descomposición,

$$\tilde{\mathbf{H}} = \begin{bmatrix} h_0 & 0 & \cdots & 0 & h_L & h_{L-1} & \cdots & \cdots & h_1 \\ \vdots & h_0 & 0 & \ddots & \ddots & h_L & \cdots & \cdots & \vdots \\ h_L & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & h_L \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & h_L & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & h_L & h_{L-1} & \cdots & \cdots & \cdots & h_0 \end{bmatrix} = \mathbf{W} \mathbf{\Lambda} \mathbf{W}^H \quad (4)$$

donde la matriz \mathbf{W} es unitaria y puede obtenerse de la Transformada Discreta de Fourier, y $\mathbf{\Lambda}$ es una matriz diagonal de dimensiones (N_c, N_c) , y los elementos de la diagonal principal dados por,

$$\Lambda(i, i) = \sum_{l=0}^L h(l) e^{-j \frac{2\pi i l}{N_c}} \quad (5)$$

Utilizando esta propiedad, y fijando $\mathbf{G} = \mathbf{W}$, después de aplicar un procesamiento a la señal recibida (con \mathbf{W}^H), esta se puede escribir como:

$$\mathbf{z}(n) = \mathbf{W}^H \tilde{\mathbf{y}}(n) = \mathbf{\Lambda} \mathbf{x}(n) + \mathbf{W}^H \mathbf{v}(n) \quad (6)$$

En particular, las operaciones $\mathbf{W} \mathbf{x}$, $\mathbf{W}^H \tilde{\mathbf{y}}$ se corresponden a la transformada discreta inversa de Fourier (IDFT en inglés) de la señal \mathbf{x} y la transformada Discreta de Fourier (DFT en inglés), respectivamente. Cada una de ellas puede realizarse de forma eficiente con la Fast Fourier Transform (FFT), con la longitud de la FFT de N_c .

Como se puede observar el canal equivalente visto por cada símbolo, en general es diferente, excepto en el caso de que el canal pueda asumirse que es plano en frecuencia y en ese caso la matriz es proporcional a la matriz identidad, $\mathbf{\Lambda} \propto \mathbf{I}$.

4.5 Precodificación MIMO OFDM

La extensión a la configuración multiantena para OFDM consiste en considerar la precodificación espacial por portadora, es decir, para la portadora c la señal recibida puede escribirse como,

$$\begin{bmatrix} z_1(c, t) \\ \vdots \\ z_N(c, t) \end{bmatrix} = \begin{bmatrix} \Lambda_{1,1}(c, c) & \cdots & \Lambda_{1,M}(c, c) \\ \vdots & \ddots & \vdots \\ \Lambda_{N,1}(c, c) & \cdots & \Lambda_{N,M}(c, c) \end{bmatrix} \mathbf{F} \begin{bmatrix} x_1(c, t) \\ \vdots \\ x_{N_{streams}}(c, t) \end{bmatrix} + \begin{bmatrix} \tilde{v}_1(c, t) \\ \vdots \\ \tilde{v}_N(c, t) \end{bmatrix} \quad (7)$$

donde $\Lambda_{n,m}(c, c)$ es el coeficiente de canal en el enlace entre la antena receptor n y la antena transmisora m , en la portadora c , \mathbf{F} es la matriz de precodificación de dimensiones $(M, N_{streams})$, y \tilde{v}_n representa la componente de ruido en la antena receptora n . Como puede apreciarse en (7), la matriz \mathbf{F} debería diseñarse para cada portadora c . En el caso de que la respuesta del canal sea plana en la banda frecuencial, los coeficientes de canal son los mismos en toda la banda.

4.6 Precodificación MIMO en este Taller

Aunque consideraremos que los datos precodificados se transmiten en OFDM, en este taller asumirá que canal se podría considerar plano en la banda de interés. Por esta razón, se considerará la estimación de canal basada en la recepción del preámbulo para realizar la precodificación espacial de los diferentes streams o modos de transmisión. Como se ha comprobado en (7), esta opción no es la óptima pues en canal puede variar a través de las portadoras. La precodificación considerada aquí, realizará una separación espacial de los terminales, sin embargo, no realizará una ecualización del canal y existirá un error que se propagará a través de las portadoras y símbolos OFDM. Este taller, analizará el impacto del error obtenido con esta aproximación.

La señal transmitida con la información se puede describir como una matriz de dimensiones $(M, N_{OFDM}N_{FFT})$,

$$\begin{aligned} \mathbf{X} &= \mathbf{F}_{00}\mathbf{s}_{00} + \mathbf{F}_{01}\mathbf{s}_{01} + \mathbf{F}_1\mathbf{s}_1 + \mathbf{F}_2\mathbf{s}_2 \\ \mathbf{s}_a &= f(\mathbf{W}[\mathbf{x}_{a,1} \cdots \mathbf{x}_{a,N_{OFDM}}]) \end{aligned} \quad (8)$$

donde los símbolos \mathbf{s} de dimensión $(1, N_{OFDM}N_{FFT})$ hacen referencia a la señal temporal OFDM a través de la función $f()$ que realiza la transformación de matriz a vector de todos los símbolos a los que se le aplica la FFT. La señal $\mathbf{s}_{00}, \mathbf{s}_{01}$ solo tienen información en las portadoras compartidas (4-12 y ceros en las restantes) con mensajes superpuestos para los 2 usuarios, mientras $\mathbf{s}_1, \mathbf{s}_2$ hacen referencia a las señales temporales OFDM que contienen información solo para el usuario 1 (49-57, 70-100 y ceros en las restantes) y usuario 2 (149-157, 170-200, y ceros en las restantes), respectivamente. Finalmente, $\mathbf{F}_{00}, \mathbf{F}_{01}, \mathbf{F}_1, \mathbf{F}_2$ son los filtros espaciales que se aplican de dimensión $(M, 1)$. La señal recibida por los usuarios se puede escribir como,

