

Anexos

Anexo A

Guías de usuario

En el presente anexo, A, se facilitan las guías de usuario básicas para el correcto manejo de los dos simuladores desarrollados e implementados en este trabajo.

Índice del anexo	
A.1	Simulador 5G - NB-IoT 66
A.1.1	Inicio de la aplicación 66
A.1.2	Sistema transmisor 66
A.1.2.1	Generar señal de banda ancha 66
A.1.2.2	Agregar señal de banda estrecha (NB-IoT) 70
A.1.2.3	Modelado de elementos no lineales 72
A.1.3	Canal de transmisión 72
A.1.4	Visualización de resultados del transmisor 74
A.1.5	Sistema receptor 79
A.1.6	Menús 80
A.1.6.1	Menú de ayuda 80
A.1.6.2	Menú de opciones 80
A.2	Simulador 5G-NR 82
A.2.1	Inicio de la aplicación 83
A.2.2	Configuración de la señal 84
A.2.3	Modelo de elementos no lineales 85
A.2.4	Modelo de canal 86
A.2.5	Visualización de resultados 87
A.2.6	Menús 88
A.2.6.1	Menú de ayuda 89
A.2.6.2	Menú de opciones 89

A.1. Simulador 5G - NB-IoT

Este simulador nos permite estudiar el enlace descendente de un sistema de radio 5G, incluyendo agregación de señales de banda estrecha, modelado de elementos no lineales como el amplificador de potencia y condiciones reales de propagación para diversos escenarios.

A.1.1. Inicio de la aplicación

El usuario debe iniciar primeramente la aplicación ejecutando en la ventana de comandos del entorno software MATLAB® la instrucción *DLSimulator*. Esta abrirá el simulador y se mostrará al usuario la interfaz principal del mismo. Como se puede ver en la figura A.1, nos encontramos con las siguientes partes:

- **Sistema transmisor.** Permite generar tanto la señal de banda ancha como las de banda estrecha con una serie de parámetros a configurar por el usuario. Además, permite elegir si se quieren tener en cuenta y modelar los elementos no lineales.
- **Visualización de las señales y otros resultados.** En esta parte se permite representar todos los recursos y señales resultantes de la simulación de la parte del transmisor. Se incluyen visualizaciones de señales en los dominios del tiempo y frecuencia, matrices de recursos o medidas como las de ACLR o EVM.
- **Modelado del canal.** El usuario puede seleccionar diversos escenarios con diferentes modelos de propagación, que modelan situaciones reales concretas.
- **Sistema receptor.** En esta parte se recoge todo lo relacionado con el receptor de la señal 5G. Incluye la demodulación completa de la señal recibida y medidas de calidad de la misma como la constelación o el BER.
- **Menús.** Se ofrece al usuario un menú de ayuda sobre las principales partes del simulador y un menú de opciones en el que cambiar y ajustar parámetros o exportar datos.

A.1.2. Sistema transmisor

A.1.2.1. Generar señal de banda ancha

En esta sección, el usuario puede generar la señal 5G, es decir, la señal de banda ancha. Los parámetros que se permite modificar en su generación son los que se

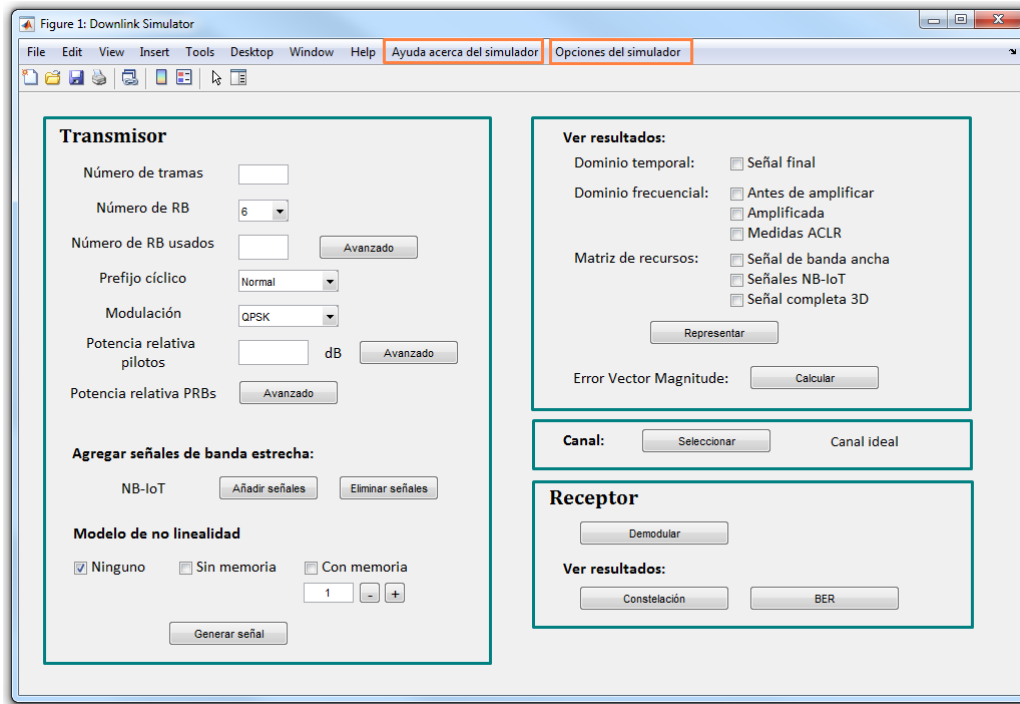


Figura A.1: Interfaz de usuario inicial del simulador 5G – NB-IoT con las partes principales de las que consta.

detallan a continuación.

Número de tramas a generar

Permite definir la longitud de la señal generada tomando las tramas como unidad de medida. Para más información sobre la estructura de tramas, consultar el apartado 1.1.3.

Número de RBs

Define el número de bloques de recursos del sistema de transmisión, es decir, el número máximo de bloques de recursos utilizables. Según el estándar tenemos 6 posibilidades, como se detalla en la tabla A.1.

Número de RBs usados

En este apartado se puede configurar la cantidad de recursos utilizados para realizar la transmisión. Esta asignación la podemos llevar a cabo en el simulador de dos maneras, de un modo simple o de otro avanzado. El modo sencillo de hacerlo consiste, simplemente en introducir en la casilla correspondiente, como se detalla en la figura A.2, el número de RBs a usar. Estos se rellenarán consecutivamente empezando por los de frecuencias más bajas.

N.º de RB	Ancho de banda (MHz)
6	1,4
15	3
25	5
50	10
75	15
100	20

Tabla A.1: Anchos de banda y su correspondiente número de RB permitidos según el estándar.

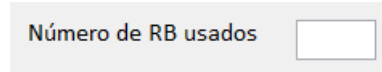


Figura A.2: Configuración del número de bloques usados para el simulador 5G – NB-IoT.

Para una asignación más precisa y con mayor libertad para el usuario se usa el modo avanzado. Accediendo a la pestaña, aparecen una serie de recuadros en los que el usuario puede rellenar, para cada subtrama los números de bloques que quiere usar, teniendo en cuenta lo siguiente:

- La numeración es *0-based*. Esto conlleva que los números de RB seleccionables serán de 0 a N_{RB}^{DL}
- El formato de los datos a introducir será el siguiente: 0, 3, 4 para indicar que se quieren usar los bloques 0, 3 y 4; 0 – 3 para indicar que se quieren usar los bloques desde el 0 hasta el 3, ambos inclusive.

A continuación, en la figura A.3, se detalla un ejemplo. En este caso, para la subtrama 0 se usarían los bloques 0, 3, 4 y 5.

Prefijo cíclico

Como se ha detallado anteriormente, el estándar permite dos longitudes de prefijo cíclico, normal y extendido, para adaptar la señal a las condiciones de transmisión, esta opción permite elegir entre ambas. Resaltar que, si queremos añadir señales de banda estrecha, sólo se podrá utilizar el prefijo normal.

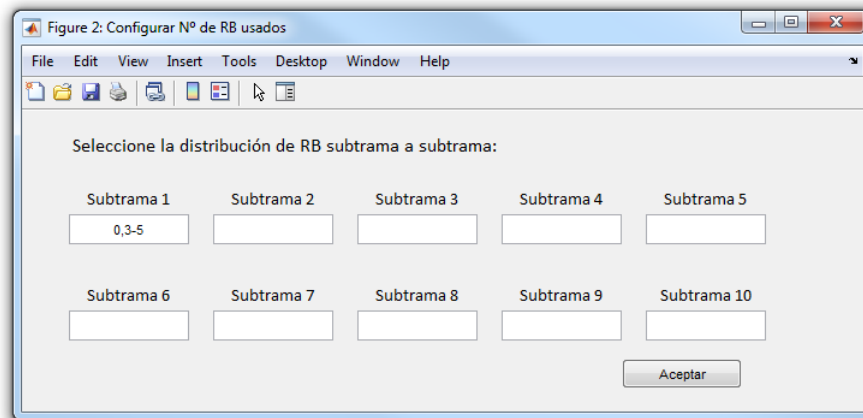


Figura A.3: Configuración del número de bloques usados para el simulador 5G – NB-IoT en modo avanzado (edición subtrama a subtrama).

Modulación

En este apartado, el usuario puede seleccionar entre las distintas modulaciones contempladas en el estándar. Como se ha explicado en el apartado 1.1.1.2, las modulaciones posibles son: QPSK, 16-QAM, 64-QAM, 256-QAM y 1024-QAM.

Niveles de potencia relativa de las señales piloto

Permite al usuario variar la potencia relativa de las señales pilotos respecto al resto de la señal. El rango dinámico permitido viene dado por el estándar y va de -3 dB y +6 dB. En este caso, en el modo sencillo podemos seleccionar el incremento de potencia que tendrán todas las señales piloto de la trama, mientras que, en el caso avanzado, podemos variarlo subtrama a subtrama.

En todas las casillas, el valor deberá introducirse en dB.

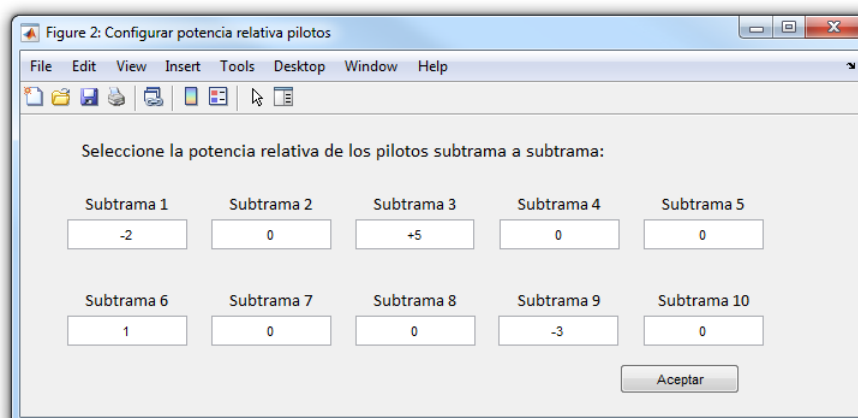


Figura A.4: Configuración de la potencia relativa de las señales piloto en el simulador 5G – NB-IoT en modo avanzado.

Niveles de potencia relativa de cada PRB

Esta opción nos permite variar la potencia relativa de los PRBs con respecto al resto de la señal. El funcionamiento es el mismo que en el caso anterior, con la salvedad de que, en este caso, sólo se da la opción avanzada. En ella se puede modificar la potencia subtrama a subtrama y para cada PRB, escribiendo los valores en dB y separados por comas.

A.1.2.2. Agregar señal de banda estrecha (NB-IoT)

Una vez se han seleccionado los parámetros de la señal de banda ancha, el usuario puede, además, generar otras de banda estrecha e integrarlas con la primera, siempre que la configuración de ambas lo permita. Como se ha explicado, existen dos posibilidades a la hora de integrar las señales de banda estrecha, puede hacerse en banda o en banda de guarda. La distribución de recursos en las tramas generadas no es modificable, debido a que su influencia es mínima en las medidas que se realizan. En la configuración con un ancho de banda de 1.4 MHz no es posible introducir señales NB-IoT en ninguna de las dos posiciones.

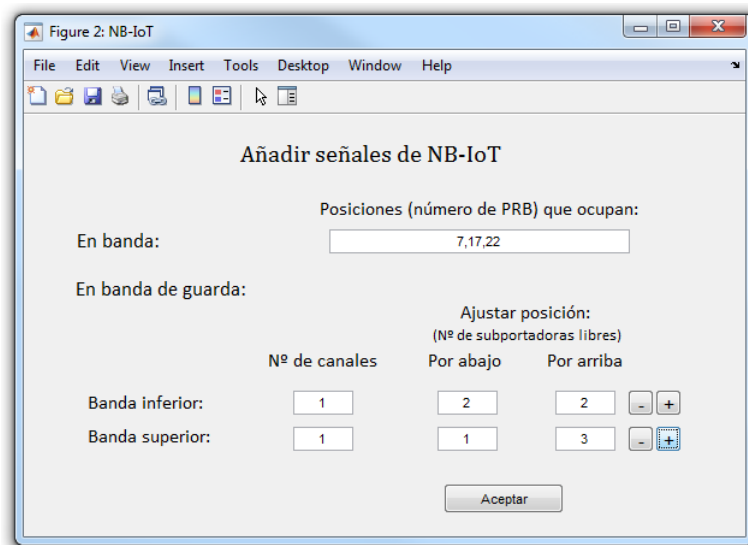


Figura A.5: Ventana de agregación de señales NB-IoT en el simulador 5G – NB-IoT.