$$\begin{aligned} \mathbf{Y}_1 &= \mathbf{H}_1\mathbf{X} + \mathbf{W}_1 = \mathbf{H}_1\mathbf{F}_{00}\mathbf{s}_{00} + \mathbf{H}_1\mathbf{F}_{01}\mathbf{s}_{01} + \mathbf{H}_1\mathbf{F}_1\mathbf{s}_1 + \mathbf{H}_1\mathbf{F}_2\mathbf{s}_2 + \mathbf{W}_1 \\ \mathbf{Y}_2 &= \mathbf{H}_2\mathbf{X} + \mathbf{W}_2 = \mathbf{H}_2\mathbf{F}_{00}\mathbf{s}_{00} + \mathbf{H}_2\mathbf{F}_{01}\mathbf{s}_{01} + \mathbf{H}_2\mathbf{F}_1\mathbf{s}_1 + \mathbf{H}_2\mathbf{F}_2\mathbf{s}_2 + \mathbf{W}_2 \end{aligned} \quad (9)$$

donde $\mathbf{H}_1, \mathbf{H}_2$ son matrices de dimensión (N, M) , y $\mathbf{Y}_1, \mathbf{Y}_2$ son matrices de dimensión $(N, N_{OFDM}N_{FFT})$. Considerando que la recepción de las secuencias pilotos tenemos una estimación de canal dada por (2), $\hat{\mathbf{H}}_1, \hat{\mathbf{H}}_2$. Para la señal que va dirigida a los diferentes usuarios en los recursos ortogonales utilizaremos el filtro adaptado,

$$\mathbf{F}_1 = \frac{1}{\sqrt{\beta}} \frac{1}{\sqrt{\text{tr}(\hat{\mathbf{H}}_1^H \hat{\mathbf{H}}_1)}} \hat{\mathbf{H}}_1^H, \quad \mathbf{F}_2 = \frac{1}{\sqrt{\beta}} \frac{1}{\sqrt{\text{tr}(\hat{\mathbf{H}}_2^H \hat{\mathbf{H}}_2)}} \hat{\mathbf{H}}_2^H \quad (10)$$

donde la variable β se utilizará para asegurar que la potencia de la señal transmitida es constante. En recepción una vez después de aplicar la recepción OFDM, la señal recibida en los recursos ortogonales seleccionados a través de \mathcal{R} se puede escribir el símbolo n OFDM como

$$\begin{aligned} f_1(\mathbf{W}^H \tilde{\mathbf{y}}_1) &\xrightarrow{\text{yields}} \mathbf{z}_1(n) = \mathbf{\Lambda}_1 \mathbf{x}_1(n) + \tilde{\mathbf{v}}_1(n) \\ f_2(\mathbf{W}^H \tilde{\mathbf{y}}_2) &\xrightarrow{\text{yields}} \mathbf{z}_2(n) = \mathbf{\Lambda}_2 \mathbf{x}_2(n) + \tilde{\mathbf{v}}_2(n) \end{aligned} \quad (10)$$

donde las dimensiones de los vectores y matrices dependen del número de portadoras que se ha asignado para cada transmisión. Si el canal fuera plano en frecuencia y la estimación de canal es ideal entonces $\mathbf{\Lambda}_1 = \mathbf{\Lambda}_2 = \mathbf{I}_{N_{FFT}}$.

En los recursos compartidos donde se quiere separar las transmisiones utilizando la dimensión espacial, se utilizará el siguiente filtro espacial,

$$[\mathbf{F}_{00} \quad \mathbf{F}_{01}] = \begin{cases} [\mathbf{F}_1 \quad \mathbf{F}_2], & \text{Filtro adaptado} \\ \frac{1}{\sqrt{\beta}} \text{norm} - \text{pinv}([\hat{\mathbf{H}}_1^H \quad \hat{\mathbf{H}}_2^H]), & \text{Forzador de ceros} \end{cases} \quad (11)$$

La primera opción es el filtro adaptado, mientras que la segunda opción considera el forzador de ceros considerando la función pseudo-inversa normalizada. La señal recibida en las portadoras compartidas por los usuarios se puede expresar como,

$$\begin{aligned} f_{00,01}(\mathbf{W}^H \tilde{\mathbf{y}}_1) &\xrightarrow{\text{yields}} \mathbf{z}_{00} = \mathbf{\Lambda}_{11} \mathbf{x}_{00} + \mathbf{\Lambda}_{12} \mathbf{x}_{01} + \tilde{\mathbf{v}}_{00} \\ f_{00,01}(\mathbf{W}^H \tilde{\mathbf{y}}_2) &\xrightarrow{\text{yields}} \mathbf{z}_{01} = \mathbf{\Lambda}_{22} \mathbf{x}_{01} + \mathbf{\Lambda}_{21} \mathbf{x}_{00} + \tilde{\mathbf{v}}_{01} \end{aligned} \quad (12)$$

donde $\mathbf{\Lambda}_{ij}$ es el canal equivalente para la señal que contiene el mensaje j y se recibe en el usuario i . Si la separación espacial fuera correcta, se esperaría tener $\mathbf{\Lambda}_{12} = \mathbf{\Lambda}_{21} \approx \mathbf{0}$. Esto se conseguiría con el forzador de ceros si existe una estimación precisa del canal, y con el filtro adaptado si las firmas espaciales de cada usuario permiten su separación.

4. Ejercicios prácticos

Como se ha introducido anteriormente, la utilización de las SDRs facilita la experimentación de las comunicaciones inalámbricas y la validación de conceptos de diseño obtenidos por simulación teórica. No obstante, debido a la complejidad de los sistemas inalámbricos, comúnmente el análisis teórico de alguno de los bloques puede realizar consideraciones o simplificaciones del comportamiento de otros bloques.

Estos ejercicios prácticos, a través de un ejemplo sencillo, están enfocados para:

- Familiarizarse con las SDRs (USRPs) y Matlab
- Introducir algunos aspectos prácticos de las comunicaciones MIMO multi-usuario y OFDM
- Identificar los principales aspectos a tener en cuenta para la correcta recepción de la señal: tipo de filtro espacial, ecualización de canal, número de modos de transmisión que se pueden soportar

La práctica está organizada en la realización de una serie de ejercicios, ejecutados de acuerdo con la siguiente guía.

Guía de experimentos

4.1 Experimento 1: Utilización del precodificador Zero Forcing

Los ficheros por ejecutar serán:

- Experimentos en laboratorio
 - Ex1_script_BS.m, Ex1_script_UE1.m, y Ex1_script_UE2.m, en 3 diferentes entornos de Matlab. Durante la ejecución se podrá observar las constelaciones de los símbolos en diferentes portadoras.
- Procesado offline de ficheros previamente almacenados
 - Se puede ver los resultados para cada uno de los usuarios ejecutando Ex1_offline_script_UE1.m, y Ex1_offline_script_UE2.m

En el ejemplo de la *Figura 12-arriba-izquierda* se muestran las constelaciones en las portadoras 5 y 9 para el UE1, mientras *Figura 12-arriba-derecha* presenta las constelaciones recibidas en las mismas portadoras (5 y 9) pero para el UE2. Por otro lado, *Figura 12-abajo-izquierda* representa las constelaciones recibidas en las portadoras 50 y 55 para el UE1, y *Figura 12-abajo-derecha* está dedicada para la constelación de la señal recibida en las portadoras 150 y 155 del asignadas al UE2.

Pregunta 1.1: Aunque se observa que los puntos de la constelación se identifican correctamente, para los 4 casos se observa que las constelaciones tienen alguna rotación, en el caso de la *Figura 12* es más pronunciado para la señal recibida en los recursos asignados solo para los UE1 y UE2 (portadoras 50,55, 150, 155), Por qué?

Respuesta 1.1: Ese se va a intentar a averiguar a través de las siguientes preguntas. No obstante, la razón fundamental es que tal como se ha visto en la ecuación (10) y (12), el hecho de aplicar una precodificación espacial asumiendo que el canal es constante en toda la banda, no es lo óptimo, y tenemos un canal que debemos compensar si no queremos que nos afecte a la fase de los símbolos recibidos.