Agregar señales NB-IoT en banda

Una vez hemos accedido a la ventana que permite configurar las señales NB-IoT a añadir, simplemente tenemos que introducir las posiciones en las que queremos añadir los canales. Los PRBs posibles para cada configuración son los que se detallan en la tabla A.2.

Ancho de banda del sistema	Índices de los PRBs en los que se pueden introducir las señales NB-IoT
1,4 MHz	—
3 MHz	2, 12
5 MHz	2, 7, 17, 22
10 MHz	4, 9, 14, 19, 30, 35, 40, 45
15 MHz	2, 7, 12, 17, 22, 27, 32, 42, 47, 52, 57, 62, 67, 72
20 MHz	4, 9, 14, 19, 24, 29, 34, 39, 44, 55, 60, 65, 70, 75, 80, 85, 90, 95

Tabla A.2: PRBs utilizables para transmitir señales NB-IoT en banda.

Agregar señales NB-IoT en banda de guarda

En la segunda mitad de la ventana, tenemos lo relativo a la introducción de canales NB-IoT en las bandas de guarda de la señal 5G. En la primera columna de celdas editables, el usuario puede introducir el número de canales que quiera en cada banda de guarda; el máximo de canales posibles para cada banda de guarda y para cada configuración de ancho de banda viene determinado en la tabla A.3. Como se puede ver, además de la configuración de 1,4 MHz, la de 3 MHz tampoco permite incluir señales NB-IoT en la banda de guarda, debido a que el tamaño de cada una de ellas es menor que el de un canal de IoT.

Ancho de banda del sistema (MHz)	Máximo número de canales que se pueden transmitir en la banda de guarda.
1,4	0
3	0
5	1
10	2
15	3
20	5

Tabla A.3: Número máximo de canales NB-IoT que se pueden transmitir en la banda de guarda

Una vez indicado el número de canales, podemos ajustar su posición dentro de la banda de guarda cambiando el número de subportadoras que quedan libres en torno a los canales IoT; es decir, entre los canales IoT y la señal de banda ancha y entre los canales IoT y el límite permitido para cada caso. El n.º de subportadoras libres por abajo indica el hueco hacia frecuencias bajas y viceversa.

Para borrar todos los canales generados, basta con hacer click en *Eliminar señales*. Al acceder a la ventana de añadir señales NB-IoT, también se hace un reset y se borran las señales IoT que hubiera anteriormente.

A.1.2.3. Modelado de elementos no lineales

En esta sección del simulador se puede configurar si se quiere o no considerar los efectos no lineales que causaría el amplificador de potencia, la fibra óptica u otros elementos. En el caso que se tenga en cuenta, se puede elegir entre modelar el dispositivo utilizando un modelo sin memoria o un modelo con memoria. El primero es más simple y da buenos resultados en los casos en que el ancho de banda de la señal es pequeño; para las configuraciones del sistema que requieren mayores anchos de banda se recomienda seleccionar el modelo con una memoria de 1 o 2.

Al elegir una memoria de 1 se usaría un modelo en el que para calcular la muestra de salida en un determinado instante de tiempo se tiene en cuenta la muestra actual de entrada y la anterior; igualmente, al seleccionar memoria igual a 2, se tendrían en cuenta las muestras de entrada de los dos instantes inmediatamente anteriores. En los casos que se modelan dichos elementos no lineales, se añade también una etapa de linealización mediante predistorsión digital para mejorar las prestaciones. Para más información acerca de los algoritmos de modelado, consultar el apartado 2.2 de la memoria.

El modelado se realiza a partir de señales reales capturadas en el laboratorio a entrada y salida del sistema no lineal (que puede incluir otros elementos adicionales). Estas señales se seleccionan desde el menú "seleccionar señales de entrenamiento". En él se tiene que elegir un fichero .mat en el que tengamos guardadas las señales de entrada y salida, llamadas s_{in} y s_{out} respectivamente, de la misma longitud, además de la frecuencia de muestreo (F_s). Por otra parte, se permiten ajustar las condiciones en las que se han tomado las medidas: atenuación introducida en la medida de s_{in} , atenuación en la medida de s_{out} e impedancia de referencia sobre la que se ha medido. En el caso de que no se modelen estos efectos, se ajusta la potencia de salida a una referencia de -18 dBm.

A.1.3. Canal de transmisión

En esta parte podemos configurar el modelo de canal a aplicar para modelar los efectos de la transmisión. Haciendo click en el botón *Seleccionar*, se accede a la ventana de configuración del canal. En ella, como se puede ver en la figura A.6, se

muestran todos los tipos de canal posibles, divididos en tres grupos: condiciones de propagación sin desvanecimiento, condiciones de propagación con desvanecimiento por multicamino y condiciones de propagación cambiantes.

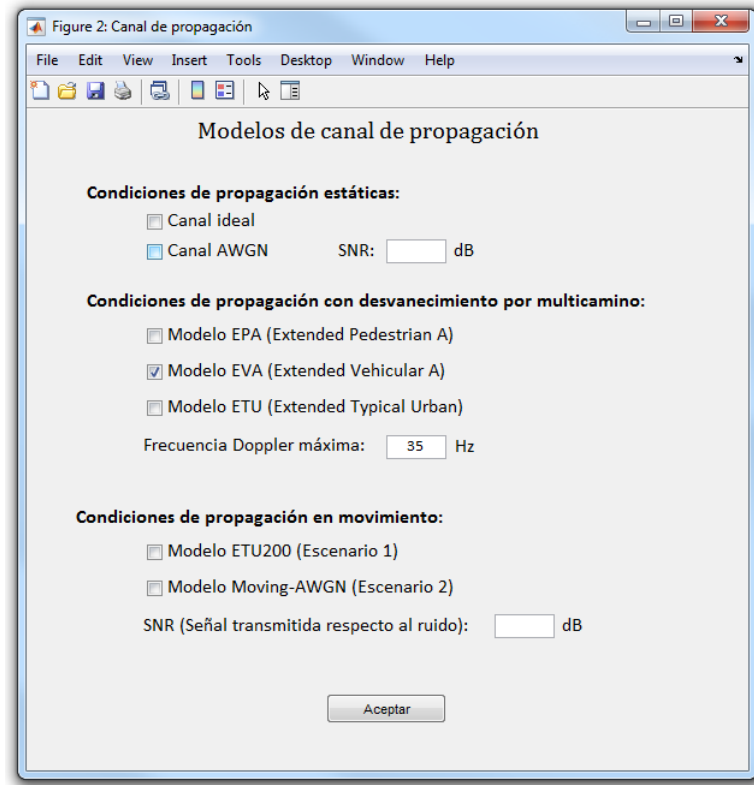


Figura A.6: Ventana de configuración del modelo de canal de propagación en el simulador 5G – NB-IoT.

Condiciones de propagación sin desvanecimiento

En ellos, la potencia media de la señal se mantiene constante. Tenemos dos tipos: el canal ideal y canal AWGN.

- **Canal ideal.** En este modelo de canal, la señal transmitida es igual que la recibida.
- **Canal AWGN.** Este canal añade un ruido aditivo, blanco y Gaussiano (AWGN) a la seña transmitida. La relación señal a ruido de la señal recibida (en dB) puede ser configurada por el usuario. Por defecto, la SNR es de 10 dB.

Condiciones de propagación con desvanecimiento por multicamino

Estos canales modelan utilizando la función de distribución Rayleigh condiciones de

propagación en las que existe un desvanecimiento por multicamino. En este caso, los desvanecimientos según las diferencias de retardo se mantienen constantes. Tenemos tres tipos: modelo EVA, modelo EPA y modelo ETU.

- **Modelo EPA.** El modelo *Extended Pedestrian A* modela condiciones de baja movilidad. La máxima frecuencia Doppler para este caso es de 5 Hz.
- **Modelo EVA.** El modelo *Extended Vehicular A* modela condiciones de movilidad media, típicas para comunicaciones desde vehículos. En este caso, la frecuencia Doppler máxima permitida es de 70 Hz.
- **Modelo ETU.** El modelo *Extended Typical Urban* modela condiciones de alta movilidad en entornos urbanos. Para este caso se permite una frecuencia Doppler de hasta 300 Hz.

Condiciones de propagación cambiantes

Estos modelos implementan canales cuyos perfiles de retardo varían con el tiempo. Tenemos dos modelos: el ETU-200 y un Moving-AWGN.

- **Modelo ETU-200.** Modelo que implementa un canal típico urbano con un desplazamiento Doppler de hasta 200 Hz, retardos variables siguiendo una función senoidal y modelo de desvanecimiento Rayleigh. Modela velocidades del UE de hasta 120 Km/h.
- **Modelo Moving-AWGN.** Modela un canal sin desvanecimiento pero que introduce retardos variables que se traducen en rotaciones de fase. Permite modelar altas velocidades.

A.1.4. Visualización de resultados del transmisor

Para una correcta interpretación y evaluación de la simulación, se ofrecen al usuario una serie de medidas a través de la ventana de comandos y unas representaciones que pretenden ilustrar y cuantificar los resultados obtenidos. Al generar una señal, en la ventana de comandos se puede ver algo similar a lo siguiente:

```
*** Ejecucion del transmisor ***
PAPR = 14,70 dB
Ganancia sin linealizacin : 42,12 dB
Ganancia con linealizacin : 39,39 dB
Diferencia de ganancia : - 2,73 dB
Numero de muestras que sobrepasan el limite : 70 (0,026042%)
```

Las figuras que se ofrecen al usuario son las siguientes:

Señal final

Se representa en el dominio del tiempo la señal que se transmite. En el caso de que se aplique un modelo de amplificador, la que se transmite es la señal linealizada y amplificada; en caso de que no se aplique, directamente es la señal generada.

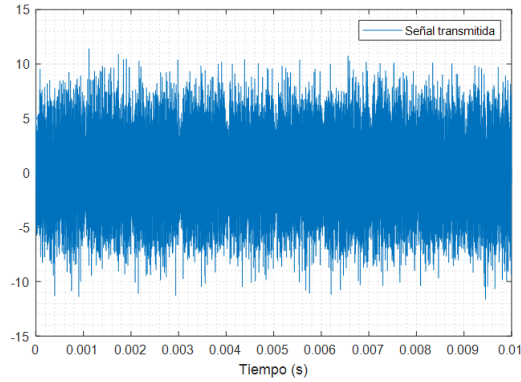


Figura A.7: Representación de la señal final transmitida en el dominio temporal en el simulador 5G–NB-IoT.

Señal antes de amplificar

Se visualiza en el dominio de la frecuencia la señal generada inicialmente, es decir, sin pasar la señal por el predistorsionador ni el modelo de amplificador.

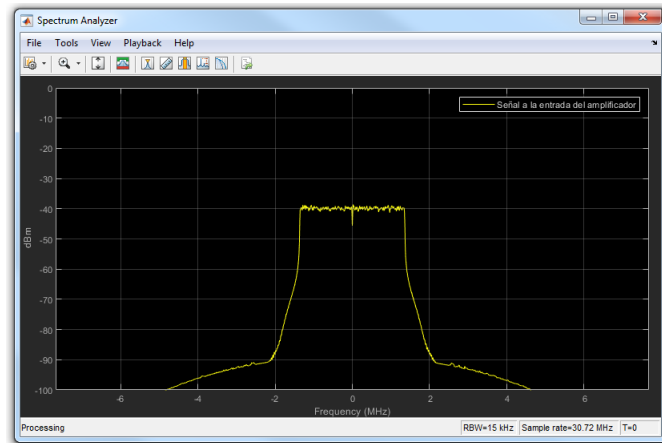


Figura A.8: Representación de la señal generada a la entrada del amplificador en el simulador 5G–NB-IoT.

Señal amplificada

En esta representación, también en el dominio de la frecuencia, se muestra al usuario la señal linealizada tras pasar por el amplificador, comparándola con la señal generada inicialmente. En el caso de que no se use amplificador las dos señales representadas van a ser la misma. En el caso de usar amplificador, se representa

también la señal amplificada sin linealizar, para poder ver el compromiso que existe entre linealidad y ganancia obtenida. Como se puede ver en la figura A.9, tenemos también la medida del ancho de banda ocupado (útil) y la potencia de la señal en ese ancho de banda, por lo que podemos medirlo para todas las señales que se comparan. A la derecha de la representación, se puede cambiar la vista, a como aparece en la figura A.10, donde se muestran los canales adyacentes y el ACLR para cada uno.