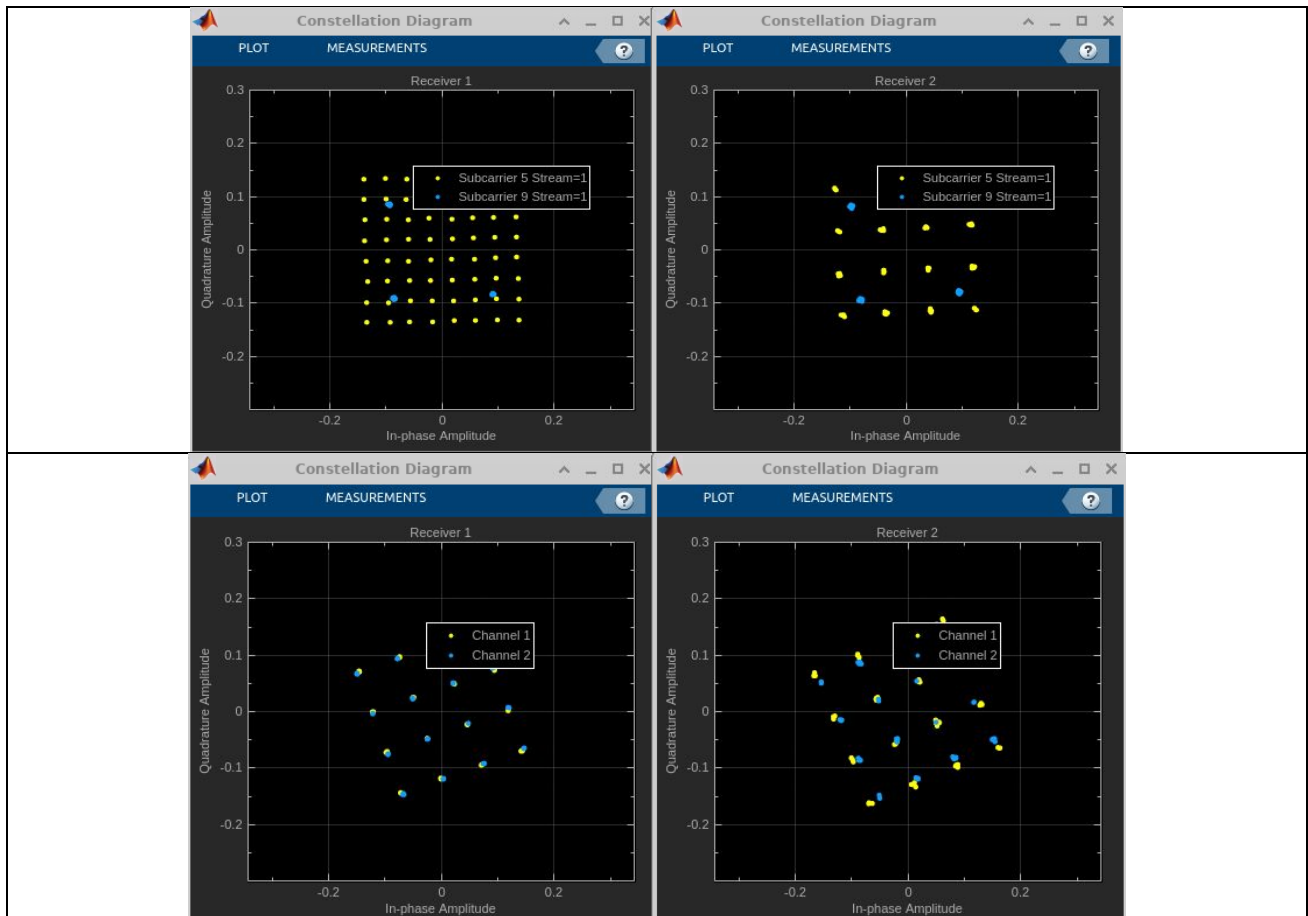


Figura 12. Ejemplo de constelaciones recibidas en el Experimento 1

Las causas que podrían ser responsables serían:

- **Error de Frecuencia transmisor-receptor:** La configuración de este experimento utiliza que todas las SDRs comparten las señales de sincronismo, por lo que el error de frecuencia que puede existir debería ser negligible.
- **Variación del canal en tiempo.** Existe posibilidad de revisar cómo evoluciona el canal reportado por cada UE en cada trama si en el script se especifica que el cada UE genere un fichero para guardar la señal, `Rx_input.save_rx_signal`. En tal caso, se puede utilizar el `script_Offline_UE_1.m` o `script_Offline_UE_2.m` para revisar los diferentes ficheros almacenados. En la Figura 13 se presenta el canal estimado en recepción durante las diferentes tramas recibidas. Como se puede observar, el canal no está variando de forma rápida, por lo que no debería influir.

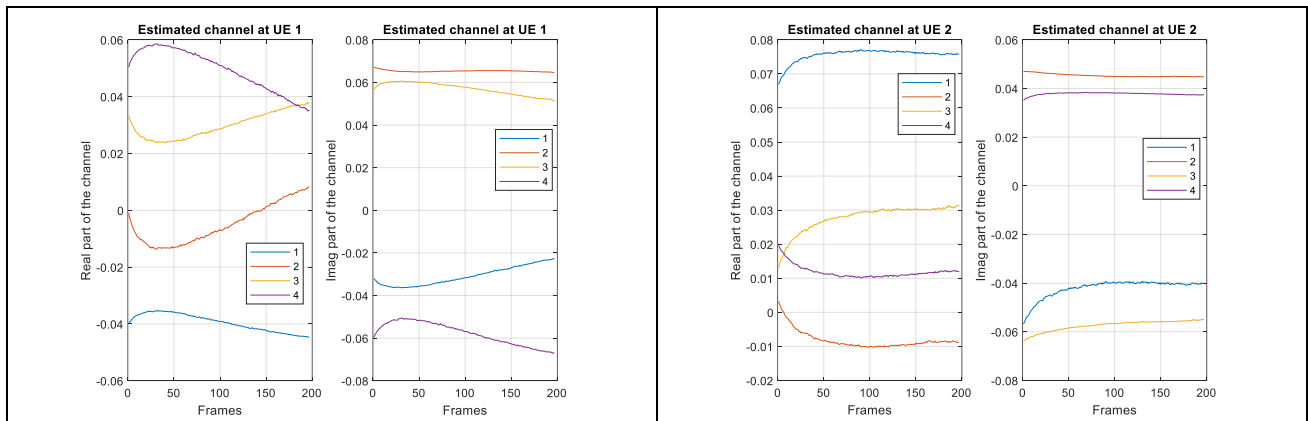


Figura 13. Canal estimado en los diferentes frames en el Experimento 1

- Variación del canal en frecuencia.** La estimación de canal se hace con el preámbulo que ocupa toda la banda y que es óptima cuando el canal no presenta selectividad en frecuencia. Sin embargo, el método implementado en este taller establece que la señal no queda ecualizada para la recepción de los datos, ver ecuación (10). ¿Cómo varía el canal entre las diferentes portadoras?

Para ver esta relación podemos explotar las secuencias *midamble* que se transmiten en las portadoras 70-100 para el UE1 y 170-200 para el UE2, ver Figura 11. La Figura 14 presenta el error de fase medido por el UE1 y UE2 en las portadoras que se envía las secuencias *midambles* en diferentes frames. Se puede observar que existe un error de fase que aproximadamente es lineal con el número de portadoras en este experimento. Lo que nos desvela esta figura es que hay un error de fase que hay que compensar en recepción para poder tener la constelación centrada.

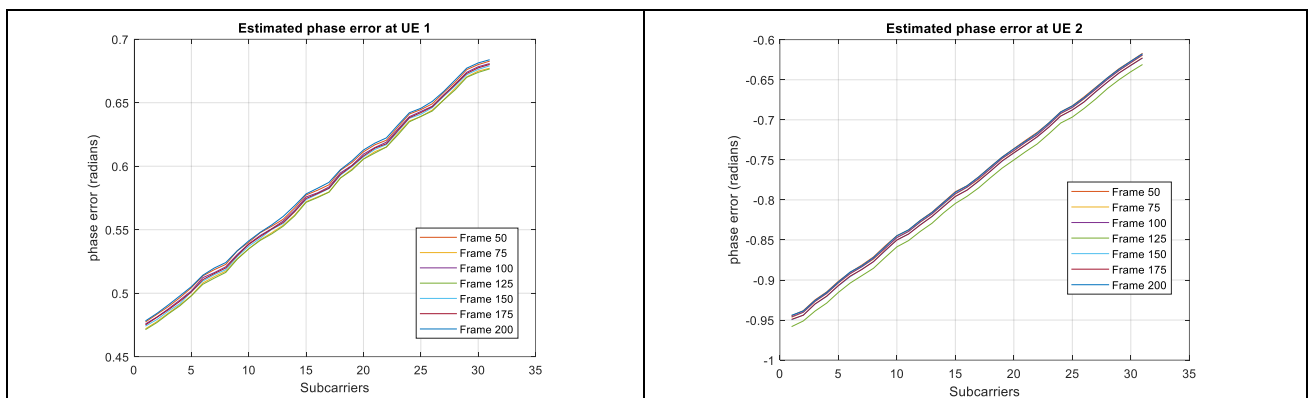


Figura 14. Error de fase medido para las portadoras 70-170 para el UE1 (izquierda) y 170-200 para el UE2 (derecha)

Pregunta 1.2: Si los símbolos de las constelaciones que se muestran en la Figura 12 se corresponden para las portadoras UE1: 5, 9, 50, 55 y para el UE2: 5, 9, 150, 155. ¿Cómo deberíamos modificar nuestro código de la función `Ex1_MU_MIMO_RX_UE_n_USRP.m` o `Ex1_offline_MU_MIMO_RX_UE_n_USRP.m`?

```
%% Lines 237 -246
%% midamble
%%
y = fftOut(4,:).';
%% Is there any phase offset to compensate??
%%
%%
%%
```

```
%%
%% Lines 268-275
%% midamble
%%
y = fftOut(SC_Orth_UE-1,:).';
%% Is there any phase offset to compensate??
%%
%%
%%
y = fftOut(SC_Orth_UE,:).';      % Extract Subcarrier 50/150
%% Is there any phase offset to compensate??
%%
%%
%%
%% Lines 289-294
y = fftOut(SC_Orth_UE+5,:).';      % Extract Subcarrier 50/150+5
%% Is there any phase offset to compensate??
%%
%%
%%
%% Lines 300-306
%% Offsets in the midamble vs carriers 70-100 or 170-200
y_matrix = fftOut(SC_Orth_UE+20:SC_Orth_UE+50,:); %% matrix
%% Is there any phase offset to compensate??
%%
%%
%%
%%
```

Respuesta 1.2: Aunque hay un error de fase que cambia con las portadoras, entre portadoras próximas este error es muy similar. Para ello deberíamos considerar los *midambles* que se transmiten en las portadoras 4, 49 para el UE1 y 4, 149 para el UE2. Y con ese error compensaremos la señal recibida en las portadoras que se muestran en la constelación (5, 9, 50, 55) y (5, 9, 150, 155). Además del método utilizado, se puede explorar otros métodos como por ejemplo el uso de un prefijo cíclico, que los símbolos que se transmiten son los mismos que en el inicio de la trama. Pero este no es el objetivo de este taller. La estimación se puede realizar calculando el error de fase medio durante los 64 símbolos OFDM que se transmiten con las secuencias *midamble* y que son conocidas por los usuarios.

```
%% Lines 237 -246
%% midamble
%%
y = fftOut(4,:).';
rx_phase_offset = mean(y.*seq_midamble. ');
rx_phase_offset = rx_phase_offset/sqrt(rx_phase_offset'*rx_phase_offset);
%%
%%
y = rx_phase_offset'*fftOut(5,:).';

%% Lines 268-275
%% midamble
%%
y = fftOut(SC_Orth_UE-1,:).';
rx_phase_offset = mean(y.*seq_midamble. ');
rx_phase_offset = rx_phase_offset/sqrt(rx_phase_offset'*rx_phase_offset);
```



```
y = rx_phase_offset'*fftOut(SC_Orth_UE,:).'; % Extract Subcarrier 50/150
y = rx_phase_offset'*fftOut(SC_Orth_UE+5,:).'; % Extract Subcarrier 50/150+5
```