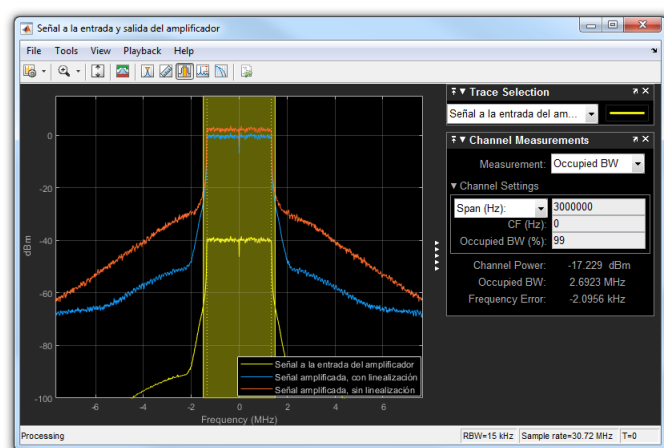


Figura A.9: Representación del espectro de la señal generada antes y después de ser amplificada, con y sin linealización.

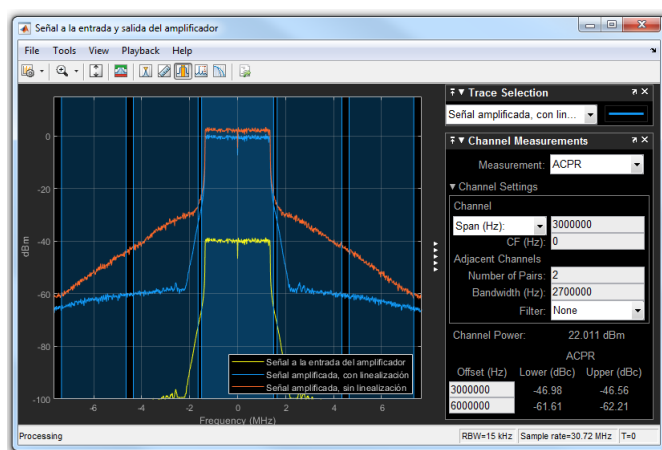


Figura A.10: Representación del espectro de la señal generada antes y después de ser amplificada, con y sin linealización. Se muestran los niveles de ACLR y la posición de los canales adyacentes.

Medidas ACLR

En este apartado aparecen dos representaciones diferentes. En una, aparecen las medidas de *ACLR*, para los cuatro canales adyacentes más cercanos a la señal. En la otra, se muestra una representación en frecuencia de la señal transmitida (la que

se puede ver en tiempo en el primer punto), en la que se indican medidas acerca de la potencia de la señal.

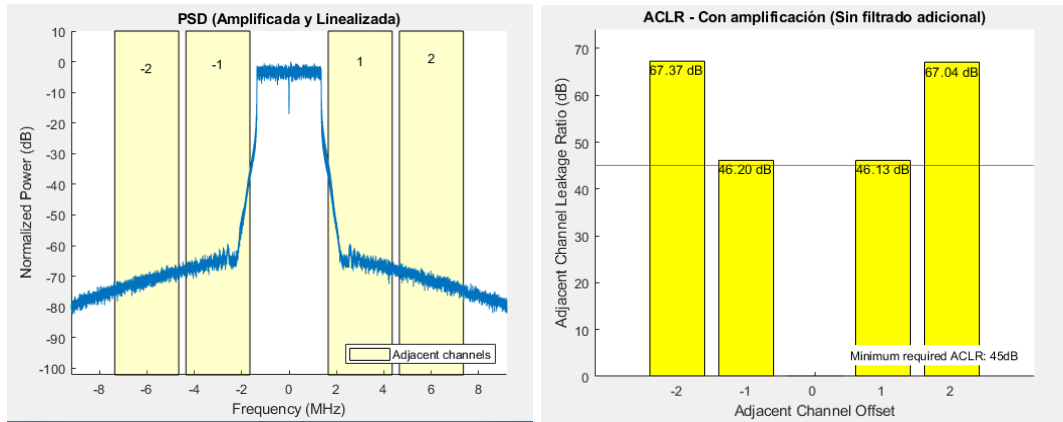


Figura A.11: Gráficas mostradas en el simulador 5G – NB-IoT en relación a las medidas de ACLR.

Además, en la ventana de comandos se ofrece información numérica acerca de las medidas y los parámetros usados para obtenerlas:

```
*** Resultados ACLR ***
Ancho de banda : 1,40 MHz
Ancho de banda util : 1,08 MHz
Frecuencias centrales de los canales adyacentes (MHz) : [ - 2,80 - 1,40 1,40 2,80]
Adjatent Channel Leakage Ratio (dB) : [65,70 46,67 45,50 64,68]
```

Señal de banda ancha

Muestra al usuario cómo se rellena la matriz de recursos en una de las tramas de la señal de banda ancha (sin incluir canales NB-IoT) que hemos generado, es decir, dónde se sitúan cada uno de los canales que se están transmitiendo y qué huecos están vacíos, tal y como se puede ver en la figura A.12. Independientemente del número de tramas que se haya indicado generar, se representará solamente una de ellas, ya que la distribución va a ser la misma en todas.

Señales NB-IoT

Se representa la distribución de las tramas de los canales NB-IoT que se hayan generado, una representación para cada tipo de canal (en banda, en banda de guarda, *anchor*, *non-anchor*). Como en el caso anterior, cada tipo de canal tiene una distribución fija, por lo que sólo se mostrará la configuración de cada uno de los tipos de trama generados una vez, indicando las posiciones en las que se sitúa ese

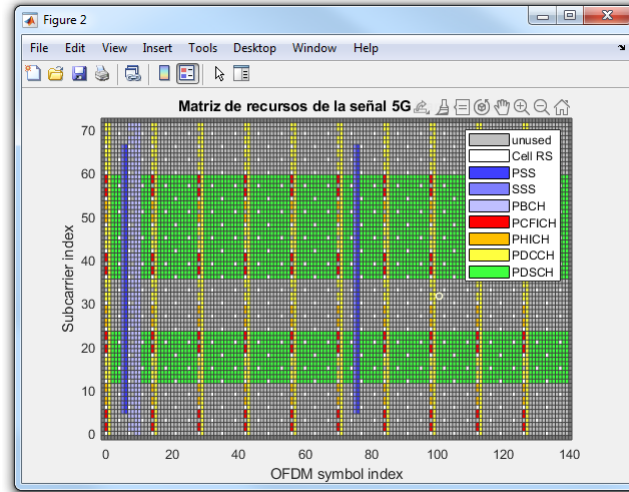


Figura A.12: Representación de la matriz de recursos de la señal 5G generada en el simulador 5G–NB-IoT. Se usa una configuración con un BW de 1,4 MHz, utilizándose los RB 1, 3 y 4.

canal. Como en este caso las tramas pares son distintas de las impares, se muestran dos tramas de cada tipo:

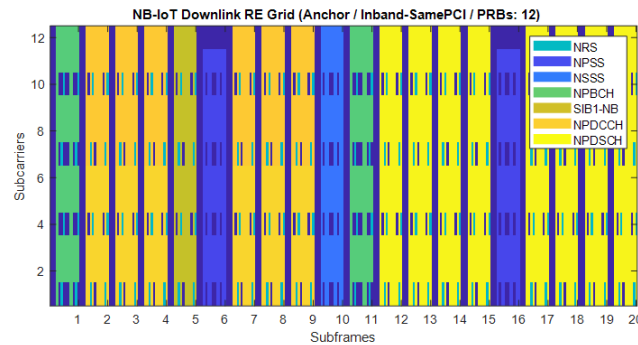


Figura A.13: Representación de la matriz de recursos de un canal IoT generado en el simulador 5G–NB-IoT.

Señal completa 3D

En este apartado aparecen dos representaciones. Por un lado el usuario puede ver la distribución en 3D de la señal de banda ancha, de manera que puede apreciar más claramente su distribución y, además, la potencia de cada símbolo transmitido, medida en dBm. Como en casos anteriores y por la misma razón, el eje de tiempos abarca únicamente una trama. Por otro lado, se muestra al usuario, tal y como se ve en la figura A.14, la configuración global de toda la señal transmitida, incluyendo tanto las señales de banda ancha como las de banda estrecha. Es importante tener en cuenta que, en este caso, para poder incluir las señales de banda estrecha que

se encuentran en banda de guarda, se toma como referencia para etiquetar las subportadoras la matriz con el número de puntos para los que se hace la IFFT en cada caso, en vez de la matriz reducida con solamente las portadoras activas.

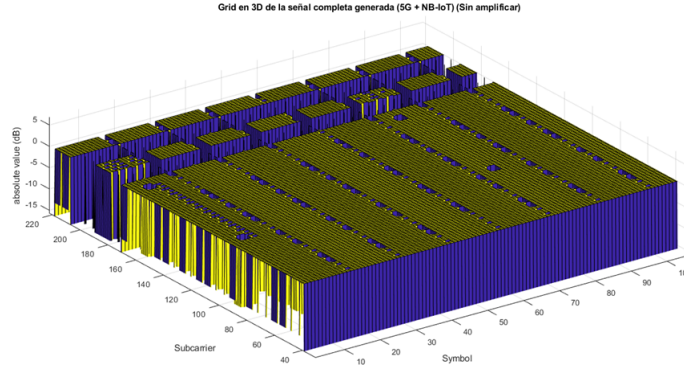


Figura A.14: Representación en 3D de la señal generada, agregando señales 5G y NB-IoT, antes de pasar por el amplificador. Se muestra una señal 5G con un BW de 15 RB, usándose del 0 al 10 y el 14, con transmisión IoT en banda en el PRB 12.

Por último, se pueden calcular los valores de EVM mostrados en %, normalizando al valor del módulo de la potencia media de los símbolos de la constelación ideal. Las representaciones exponen la evolución del EVM con respecto a las dimensiones del tiempo y de la frecuencia, en concreto, según el símbolo OFDM, el número de slot, el número de RB y el número de subportadora. Además, en todos los casos se muestra el valor rms y el valor de pico alcanzado. Para las señales de banda estrecha transmitidas se muestra también el EVM, en este caso, en función del símbolo OFDM y en función del número de subportadora.

Además de las representaciones, se muestran los valores numéricos por la ventana de comandos.

A.1.5. Sistema receptor

El sistema receptor simula la demodulación de la señal. Para ello, realiza las pertinentes ecualizaciones y corrección de offsets y extrae los símbolos recibidos. Una vez se ha llevado a cabo la demodulación, se ofrece información para la evaluación de la transmisión que se ha simulado. Por un lado, se puede ver cualitativamente la calidad de la misma mediante la constelación y, por otro, se muestra una cuantificación de esta degradación mediante el BER, ofrecido a través de la ventana de comandos.

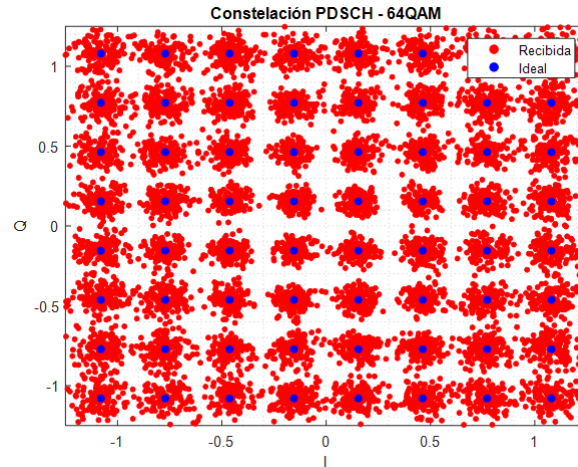


Figura A.15: Representación la constelación recibida para una simulación con modulación 64-QAM.

A.1.6. Menús

En la barra de herramientas de la ventana principal del simulador se incluyen dos menús: "Ayuda acerca del simulador" y "Opciones del simulador". El primero, ofrece al usuario una ayuda básica sobre el simulador y, el segundo, permite cambiar el modo de funcionamiento del simulador, ajustar parámetros y exportar señales, como se va a detallar a continuación.

A.1.6.1. Menú de ayuda

Para cada una de las secciones relevantes, se ofrece una pequeña ayuda, con los aspectos más relevantes de la presente guía, para poder acceder a ellos de una manera más ágil.

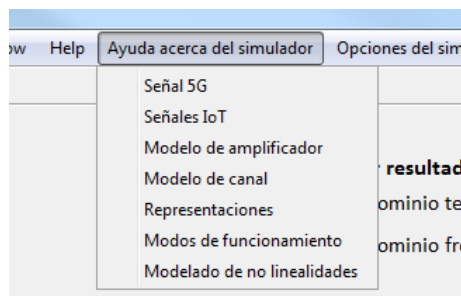


Figura A.16: Menú de ayuda en el simulador 5G–NB-IoT.

A.1.6.2. Menú de opciones

Modo de evaluación

En el submenú "Modo de evaluación", el usuario puede cambiar el modo de

funcionamiento del simulador entre: evaluar integración de señales y evaluar efectos de cumplimiento de máscara.

- **Evaluar integración de señales.** Funcionamiento según lo descrito en la guía hasta este punto. Permite ver cómo se ve afectada la calidad de la señal en diversas configuraciones y agregando señales de diferente naturaleza.
- **Evaluar efectos de cumplimiento de máscara.** En este modo, se simulan los efectos que aparecen en la señal si nos centramos en cumplir la máscara y niveles de ACLR requeridos. Para ello, se aplica a la señal un filtrado bastante restrictivo antes del amplificador.