4.2 Experimento 2: Utilización del precodificador Matched Filter

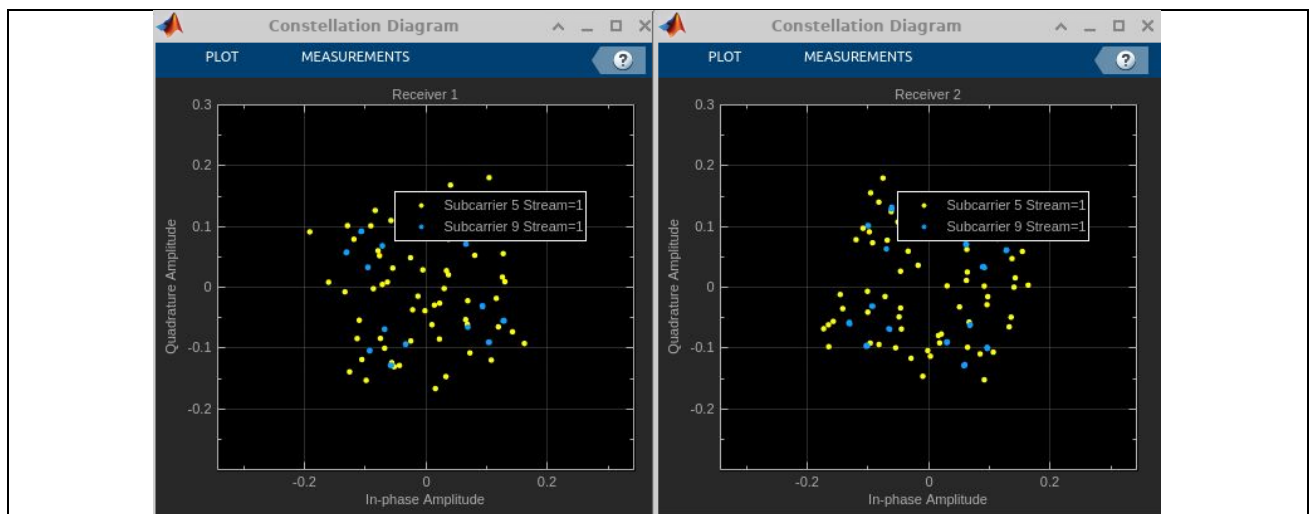
Los ficheros a ejecutar serán:

- Experimentos en laboratorio
 - Ex2_script_BS.m, Ex2_script_UE1.m, y Ex2_script_UE2.m, en 3 diferentes entornos de Matlab.
- Procesado offline de ficheros previamente almacenados
 - Se puede ver los resultados para cada uno de los usuarios ejecutando Ex2_offline_script_UE1.m, y Ex2_offline_script_UE2.m

Durante la ejecución se podrá observar las constelaciones de los símbolos en diferentes portadoras. En el ejemplo de la *Figura 15-arriba-izquierda* se muestran las constelaciones en las portadoras 5 y 9 para el UE1, mientras *Figura 15-arriba-derecha* presenta las constelaciones recibidas en las mismas portadoras (5 y 9) pero para el UE2. Por otro lado, *Figura 15-abajo-izquierda* representa las constelaciones recibidas en las portadoras 50 y 55 para el UE1, y *Figura 15-abajo-derecha* está dedicada para la constelación de la señal recibida en las portadoras 150 y 155 del asignadas al UE2. En este caso se ha aplicado la corrección de error de fase en recepción derivada en la sección anterior y se observa que la constelación de los símbolos en la portadoras ortogonales está corregida. Sin embargo:

Pregunta 2.1: ¿Por qué no se reciben correctamente los símbolos en las portadoras compartidas?

Respuesta 2.1: El filtro adaptado en un entorno MIMO permite enfocar la señal en una dirección. Sin embargo, la amplitud del haz por el que se transmite la señal depende del número de antenas y de la distancia de recepto al transmisor. Además, también se generan espurios de señal en otras direcciones, lo que da lugar a interferencia. En el caso particular de este taller, los receptores están muy próximos al transmisor y no es posible separarlos solo con la filtra espacial individual de cada uno de ellos.



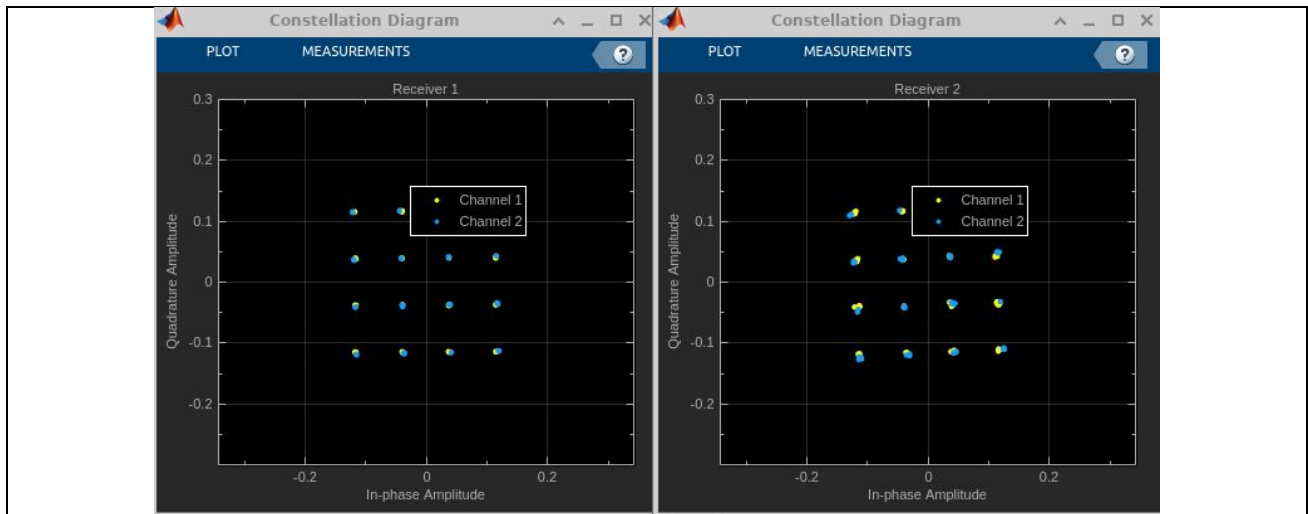


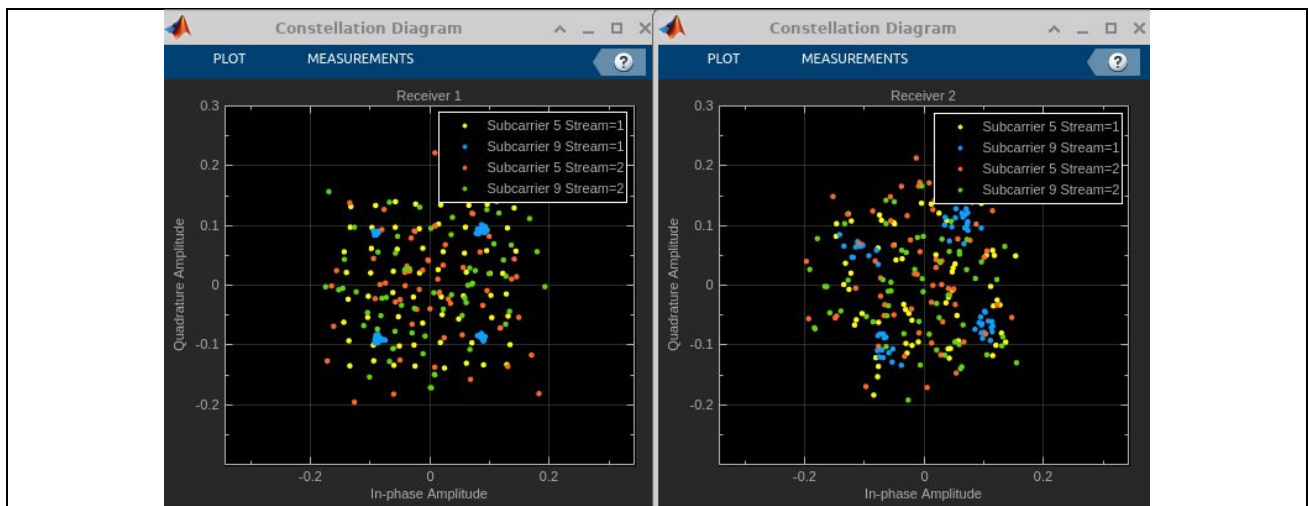
Figura 15. Ejemplo de constelaciones recibidas en el Experimento 2

4.3 Experimento 3: Utilización de múltiples antenas en recepción

Los ficheros a ejecutar serán:

- Experimentos en laboratorio
 - Ex3_script_BS.m, Ex3_script_UE1.m, y Ex3_script_UE2.m, en 3 diferentes entornos de Matlab.
- Procesado offline de ficheros previamente almacenados
 - Se puede ver los resultados para cada uno de los usuarios ejecutando Ex3_offline_script_UE1.m, y Ex3_offline_script_UE2.m

Durante la ejecución se podrá observar las constelaciones de los símbolos en diferentes portadoras. En el ejemplo de la *Figura 16*-arriba-izquierda se muestran las constelaciones en las portadoras 5 y 9 para el UE1, mientras *Figura 16*-arriba-derecha presenta las constelaciones recibidas en las mismas portadoras (5 y 9) pero para el UE2. Por otro lado, *Figura 16*abajo-izquierda representa las constelaciones recibidas en las portadoras 50 y 55 para el UE1, y *Figura 16*-abajo-derecha está dedicada para la constelación de la señal recibida en las portadoras 150 y 155 del asignadas al UE2. A diferencia de los casos anteriores, ahora cada UE está equipado con $N=2$ antenas, y se aprovechan para que el transmisor intente transmitir 2 modos a cada usuario.



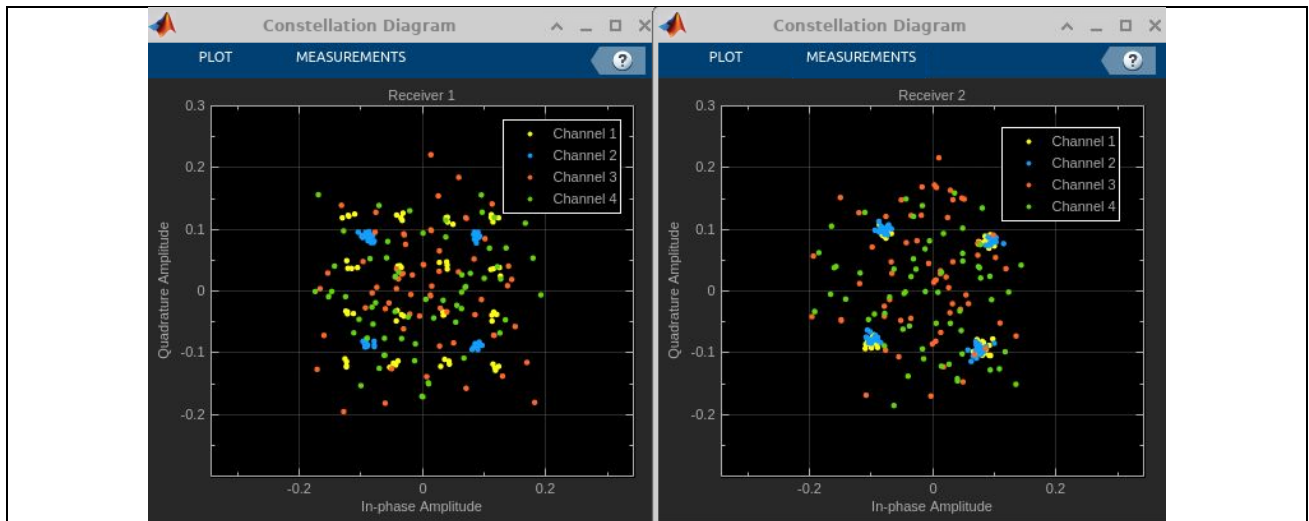


Figura 16. Ejemplo de constelaciones recibidas en el Experimento 3

A la vista de los resultados los usuarios no pueden recibir todos sus datos en los diferentes modos.