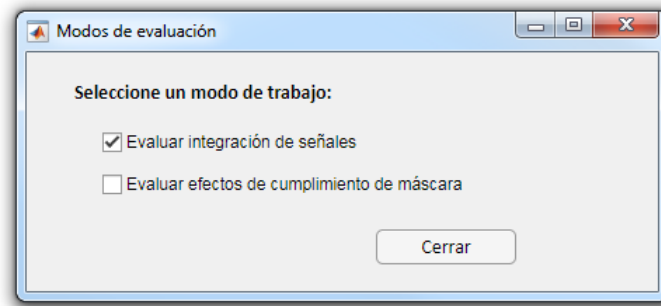


Figura A.17: Ventana de modos de funcionamiento en el simulador 5G–NB-IoT.

Ajustar señales de entrenamiento

En el submenú "ajustar señales de entrenamiento", se puede seleccionar las señales de entrada y salida con las que modelar el sistema no lineal (amplificador o amplificador + fibra), como se puede ver en la figura A.18. Además, se permite el ajuste de varios parámetros en función de las condiciones de medida en las que se hayan tomado las señales. La atenuación en la medida habría que tenerla en cuenta si se ha utilizado un atenuador como protección a la hora de realizar las medidas y la impedancia de referencia es la impedancia sobre la que se realizan las medidas, por defecto 50 Ω . De este modo, se tiene total libertad sobre el sistema a modelar.

Para el correcto funcionamiento, dentro del fichero de matlab (.mat), las señales de entrada y salida deben estar guardadas como *s_in* y *s_out*, respectivamente, y tener la misma dimensión.

Exportar señales

En este submenú, el usuario puede guardar señales generadas durante la simulación en su espacio de trabajo, tal y como se indica en la figura A.19. Las señales

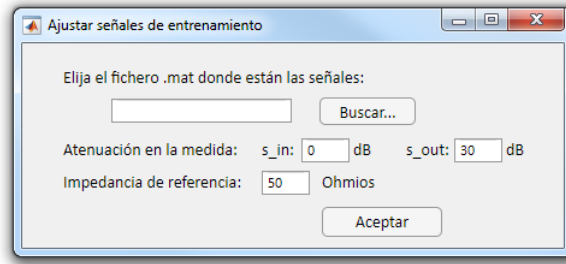


Figura A.18: Ventana que permite seleccionar las señales usadas para el modelado de los elementos no lineales en el simulador 5G–NB-IoT.

que se permiten guardar son las correspondientes a antes y después de la etapa predistorsión-amplificación. Se puede elegir qué señal, o ambas, guardar en un fichero cuyo nombre es elegido también por el usuario. Además de las señales seleccionadas, en el fichero se guarda también la frecuencia de muestreo de dichas señales. Cada una de estas variables, se almacena en el fichero ".mat" con los siguientes nombres:

- **txWaveform**: Hace referencia a la señal sin amplificar.
- **txAmpWaveform**: Hace referencia a la señal amplificada y linealizada.
- **Fs**: Frecuencia de muestreo.

Es importante destacar que, si estamos en el segundo modo de evaluación del simulador, las señales que se exportarán serán las filtradas.

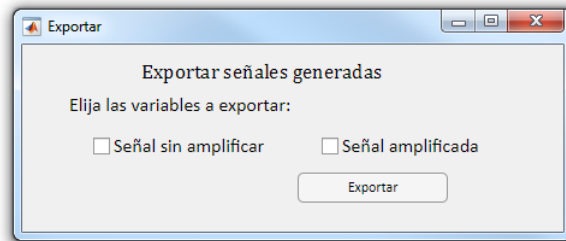


Figura A.19: Ventana de exportación de señales en el simulador 5G – NB-IoT.

A.2. Simulador 5G-NR

Esta segunda parte del simulador nos permite estudiar el enlace descendente de un sistema de radio 5G-NR, en este caso no nos centramos en el estudio de los efectos de integrar señales de diferente naturaleza, sino en los efectos que se introducen debido a las nuevas numerologías o anchos de banda. Además, se incluye modelado de elementos no lineales como el amplificador de potencia y condiciones reales de propagación para diversos escenarios.

A.2.1. Inicio de la aplicación

Lo primero de todo, debemos iniciar el simulador, ejecutando en la ventana de comandos del entorno software MATLAB® la instrucción *DL_5GNR_Simulator*. Esta abrirá la aplicación y se mostrará al usuario la interfaz principal de la misma. Como se puede ver en la figura A.20, nos encontramos con las siguientes partes:

- **Configuración de la señal.** Permite ajustar los parámetros característicos de la señal que queremos simular.
- **Selección del modelo de no linealidad.** En ella, se puede elegir el modelo que queremos utilizar para el modelado de elementos no lineales.
- **Selección del modelo de canal.** En esta sección, se puede configurar el modelo de canal de transmisión que queremos aplicar.
- **Visualización de resultados.** Permite al usuario, mediante una serie de representaciones, ver los resultados de la simulación.
- **Menús.** Hay un menú de ayuda con las pautas básicas para el manejo del simulador y otro de opciones que permite ajustar parámetros o exportar señales.

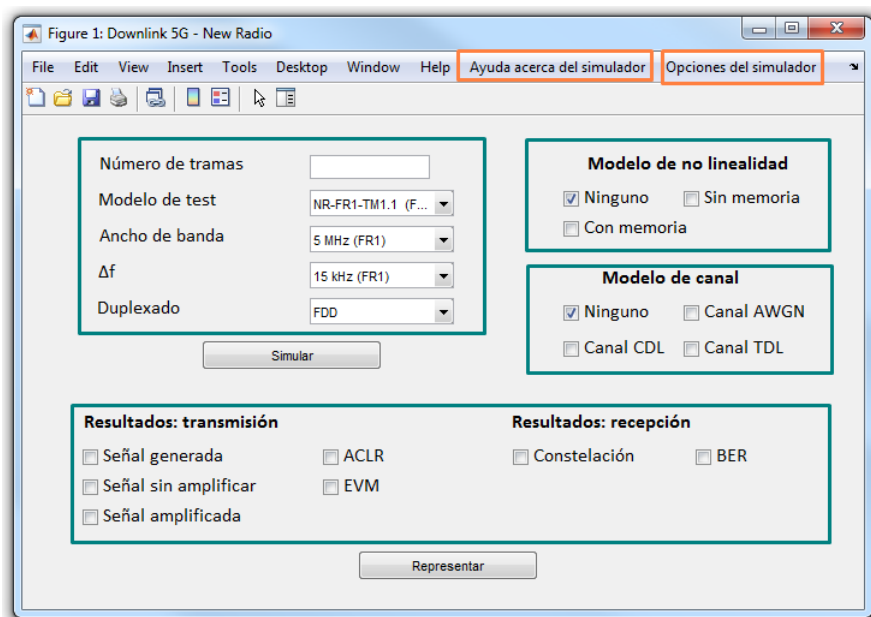


Figura A.20: Interfaz de usuario inicial y partes del simulador 5G-NR.

A.2.2. Configuración de la señal

Número de tramas

El usuario puede configurar la longitud temporal de la señal que se genera. Esta longitud influirá para realizar las representaciones o a la hora de exportarla, sin embargo, los cálculos de EVM y BER se harán solo con la primera trama.

Modelo de test

Como se ha apuntado previamente, existen dos rangos de frecuencia definidos para señales 5G-NR, uno sub-6GHz y otro en la banda de las ondas milimétricas:

Frequency Range 1	Frequency Range 2
450 MHz - 6000 MHz	24520 MHz - 52600 MHz

Tabla A.4: Rangos de frecuencia definidos para señales 5G-NR.

Para cada una de las dos bandas se proponen una serie de modelos definidos en el estándar para la evaluación de prestaciones. Dichos modelos de test se explican con un mayor detalle en [31, pp. 44-94] para las frecuencias FR1 y los definidos en [32, pp. 50-53] para FR2.

En el simulador, al seleccionar el desplegable, aparece cada uno de ellos, indicándose el FR en el que se encuentra, la modulación utilizada, el ancho de banda rellenado y como se distribuye la potencia. En los casos *full band*, se rellena todo el ancho de banda mientras que en los *single PRB* se transmite solamente un PRB. Por otra parte, en los casos *uniform* la potencia se distribuye uniformemente, pero en los *boosted/deboosted* hay PRBs con más potencia que otros.

Separación entre subportadoras. Numerología.

En este apartado podemos elegir entre las diferentes numerologías 5G permitidas, que determinan la separación entre subportadoras (SCS). Esta separación viene definida por:

$$SCS = 2^\mu \cdot 15kHz, \text{ con } \mu = 0, 1, 2, 3 \quad (\text{A.1})$$

Para cada numerología se definen el resto de parámetros según lo especificado a continuación. Resaltar que, aunque la numerología hace variar la duración de la ráfaga de información, no vamos a entrar en ello en este trabajo porque en cada modelo de test se define como se rellena la matriz de recursos.

Ancho de banda

Según el rango de frecuencias (FR1 o FR2) en el que trabajamos, se definen una serie de anchos de banda y numerologías permitidas. Las cuales se recogen en las tablas A.5 para FR1 y A.6 para FR2.

BW (MHz)	5	10	15	20	25	30	40	50	60	80	90	100
$\mu = 0$	25	52	79	106	133	160	216	270	–	–	–	–
$\mu = 1$	11	24	38	51	65	78	106	133	162	217	245	273
$\mu = 2$	–	11	18	24	31	38	51	65	79	107	121	135

Tabla A.5: Anchos de banda definidos para las diferentes numerologías en señales 5G-NR en FR1. Se especifica para ellas el número de RB.

BW (MHz)	50	100	200	400
$\mu = 2$	66	132	264	–
$\mu = 3$	32	66	132	264

Tabla A.6: Anchos de banda definidos para las diferentes numerologías en señales 5G-NR en FR2. Se especifica para ellas el número de RB.

Duplexado

Permite seleccionar el tipo de duplexado utilizado en la transmisión, pudiendo elegir entre duplexado en el dominio de la frecuencia (FDD) o duplexado en el dominio del tiempo (TDD);

A.2.3. Modelo de elementos no lineales

En esta parte del simulador, se puede elegir si tener en cuenta o no los efectos que causarían los elementos no lineales en el transmisor, como pueden ser el amplificador de potencia o la fibra óptica, igual que en el simulador anterior, pudiendo elegir entre un modelado con memoria o sin memoria. El primero es más simple y da buenos resultados en los casos en que el ancho de banda de la señal es pequeño; para las configuraciones del sistema que requieren mayores anchos de banda se recomienda seleccionar el modelo con una memoria de 1 o 2.

Por ejemplo, al elegir una memoria de 1 se usaría un modelo en el que para calcular la muestra de salida en un determinado instante de tiempo se tiene en cuenta la muestra actual de entrada y la anterior y, análogamente para memoria igual a 2, influirían las muestras de dos instantes anteriores.

En los casos que se utiliza un modelo de amplificador, se añade también una etapa de linealización mediante DPD para mejorar las prestaciones. Para más información acerca de los algoritmos de modelado, consultar el apartado 2.2 de la memoria.

El modelado se lleva acabo usando señales reales capturadas previamente a entrada y salida del sistema no lineal (que puede incluir otros elementos adicionales) y almacenadas en un fichero .mat. Este archivo se selecciona desde el menú "seleccionar señales de entrenamiento". Es importante guardar las señales como s_{in} y s_{out} , respectivamente, y con la misma longitud, además de la frecuencia de muestreo (F_s). Por otra parte, se permiten ajustar las condiciones en las que se han tomado las medidas: atenuación introducida en la medida de s_{in} , atenuación en la medida de s_{out} e impedancia de referencia sobre la

A.2.4. Modelo de canal

En esta parte se puede configurar el modelo de canal a aplicar. Haciendo click en el modelo deseado, se abre una ventana donde poder ajustar algunos parámetros adicionales o elegir el submodelo deseado según lo detallado a continuación. Para obtener más información acerca de los mismos, consultar [29, pp. 65-72].

Canal ideal (Ninguno)

Este es un modelo meramente teórico, en el que la señal recibida es igual a la transmitida, es decir, se eliminan los efectos que pueda causar el canal.

Canal AWGN

En este canal, la señal recibida es la transmitida más un ruido aditivo, blanco y Gaussiano (AWGN). La relación señal a ruido en recepción (en dB) puede ser configurada por el usuario. Por defecto, la SNR es de 10 dB.

Canales CDL

Los modelos *Clustered Delay Line* se utilizan especialmente para el simular entornos de propagación en transmisiones MIMO. Existen cinco modelos dentro de este grupo. Los modelos CDL-A, CDL-B y CDL-C, modelan condiciones NLOS, mientras que los CDL-D y los CDL-E, simulan condiciones de LOS.

Canales TDL

Son modelos *Tapped Delay Line*, que buscan representar canales con desvanecimiento usando la función de distribución Rayleigh. Dentro de este grupo, existen también

cinco modelos. Los tres primeros TDL-A, TDL-B y TDL-C, se han diseñado para simular condiciones NLOS, mientras que los otros dos, TDL-D y TDL-E, hacen lo propio con condiciones LOS. En los modelos TDL-D y TDL-E, debido a las condiciones de visión directa, también se incluye modelado utilizando la distribución Rice.

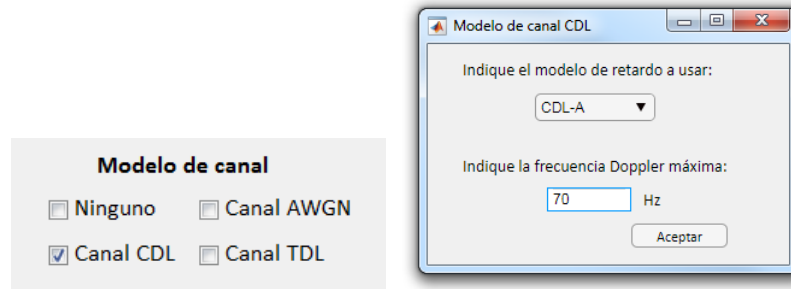


Figura A.21: Ventana de configuración de los modelos de canal de propagación en el simulador 5G-NR.

A.2.5. Visualización de resultados

En este apartado se ofrecen al usuario una serie de representaciones que buscan ilustrar y cuantificar los resultados obtenidos, facilitando así la interpretación y evaluación de los resultados de la simulación. Las figuras ofrecidas son las siguientes:

Señal generada

Se representa en el dominio del tiempo la señal generada por el transmisor, antes de realizarse la predistorsión y amplificación.

Señal sin amplificar y amplificada

Se muestran en el dominio de la frecuencia las señales antes y después de pasar por la etapa de predistorsión-amplificación. Se puede seleccionar qué señales se quieren ver en la misma figura, si se seleccionan las dos, se mostrará una comparativa de ambas. En el caso de haber seleccionado no usar amplificador, ambas señales serán iguales; en caso de aplicar un modelo, se mostrará también la señal sin linealizar, para poder ver la comparativa entre ellas y el compromiso que existe entre linealidad y ganancia. Además, se ofrece la medida de ancho de banda útil del canal y potencia dentro de la misma para cada señal.

Medidas de ACLR

En este caso, se ofrecen dos figuras diferentes. En la primera, aparece la densidad

espectral de potencia de la señal transmitida (la amplificada si se modela el amplificador o la generada directamente si no) en la que se indica la posición de los canales adyacentes y, en la segunda, un diagrama de barras que indica el ACLR para cada uno de estos canales adyacentes.

Constelación

Se muestra una representación en la que aparecen superpuestas la constelación ideal y la recibida para el canal de datos PDSCH. La forma de esta constelación coincide con la típica para la modulación seleccionada en la configuración de la señal transmitida. En el caso de que se hubieran combinado dos modulaciones diferentes, las constelaciones de ambas se dibujan superpuestas.

Medidas de EVM

Se ofrecen al usuario diferentes medidas de EVM para la señal 5G-NR enviada. Los valores se muestran en %, normalizando al valor del módulo de la potencia media de los símbolos de la constelación ideal. Las representaciones exponen la evolución del EVM con respecto al tiempo para las dimensiones del tiempo y de la frecuencia, en concreto, según el símbolo OFDM, el número de slot y el número de subportadora. Además, en todos los casos se muestra el valor rms y el valor de pico alcanzado, de la forma en que se puede ver en la figura A.22.

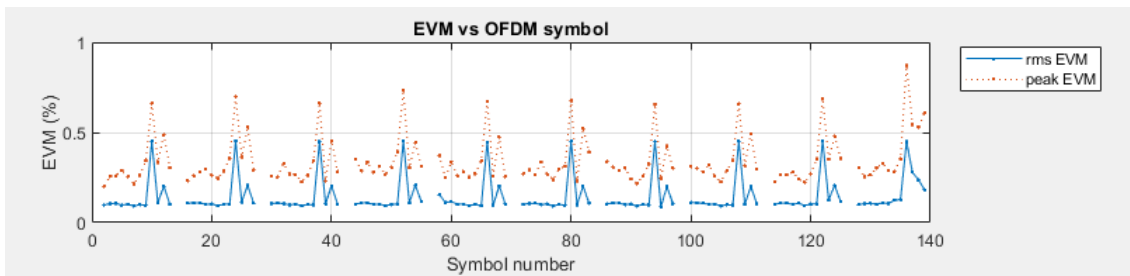


Figura A.22: Detalle representación EVM en el simulador 5G-NR, para el caso vs. símbolo OFDM.

A.2.6. Menús

En la barra de herramientas de la ventana principal del simulador se incluyen dos menús: "Ayuda acerca del simulador" y "Opciones del simulador". El primero, contiene una ayuda básica para facilitar el uso del simulador y, el segundo, permite cambiar algunos parámetros o exportar señales, como se detalla a continuación.

A.2.6.1. Menú de ayuda

Para cada una de las secciones que intervienen en el simulador se ofrecen al usuario las claves básicas para el correcto uso y funcionamiento del mismo, basadas en las explicaciones recogidas en esta guía.

A.2.6.2. Menú de opciones

Modo de evaluación

En este menú, como se ha mencionado, en el submenú "Modo de evaluación" se puede cambiar el modo de funcionamiento del simulador entre: evaluar configuración de las señales 5G-NR y evaluar efectos de cumplimiento de máscara.

- **Evaluar configuración de las señales 5G-NR.** El funcionamiento es tal cual se ha descrito en la guía hasta este punto. Nos centramos solamente en los efectos que producen elementos como el amplificador en la señal en función de la configuración de la misma, sin exigir requerimientos adicionales.
- **Evaluar efectos de cumplimiento de máscara.** En este caso, exigimos que la señal cumpla con los requerimientos de máscara de potencia necesarios, así como de ACLR. Para ello, se aplica un filtrado exigente antes de la etapa linealización-amplificación, de manera que se puedan ver los efectos que esto conlleva y comparar los resultados con los obtenidos en el otro modo.

Ajustar señales de entrenamiento

Por otra parte, en el submenú "ajustar señales de entrenamiento", se puede seleccionar las señales de entrada y salida con las que modelar el sistema no lineal (amplificador, amplificador + fibra, u otros) de manera análoga a como se ha explicado para el simulador 5G-NB-IoT.

Exportar señales

Igual que en el caso anterior, este menú permite al usuario exportar la señal antes y después de pasar por la etapa no lineal al directorio deseado. En función del modo de evaluación elegido, las señales exportadas estarán o no filtradas (primer y segundo modo, respectivamente).

Anexo B

Resultados de las simulaciones

En este anexo se recogen en formato tabla todas las medidas experimentales realizadas en el presente TFG, y a partir de las cuales se extrae todo lo expuesto en el capítulo 3. Además, se incluyen una serie de figuras representativas de los resultados que no se hayan incluido previamente en la memoria.

Índice del anexo	
B.1	Escenario 1: PRB en el medio del BW, no adyacente con canales IoT . . . 92
B.2	Escenario 2: PRB en el medio del BW, adyacente a un canal IoT . . . 93
B.3	Escenario 3: PRB en el extremo del BW, con canales IoT en banda de guarda. 94
B.4	Escenario 4: Canal IoT transmitido en banda 96
B.5	Escenario 5: Canal IoT transmitido en banda de guarda 97
B.6	Escenario 6: Evaluación del compromiso EVM/ACLR 99
B.7	Escenario 7: Evaluación de los efectos al cambiar de numerología . . . 101
B.8	Escenario 8: Simulación de modelos de canal 104

B.1. Escenario 1: PRB en el medio del BW, no adyacente con canales IoT

Situación	EVM avg. (%)	EVM max. (%)
A	0,095	0,101
B.1	0,123	0,153
B.2	0,114	0,132
C.1	0,122	0,131
C.2	0,111	0,123
D	0,214	0,234

Tabla B.1: EVM(%) para el PRB de referencia en el escenario 1, todos los PRB con la misma potencia.

Situación	EVM avg. (%)			
	$\Delta P = 0\text{dB}$	$\Delta P = 1\text{dB}$	$\Delta P = 2\text{dB}$	$\Delta P = 3\text{dB}$
B.2	0,111	0,152	0,110	0,105
C.2	0,109	0,100	0,100	0,095
D	0,211	0,227	0,220	0,220

Tabla B.2: EVM(%) promedio para el PRB de referencia en el escenario 1, variando su potencia.

Situación	EVM avg-max (%)			
	$\Delta P = 0\text{dB}$	$\Delta P = 1\text{dB}$	$\Delta P = 2\text{dB}$	$\Delta P = 3\text{dB}$
B.2	0,130	0,120	0,130	0,120
C.2	0,120	0,110	0,105	0,105
D	0,230	0,255	0,250	0,240

Tabla B.3: EVM(%) máximo para el PRB de referencia en el escenario 1, variando su potencia.

Situación	EVM avg (%)			
	$\Delta P = 0\text{dB}$	$\Delta P = 1\text{dB}$	$\Delta P = 2\text{dB}$	$\Delta P = 3\text{dB}$
B.2	0,112	0,125	0,135	0,151
C.2	0,113	0,115	0,144	0,167
D	0,211	0,229	0,281	0,352

Tabla B.4: EVM(%) promedio para el PRB de referencia en el escenario 1, variando la potencia de los PRBs adyacentes.

Situación	EVM avg. (%)			
	$\Delta P = 0\text{dB}$	$\Delta P = 1\text{dB}$	$\Delta P = 2\text{dB}$	$\Delta P = 3\text{dB}$
B.2	0,131	0,136	0,138	0,165
C.2	0,122	0,119	0,161	0,181
D	0,233	0,249	0,309	0,451

Tabla B.5: EVM(%) máximo para el PRB de referencia en el escenario 1, variando la potencia de los PRBs adyacentes.

B.2. Escenario 2: PRB en el medio del BW, adyacente a un canal IoT

Situación	EVM avg. (%)	EVM max. (%)
A	0,095	0,101
B.1	0,123	0,153
B.2	0,105	0,109
D	0,150	0,160

Tabla B.6: EVM(%) para el PRB de referencia en el escenario 2, todos los PRB con la misma potencia.

Situación	EVM avg. (%)			
	$\Delta P = 1\text{dB}$	$\Delta P = 2\text{dB}$	$\Delta P = 3\text{dB}$	$\Delta P = 4\text{dB}$
B.2	0,120	0,128	0,111	0,143
C.2	0,115	0,141	0,109	0,135
D	0,201	0,200	0,192	0,181

Tabla B.7: EVM(%) promedio para el PRB de referencia en el escenario 2, variando su potencia.

Situación	EVM max. (%)			
	$\Delta P = 1\text{dB}$	$\Delta P = 2\text{dB}$	$\Delta P = 3\text{dB}$	$\Delta P = 4\text{dB}$
B.2	0,137	0,140	0,150	0,170
C.2	0,131	0,151	0,125	0,156
D	0,220	0,213	0,202	0,195

Tabla B.8: EVM(%) máximo para el PRB de referencia en el escenario 2, variando su potencia.

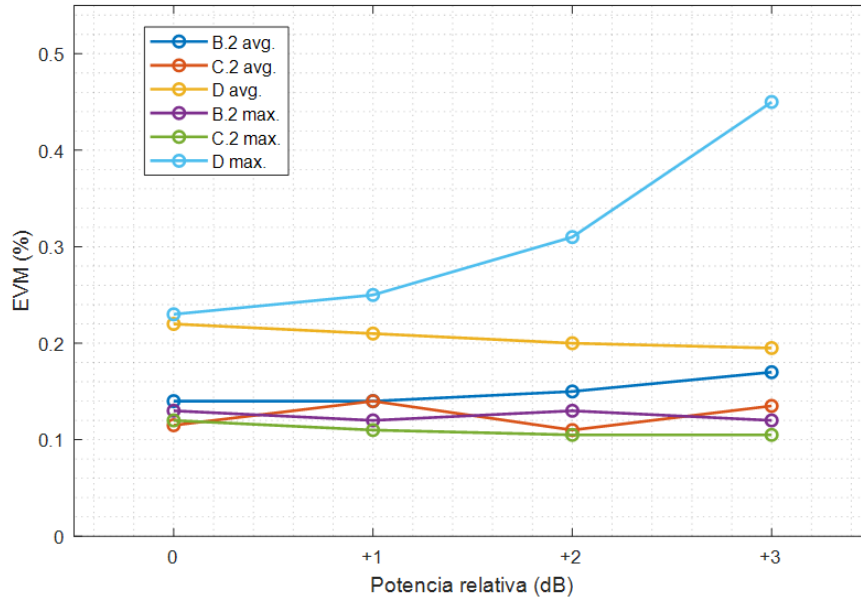


Figura B.1: EVM(%) para el PRB de referencia en el escenario 2, variando su potencia.

B.3. Escenario 3: PRB en el extremo del BW, con canales IoT en banda de guarda.

Situación	EVM avg. (%)	EVM max (%)
A	0.110	0.142
B.2	0,138	0.151
C.2	0,129	0,137
D	0,158	0,171

Tabla B.9: EVM(%) para el PRB de referencia en el escenario 3, todos con la misma potencia.

Situación	EVM avg. (%)			
	$\Delta P = 1\text{dB}$	$\Delta P = 2\text{dB}$	$\Delta P = 3\text{dB}$	$\Delta P = 4\text{dB}$
B.2	0,331	0,321	0,320	0,350
C.2	0,300	0,313	0,332	0,323
D	0,387	0,410	0,405	0,400

Tabla B.10: EVM(%) promedio para el PRB de referencia en el escenario 3, variando la potencia del PRB de referencia.