Pregunta 3.1: ¿Cuál puede ser el motivo si no existen errores en la estimación de canal o errores de frecuencia/fase? Pista: utilizar la variable: `Incremental_H_m` que almacena los coeficientes de canal.

Respuesta 3.2. El número de modos que se pueden transmitir en una enlace MIMO depende del número de antenas transmisoras, M , número de antenas receptoras, N , y del rango del canal, H . En general la comunicación se puede modelar como Line of Sight (LoS), y cada terminal puede explotar un solo modo, aunque tenga más antenas. Para ver el número de modos, se puede pintar los autovalores del canal

```
H = zeros(2,4); svd_m = zeros(2,200);
for kk=1:200
    H(1,:) = Incremental_H_m(1:4,kk).';
    H(2,:) = Incremental_H_m(5:8,kk).';

    svd_m(:,kk) = svd(H*H');
end
```

Al representar los autovalores de canal en la Figura 17, se observa que solo se podrá existir un modo por cada usuario, ya que el otro modo tiene un valor muy pequeño.

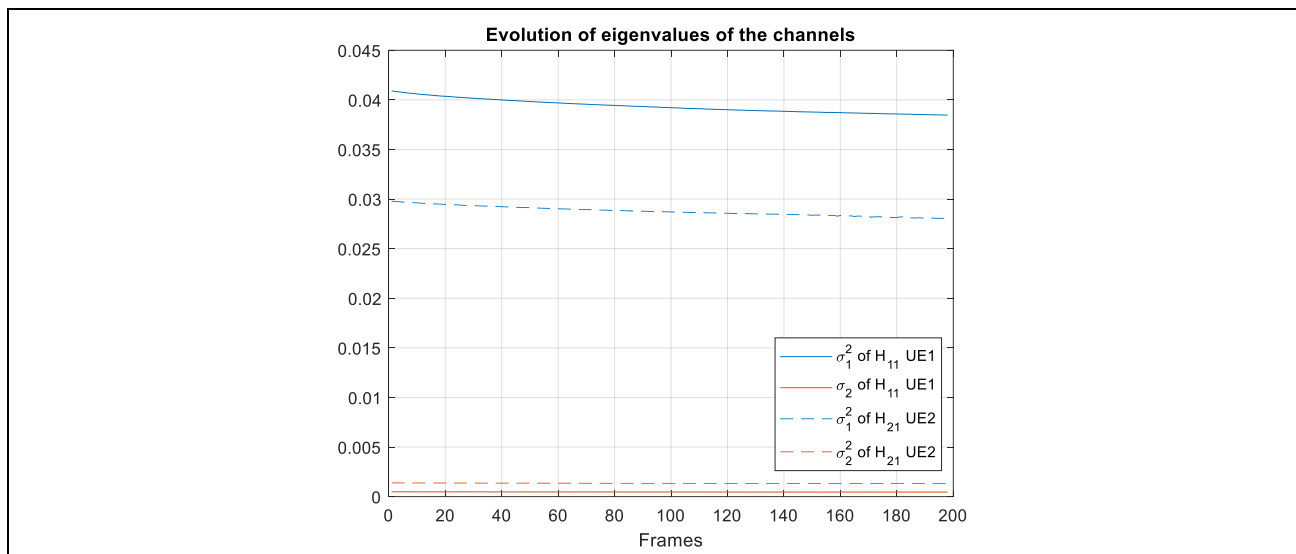


Figura 17. Autovalores de los canales del UE1 y UE2 en el Experimento 3

5. Referencias

- [M-CUSRP] <https://es.mathworks.com/help/supportpkg/usrpradio/index.html>
- [M-FSDR] <https://es.mathworks.com/help/supportpkg/usrpradio/ug/findsdru.html>
- [M-SPECA1] <https://es.mathworks.com/help/dsp/ref/spectrumanalyzer.html>
- [M-SPECA2] <https://es.mathworks.com/help/dsp/ug/configure-spectrum-analyzer.html>
- [M-CONST] <https://es.mathworks.com/help/comm/ref/comm.constellationdiagram-system-object.html>
- [M-Release] <https://es.mathworks.com/help/matlab/ref/matlab.system.releasesystemobject.html>
- [M-step] <https://es.mathworks.com/help/matlab/ref/step.html>
- [M-SDRuTx] <https://es.mathworks.com/help/supportpkg/usrpradio/ug/comm.sdrutransmitter-system-object.html>
- [M-FileReader] <https://es.mathworks.com/help/comm/ref/comm.basebandfilereader-system-object.html>
- [M-FileWriter] <https://es.mathworks.com/help/comm/ref/comm.basebandfilewriter-system-object.html>
- [ETT-UHD] <https://kb.ettus.com/UHD>
- [ETT-UHD2] [https://kb.ettus.com/Building_and_Installing_the_USRP_Open-Source_Toolchain_\(UHD_and_GNU_Radio\)_on_Linux](https://kb.ettus.com/Building_and_Installing_the_USRP_Open-Source_Toolchain_(UHD_and_GNU_Radio)_on_Linux)
- [ETT-X310] https://files.ettus.com/manual/page_usrp_x3x0.html
- [ETT-OCTO] https://kb.ettus.com/OctoClock_CDA-2990#OctoClock
- [SDR] https://es.wikipedia.org/wiki/Radio_definida_por_software
- [BAR05] S. Barbarossa, Multiantenna Wireless Communication Systems. Artech House. 2005.



Advanced research for everyday life