Situación	EVM max. (%)			
	$\Delta P = 1\text{dB}$	$\Delta P = 2\text{dB}$	$\Delta P = 3\text{dB}$	$\Delta P = 4\text{dB}$
B.2	0,401	0,380	0,400	0,379
C.2	0,392	0,382	0,373	0,387
D	0,450	0,550	0,461	0,481

Tabla B.11: EVM(%) máximo para el PRB de referencia en el escenario 3, variando la potencia del PRB de referencia.

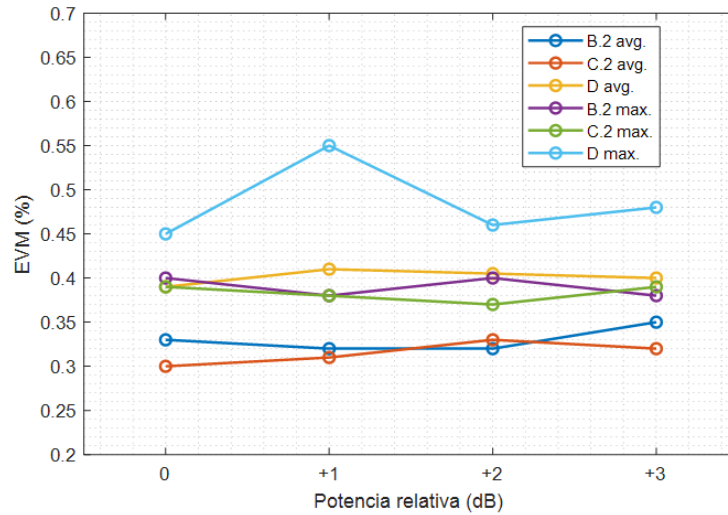


Figura B.2: EVM(%) para el PRB de referencia en el escenario 3, variando la potencia del PRB de referencia.

B.4. Escenario 4: Canal IoT transmitido en banda

Situación	EVM avg. (%)	EVM max. (%)
A	0,097	0,106
B.1	0,106	0,120
B.2	0,105	0,124
D	0,115	0,135

Tabla B.12: EVM(%) para un canal de IoT en el escenario 4, manteniendo constante la potencia de los PRBs adyacentes.

Situación	EVM avg. (%)			
	$\Delta P = 1\text{dB}$	$\Delta P = 2\text{dB}$	$\Delta P = 3\text{dB}$	$\Delta P = 4\text{dB}$
B.2	0,109	0,130	0,155	0,203
C.2	0,115	0,127	0,142	0,149
D	0,140	0,151	0,155	0,150

Tabla B.13: EVM(%) promedio para un canal de IoT en el escenario 4, variando la potencia de los PRBs adyacentes.

Situación	EVM max. (%)			
	$\Delta P = 1\text{dB}$	$\Delta P = 2\text{dB}$	$\Delta P = 4\text{dB}$	$\Delta P = 3\text{dB}$
B.2	0,125	0,161	0,201	0,348
C.2	0,135	0,150	0,209	0,180
D	0,152	0,180	0,196	0,193

Tabla B.14: EVM(%) máximo para un canal de IoT en el escenario 4, variando la potencia de los PRBs adyacentes.

B.5. Escenario 5: Canal IoT transmitido en banda de guarda

Separación (subportadoras)	EVM avg. (%)				
	$\Delta P = 0\text{dB}$	$\Delta P = 1\text{dB}$	$\Delta P = 2\text{dB}$	$\Delta P = 3\text{dB}$	$\Delta P = 4\text{dB}$
0	0,113	0,149	0,151	0,214	0,207
1	0,123	0,128	0,146	0,201	0,200
2	0,111	0,122	0,145	0,172	0,185
3	0,112	0,115	0,140	0,163	0,199
4	0,122	0,111	0,133	0,155	0,205
5	0,121	0,115	0,135	0,152	0,184

Tabla B.15: EVM(%) promedio para un canal de IoT en el escenario 5, variando la potencia de los PRBs adyacentes y cambiando la separación con la señal de banda ancha.

Separación (subportadoras)	EVM peak. (%)				
	$\Delta P = 0\text{dB}$	$\Delta P = 1\text{dB}$	$\Delta P = 2\text{dB}$	$\Delta P = 3\text{dB}$	$\Delta P = 4\text{dB}$
0	0,603	0,901	1,912	1,497	1,152
1	0,652	0,510	1,322	1,110	1,103
2	0,612	0,852	0,400	1,612	1,033
3	0,721	0,733	0,353	1,033	1,378
4	0,652	0,912	0,803	0,821	0,901
5	0,650	0,601	0,602	1,232	1,221

Tabla B.16: EVM(%) máximo para un canal de IoT en el escenario 5, variando la potencia de los PRBs adyacentes y cambiando la separación con la señal de banda ancha.

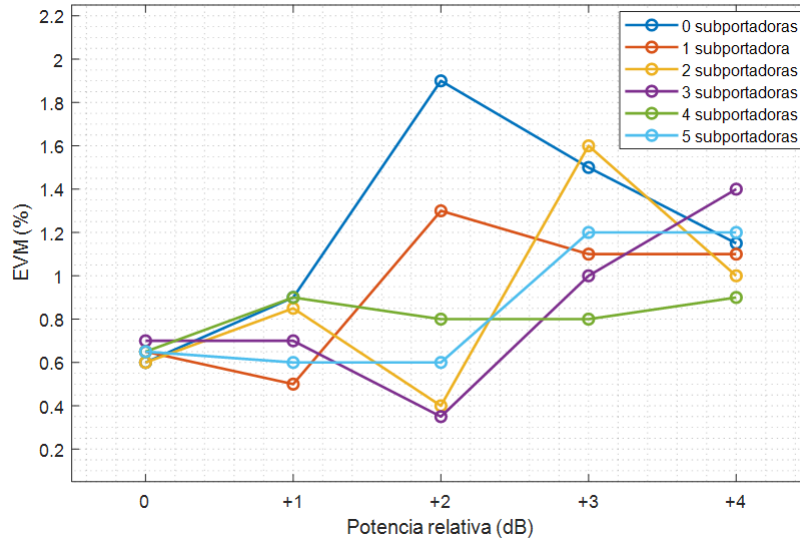


Figura B.3: EVM(%) máximo para un canal de IoT en el escenario 5, variando la potencia de los PRBs adyacentes y cambiando la separación con la señal 5G.

Canal NB-IoT	EVM avg. (%)				
	$\Delta P = 0\text{dB}$	$\Delta P = 1\text{dB}$	$\Delta P = 2\text{dB}$	$\Delta P = 3\text{dB}$	$\Delta P = 4\text{dB}$
Canal 0	0,161	0,210	0,251	0,309	0,517
Canal 1	0,163	0,221	0,250	0,321	0,520

Tabla B.17: EVM(%) promedio en canales de IoT en el escenario 5, variando la potencia de los PRBs adyacentes. Se transmiten dos canales en la banda de guarda.

Canal NB-IoT	EVM max. (%)				
	$\Delta P = 0\text{dB}$	$\Delta P = 1\text{dB}$	$\Delta P = 2\text{dB}$	$\Delta P = 3\text{dB}$	$\Delta P = 4\text{dB}$
Canal 0	0,183	0,222	0,301	0,361	0,578
Canal 1	0,212	0,231	0,300	0,398	0,591

Tabla B.18: EVM(%) máximo en canales de IoT en el escenario 5, variando la potencia de los PRBs adyacentes. Se transmiten dos canales en la banda de guarda.

B.6. Escenario 6: Evaluación del compromiso EVM/ACLR

Separación (en subportadoras)	EVM avg. (%)		EVM max. (%)	
	Señal 5G	Señal IoT	Señal 5G	Señal IoT
0	0,138	0,823	0,154	0,976
2	0,136	0,721	0,145	0,865
4	0,141	0,744	0,156	0,813
6	0,132	0,589	0,155	0,634
8	0,136	1,222	0,205	1,325
10	0,133	2,523	0,234	2,734
12	0,138	4,485	0,376	5,234
14	0,134	8,992	0,432	9,656
16	0,154	19,892	0,476	22,153

Tabla B.19: EVM (%) en el escenario 6 en el caso en que no se modelan los efectos del amplificador.

Situación	ACLR(dB)			
	Canal -2	Canal -1	Canal 1	Canal 2
Sin IoT, sin filtrar	79,25	50,04	51,46	79,23
Con IoT pegada, sin filtrar	78,87	45,22	45,65	78,88
Con IoT centrada, sin filtrar	78,91	45,61	44,21	78,91
Sin IoT, filtrado	128,21	120,39	121,30	128,48
Con IoT pegada, filtrado	129,48	121,02	120,80	129,38
Con IoT centrada, filtrado	128,60	120,00	120,06	128,61

Tabla B.20: ACLR en el escenario 6 en el caso en que no se modelan los efectos del amplificador.

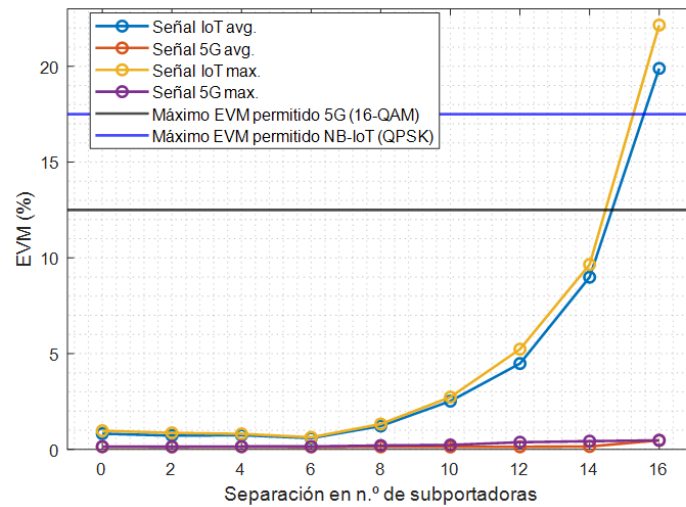


Figura B.4: EVM (%) en el escenario 6 en el caso en que no se modelan los efectos del amplificador.

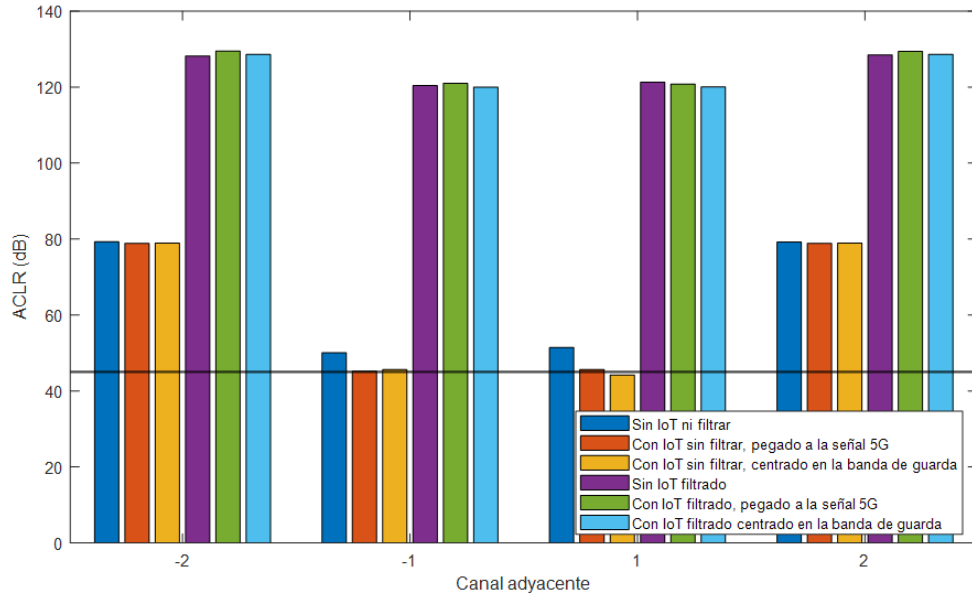


Figura B.5: ACLR en el escenario 6 en el caso en que no se modelan los efectos del amplificador.

Separación (en subportadoras)	EVM avg. (%)		EVM max. (%)	
	Señal 5G	Señal IoT	Señal 5G	Señal IoT
0	0,521	1,409	0,723	1,717
2	0,535	1,338	0,681	1,523
4	0,344	1,124	0,547	1,421
6	0,401	0,952	0,542	1,325
8	0,581	1,483	0,676	1,565
10	0,547	2,953	0,673	3,213
12	0,487	5,067	0,589	5,213
14	0,398	9,434	0,602	10,547
16	0,449	22,721	0,734	26,432

Tabla B.21: EVM (%) en el escenario 6 en el caso en que se utiliza un modelo de amplificador con memoria 1.

Situación	ACLR (dB)			
	Canal -2	Canal -1	Canal 1	Canal 2
Sin IoT, sin filtrar	67,26	50,10	49,94	67,21
Con IoT pegada, sin filtrar	65,27	44,10	43,51	64,98
Con IoT centrada, sin filtrar	64,81	41,90	42,72	64,81
Sin IoT, filtrado	63,91	55,43	55,33	63,70
Con IoT pegada, filtrado	62,95	55,00	55,10	63,03
Con IoT centrada, filtrado	63,03	52,90	53,62	63,23

Tabla B.22: ACLR en el escenario 6 en el caso en que se usa un modelo de amplificador con memoria de 1.

B.7. Escenario 7: Evaluación de los efectos al cambiar de numerología

NR-FR1-TM3.1						
BW (MHz)	EVM avg. (%)			EVM max. (%)		
	$\mu = 0$	$\mu = 1$	$\mu = 2$	$\mu = 0$	$\mu = 1$	$\mu = 2$
10	0,097	0,093	0,103	0,471	0,323	0,546
15	0,130	0,106	0,112	1,907	0,985	0,794
20	0,098	0,100	0,093	0,430	0,864	0,341
25	0,160	0,098	0,100	1,747	0,644	0,527
30	0,122	0,109	0,108	1,673	1,032	1,139
40	0,113	0,106	0,104	1,125	1,164	1,492

Tabla B.23: EVM(%) al cambiar numerología y ancho de banda usando el modelo de test NR-FR1-TM3.1, no se modelan los efectos del amplificador.

NR-FR1-TM3.2						
BW (MHz)	EVM avg. (%)			EVM max. (%)		
	$\mu = 0$	$\mu = 1$	$\mu = 2$	$\mu = 0$	$\mu = 1$	$\mu = 2$
10	0,167	0,154	0,179	0,803	0,960	0,746
15	0,266	0,162	0,206	5,300	2,382	0,952
20	0,152	0,190	0,156	0,731	3,763	1,167
25	0,203	0,152	0,259	1,681	2,401	1,701
30	0,187	0,246	0,155	1,059	3,255	2,383
40	0,159	0,153	0,171	1,445	0,908	3,765

Tabla B.24: EVM(%) al cambiar numerología y ancho de banda usando el modelo de test NR-FR1-TM3.2, no se modelan los efectos del amplificador.

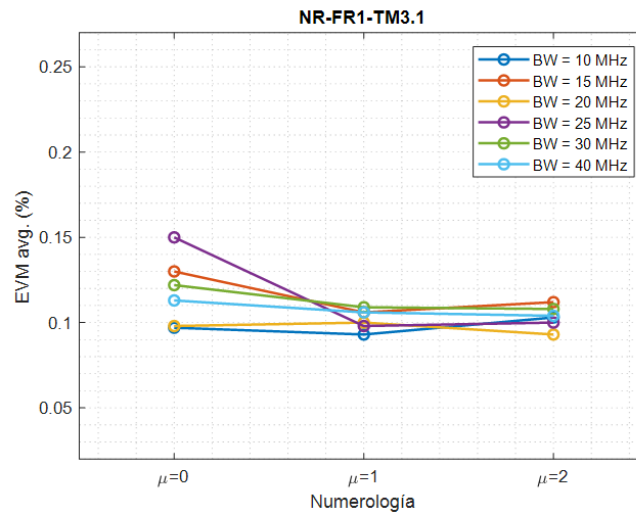


Figura B.6: EVM(%) promedio al cambiar numerología y ancho de banda usando el modelo de test NR-FR1-TM3.1, no se modelan los efectos del amplificador.

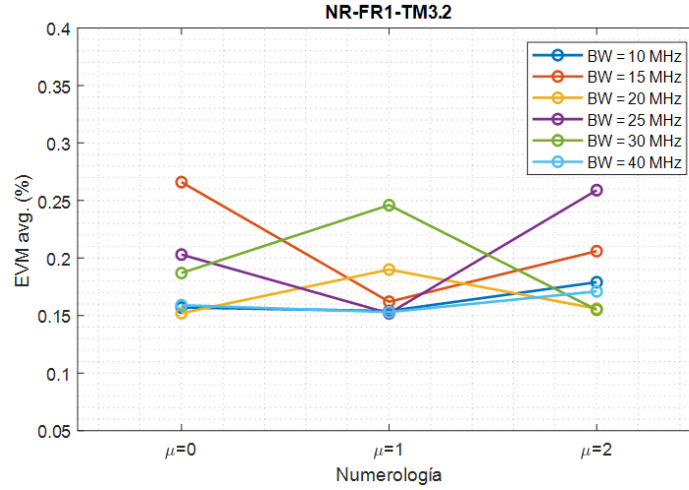


Figura B.7: EVM(%) promedio al cambiar numerología y ancho de banda usando el modelo de test NR-FR1-TM3.2, no se modelan los efectos del amplificador.

NR-FR1-TM3.1						
BW (MHz)	EVM avg. (%)			EVM max. (%)		
	$\mu = 0$	$\mu = 1$	$\mu = 2$	$\mu = 0$	$\mu = 1$	$\mu = 2$
10	0,108	0,118	0,113	0,464	0,564	0,541
15	0,136	0,118	0,123	1,932	0,955	0,809
20	0,110	0,113	0,117	0,432	0,768	0,564
25	0,152	0,114	0,108	1,597	0,682	0,515
30	0,127	0,113	0,120	1,647	0,623	1,003
40	0,121	0,117	0,115	1,237	1,073	1,416

Tabla B.25: EVM(%) al cambiar numerología y ancho de banda usando el modelo de test NR-FR1-TM3.1 y un modelado de amplificador con memoria de 1.

NR-FR1-TM3.2						
BW (MHz)	EVM avg. (%)			EVM max. (%)		
	$\mu = 0$	$\mu = 1$	$\mu = 2$	$\mu = 0$	$\mu = 1$	$\mu = 2$
10	0,175	0,178	0,192	0,962	1,330	0,795
15	0,264	0,181	0,233	5,034	2,297	1,108
20	0,159	0,192	0,178	0,898	3,219	1,330
25	0,204	0,166	0,265	1,363	2,005	1,652
30	0,190	0,261	0,177	0,840	3,049	2,300
40	0,168	0,160	0,179	1,371	0,898	3,221

Tabla B.26: EVM(%) al cambiar numerología y ancho de banda usando el modelo de test NR-FR1-TM3.2 y un modelado de amplificador con memoria de 1.

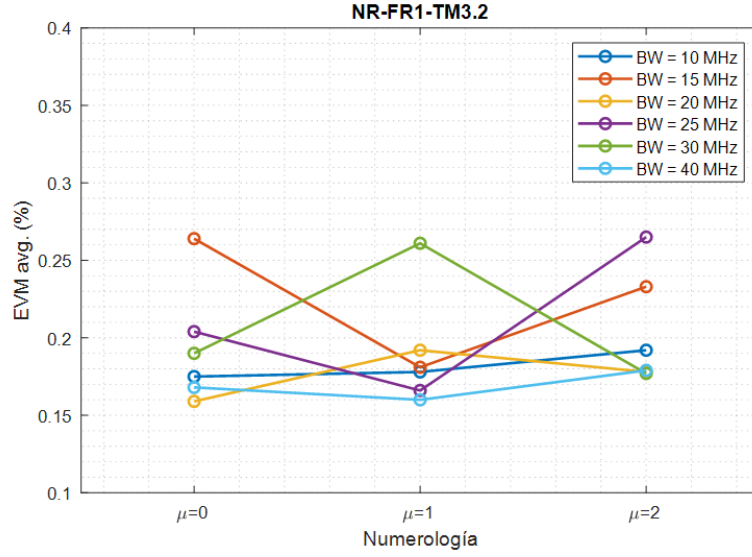


Figura B.8: EVM(%) al cambiar numerología y ancho de banda usando el modelo de test NR-FR1-TM3.2 y un modelado de amplificador con memoria de 1.

NR-FR1-TM3.1; ACLR (dB)												
BW (MHz)	Ch. -2			Ch. -1			Ch. 1			Ch. 2		
	$\mu = 0$	$\mu = 1$	$\mu = 2$	$\mu = 0$	$\mu = 1$	$\mu = 2$	$\mu = 0$	$\mu = 1$	$\mu = 2$	$\mu = 0$	$\mu = 1$	$\mu = 2$
10	68,31	68,82	70,31	48,62	48,50	51,22	47,41	51,71	51,11	68,91	68,57	71,19
15	66,52	65,25	69,22	46,11	46,42	51,11	46,03	47,47	50,42	67,02	65,29	69,89
20	67,42	68,47	68,83	46,15	45,40	48,43	50,89	48,41	52,68	67,68	69,31	68,72
25	65,44	68,71	68,61	48,09	46,91	49,90	48,41	51,67	53,78	65,48	69,31	69,33
30	64,31	66,31	65,29	48,42	45,09	45,23	46,22	49,69	45,89	64,38	66,78	65,49
40	67,89	67,38	68,47	48,77	46,20	45,34	47,19	46,71	48,41	68,41	67,71	69,29

Tabla B.27: ACLR al cambiar numerología y ancho de banda usando el modelo de test NR-FR1-TM3.1 y un modelado de amplificador con memoria de 1.

NR-FR1-TM3.2; ACLR (dB)												
BW (MHz)	Ch. -2			Ch. -1			Ch. 1			Ch. 2		
	$\mu = 0$	$\mu = 1$	$\mu = 2$	$\mu = 0$	$\mu = 1$	$\mu = 2$	$\mu = 0$	$\mu = 1$	$\mu = 2$	$\mu = 0$	$\mu = 1$	$\mu = 2$
10	66,71	69,43	69,82	45,88	50,78	51,21	49,19	54,27	54,21	67,31	70,48	70,56
15	66,10	65,32	69,21	44,81	46,32	50,84	44,91	46,32	52,52	66,44	66,08	70,01
20	64,09	65,21	69,43	46,02	49,27	50,71	47,62	45,19	51,31	64,77	64,69	70,48
25	66,62	68,22	69,10	48,62	49,78	52,22	46,55	50,57	51,64	67,21	68,78	69,12
30	63,08	66,45	65,07	47,13	44,69	45,07	48,28	44,82	46,12	63,31	66,58	65,90
40	64,78	64,19	65,19	48,71	46,72	49,09	48,89	47,18	45,11	65,32	64,91	64,68

Tabla B.28: ACLR al cambiar numerología y ancho de banda usando el modelo de test NR-FR1-TM3.2 y un modelado de amplificador con memoria de 1.

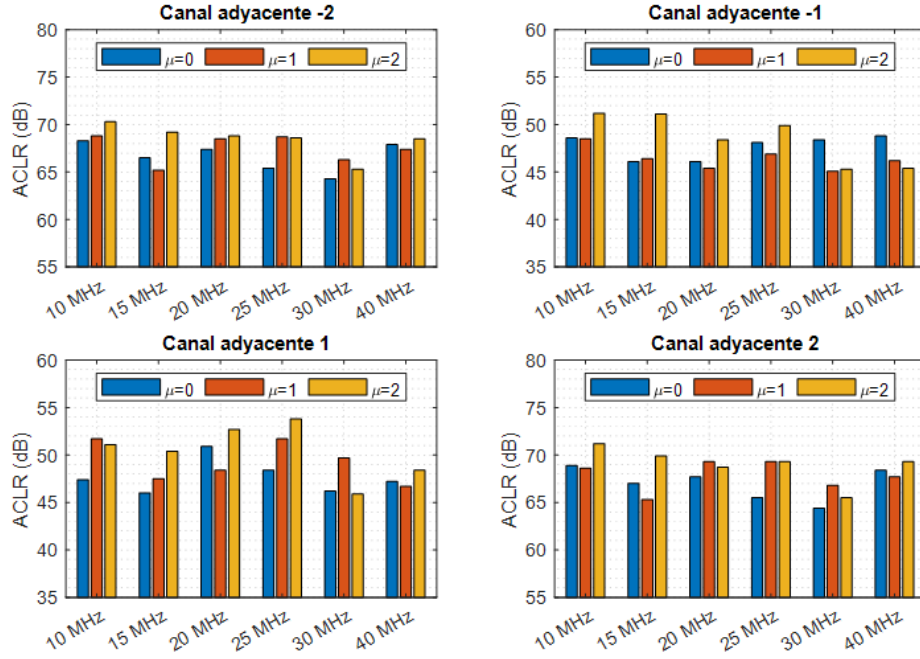


Figura B.9: ACLR al cambiar numerología y ancho de banda usando el modelo de test NR-FR1-TM3.2 y un modelado de amplificador con memoria de 1.

B.8. Escenario 8: Simulación de modelos de canal

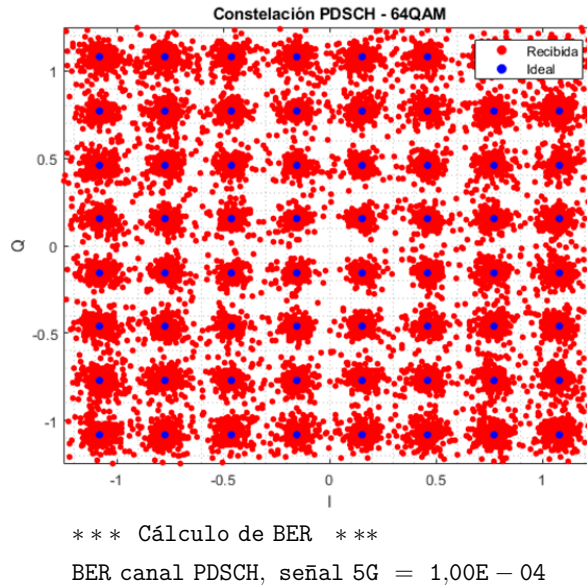


Figura B.10: Constelación recibida para una transmisión a través de un modelo de canal ETU con frecuencia Doppler máxima de 70 Hz y utilizando modulación 64-QAM (arriba). Resultados de BER obtenidos para dicha transmisión (abajo).

Anexo C

Códigos de MATLAB®

En el presente anexo, C, se recoge una parte del código desarrollado en el entorno MATLAB® para la obtención de los resultados recogidos en este TFG. Este código desarrollado para simulador constituye, además, un segundo resultado fundamental del TFG, ya que puede ser especialmente útil para trabajos futuros.

Por un lado, se recogen primero las funciones auxiliares llamadas por los simuladores. Son las siguientes:

1. calcPRB.m
2. calPAMPM.m
3. calPAPM.m
4. correctoNBloT.m
5. corrPow.m
6. corrPow2.m
7. generateNBloT.m
8. matrix_data_pseudoM.m
9. trainKDriver.m
10. trainPM.m
11. trainMPM.m
12. trainPDPM.m
13. trainPDMPM.m

Y, por otro, una estructura del código de los dos simuladores. No se incluye en este caso la totalidad del código porque alargaría excesiva e innecesariamente la memoria.

13. DLSimulator.m

14. DL_5G NR_Simulator.m

Además, se utilizan una serie de funciones de MATLAB® ligeramente modificadas para adaptarlas a nuestros supuestos. Estas pueden ser consultadas junto con el código completo.

El código utilizado es el siguiente:

1. Función: calcPRB.m

```

1  % Hace el parsing de la cadena introducida por el usuario indicando los
2  % PRBs a utilizar y comprueba que los valores introducidos son correctos
3  %
4  % [PRB,correcto] = calcPRB(input,nRB,link,prbsIot)
5  %   Parámetros de ENTRADA:
6  %       - input: Cadena introducida por el usuario
7  %       - nRB: Número de bloques de recursos para los que se configura en enlace
8  %       - link: Indica si se trata de enlace ascendente 'UL' o descendente 'DL'
9  %       - prbsIot: Indica los PRBs asignados para transmitir señales NB-Iot y que
10 %                   no podrán rellenarse
11 %
12 %   Parámetros de Salida
13 %       - PRB: String con los PRBs que se van a usar
14 %       - correcto: Indica si el resultado de la asignación es correcto o no
15 %
16
17 function [PRB,correcto] = calcPRB(varargin)
18
19     % Obtener algunos parámetros de entrada
20     a = cell2mat(varargin(2));
21     b = nargin;
22     PRB = [];
23     correcto = true;
24     cad = char(strsplit(char(varargin(1))','));
25     nprb = cell2mat(varargin(2));
26     [n,m] = size(cad);
27
28     if(strcmp(char(varargin(3)),'DL')) % Calculo para el DL
29         % Obtenemos PRBs usados por señales Iot
30         prbIot = cell2mat(varargin(4));
31
32         % Realizamos el parsing de la cadena introducida por el usuario
33         % para obtener los numeros de PRB utilizados
34         for j=1:n
35             if(strfind(strrep(cad(j,:),','),'-'))
36                 aux = strsplit(char(cad(j,)),'-');
37                 PRB = [PRB str2num(char(aux(1))):str2num(char(aux(2)))];
38             else
39                 PRB = [PRB str2num(char(cad(j,)))];
40             end
41         end
42
43         % Comprobamos que los PRBs son correctos
44         if(~isempty(PRB))
45             if(max(PRB)>=nprbmin(PRB)<0)
46                 correcto = false;
47             end
48             if(~isempty(prbIot))
49                 for ii=1:length(prbIot)
50                     if(PRB(PRB==prbIot(ii)))
51                         correcto = false;
52                         break
53                     end
54                 end
55             end
56         end

```

```

57 elseif(strcmp(char(varargin(3)),'UL'))
58     for j=1:n
59         slotsLim = cell2struct(varargin(4),'aux');
60         slotsLim = slotsLim.aux;
61         iniSlot1 = slotsLim.ini1
62         finSlot1 = slotsLim.fin1
63         iniSlot2 = slotsLim.ini2
64         finSlot2 = slotsLim.fin2
65
66         if(strfind(strrep(cad(j,:),','),'-'))
67             aux = strsplit(char(cad(j,:)),'-');
68             aux2 = [iniSlot1+(str2num(char(aux(1)))) : iniSlot1+str2num(char(aux(2))) ;
69                 iniSlot2+(str2num(char(aux(1)))) : iniSlot2+str2num(char(aux(2)))];
70             PRB = [PRB aux2];
71         else
72             aux = [iniSlot1 + str2num(char(cad(j,:))) ;
73                 iniSlot2 + str2num(char(cad(j,:)))];
74             PRB = [PRB aux];
75         end
76     end
77
78     if(~isempty(PRB))
79         if( (max(PRB(1,:)) > (nprb-finSlot1)) (max(PRB(2,:)) > (nprb-finSlot2)) ...
80             (min(PRB(1,:)) < iniSlot1) (min(PRB(1,:)) > iniSlot2) )
81             correcto = false;
82         end
83     end
84     PRB = PRB';
85 end

```

2. Función: calPAMPM.m

```

1  % Calcula la salida de aplicar un modelo polinomial con memoria
2  %
3  % [y_est] = calPAMPM(buffin,memoria,a)
4  %   Parámetros de ENTRADA:
5  %       - buffin: señal de entrada
6  %       - memoria: memoria del modelo (1 es sin memoria)
7  %       - a: coeficientes del modelo
8  %   Parámetros de SALIDA:
9  %       - y_est: señal tras aplicar el modelo
10
11 function y_est = calPAMPM(buffin,memoria,a)
12     % Calculamos el orden del modelo
13     orden = length(a)/memoria;
14
15     % Ponemos la señal de entrada en formato matricial, segun lo
16     % especificado en la teoria
17     espacio = matrix_data_pseudoM_v(buffin,orden,memoria);
18
19     % Calculamos la señal de salida
20     y_est = espacio*a;
21
22 end

```

3. Función: calPAPM.m

```

1  % Calcula la salida de aplicar un modelo polinomial sin memoria
2  %
3  % [y_est] = calPAPM(sigin,a)
4  %   Parámetros de ENTRADA:
5  %       - sigin: señal de entrada
6  %       - a: coeficientes del modelo
7  %   Parámetros de SALIDA:

```

```

8 % - y_est: señal tras aplicar el modelo
9
10 function sigout = calPAPM(sigin,a)
11 % Aplicamos ecuación del modelo
12 sigout = sigin.*polyval(flip(a),abs(sigin));
13 end

```

4. Función: correctoNBioT.m

```

1 % Hace el parsing de la cadena introducida por el usuario indicando la
2 % posición de los canales de NB-Iot y comprueba que es correcta.
3 %
4 % [correcto, pos] = correctoNBioT(type, input, NDLRB, prbsUsed,varargin)
5 % Parámetros de ENTRADA:
6 % - type: Indica si se está trabajando en modo en banda 'In-band' o en
7 % banda de guarda 'GuardBand'
8 % - input: Cadena introducida por el usuario indicando la posición de
9 % los canales de NB-Iot
10 % - NDLRB: Número de bloques de recursos para los que se ha configurado
11 % el enlace
12 % - prbsUsed: PRBs usados en la señal de banda ancha, los cuales no
13 % podrán ser asignados al canal de Iot
14 % - varargin: Permite discriminar si estamos en la banda de guarda
15 % superior o inferior en el caso de iot en banda de guarda
16 % Parámetros de SALIDA:
17 % - correcto: Indica si la cadena introducida por el usuario es o no
18 % correcta
19 % - pos: Recoge las subportadoras en las que se transmitiría el canal y
20 % los PRBs a los que corresponden en el caso de transmisión en
21 % banda.
22
23 function [correcto, pos] = correctoNBioT(type, input, NDLRB, prbsUsed,varargin)
24
25 % Definimos algunos parámetros
26 pos = [];
27 correcto = true;
28
29 % numero máximo de canales por banda de guarda y posicion (en subportadoras)
30 NB_IoT_GB = struct();
31 NB_IoT_GB.RB25 = struct('nChan', 1, 'LowGB', 100, 'HighGB', 412);
32 NB_IoT_GB.RB50 = struct('nChan', 2, 'LowGB', 206, 'HighGB', 818);
33 NB_IoT_GB.RB75 = struct('nChan', 4, 'LowGB', 568, 'HighGB', 1480);
34 NB_IoT_GB.RB100 = struct('nChan', 5, 'LowGB', 418, 'HighGB', 1630);
35
36 % limites de la banda (en subportadoras) y prbs utilizables para señales iot
37 NB_IoT_IB = struct();
38 NB_IoT_IB.RB15 = struct('limInf', 39,'prb', [2 12]);
39 NB_IoT_IB.RB25 = struct('limInf', 107,'prb', [2 7 17 22]);
40 NB_IoT_IB.RB50 = struct('limInf', 213,'prb', [4 9 14 19 30 35 40 45]);
41 NB_IoT_IB.RB75 = struct('limInf', 575,'prb', ...
42 [2 7 12 17 22 27 32 42 47 52 57 62 67 72]);
43 NB_IoT_IB.RB100 = struct('limInf', 425,'prb', ...
44 [4 9 14 19 24 29 34 39 44 55 60 65 70 75 80 85 90 95]);
45
46 if(strcmp(type,'GuardBand')) %Guardband
47 % Obtenemos la informacion necesaria
48 info = NB_IoT_GB.(strcat('RB',num2str(NDLRB)));
49
50 % Error si se quieren introducir más canales de los permitidos
51 if(input>info.nChan)
52 correcto = false;
53 end
54
55 % Calculamos la posicion (en subportadoras) que ocuparian los
56 % canales en banda de guarda
57 if(correcto)
58 if(~isempty(input))
59 if(~varargin{1})

```

```

60         % Si el argumento adicional es un 0 estamos en la banda de
61         % guarda inferior
62         pos.subcarriers = info.LowGB-12*(input-1): 12: info.LowGB;
63     else
64         % Si el argumento adicional es un 1 estamos en la banda de
65         % guarda superior
66         pos.subcarriers = info.HighGB : 12 : info.HighGB+12*(input-1);
67     end
68 end
69 end
70
71 else %Inband
72     correcto_aux = true;
73
74     % Obtenemos la informacion necesaria
75     info = NB_IoT_IB.(strcat('RB',num2str(NDLRB)));
76     prbs = info.prb;
77
78     % Ordenamos los prbs introducidos por el usuario
79     prbs_iot = unique(str2num(char(split(char((input))',''))));
80
81     % Comprobamos que son correctos, viendo que todos los prbs
82     % introducidos están entre los permitidos
83     for ii = 1:length(prbs_iot)
84         correcto_aux = correcto_aux && ismember(prbs_iot(ii),prbs);
85     end
86     correcto = correcto_aux;
87
88     % En caso de que lo anterior sea válido, comprobar que los prbs no
89     % están siendo utilizados para transmitir datos
90     if(correcto)
91         for i = 1:10
92             auxUsed = prbsUsed.(['st' num2str(i)]);
93             contain = ismember(prbs_iot, auxUsed);
94             if(contain)
95                 correcto = false;
96                 break
97             end
98         end
99     end
100
101     % Si se han pasado todas las comprobaciones guardar las posiciones,
102     % tanto en prb como en subportadoras
103     if(correcto)
104         pos.subcarriers = info.limInf + 5 + prbs_iot*12;
105         pos.prb = prbs_iot;
106     end
107
108 end
109
110 end

```

5. Función: corrPow.m

```

1 % Corrige en recepción las diferencias de potencia entre prbs y pilotos introducidas
2 % en transmisión para evaluar el comportamiento del amplificador.
3 % Función adaptada para su llamada directamente desde la demodulación hecha
4 % en el simulador. (Recibe como entrada la señal completa)
5 %
6 % [rxGridLTE]=corrPow(enb, rxGridLTE,powerFactor_pilots,powerFactor_PRBs,indices,nTramas)
7 % Parámetros de ENTRADA:
8 %     - enb: Parametros del enodeB.
9 %     - rxGridLTE: grid de la señal de banda ancha recibido.
10 %     - powerFactor_pilots: potencias relativas aplicadas a los pilotos
11 %     - powerFactor_PRBs: potencias relativas aplicadas a los PRBs
12 %     - indices: indices de cada uno de los canales que forman la señal
13 %     - nTramas: número de tramas de las que consta la señal
14 % Parámetros de SALIDA:

```

```

15 % - rxGridLTE: grid de la señal de banda ancha con las diferencias de
16 % potencia introducidas en transmisión corregidas.
17
18
19 function [rxGridLTEcorr] = corrPow(enb, rxGridLTE,powerFactor_pilots, ...
20                                     powerFactor_PRBs,indices,nTramas)
21     % Ver si tenemos prefijo normal o extendido
22     u = 7*2;
23     if(strcmp(enb.CyclicPrefix,'Extended'))
24         u = 6*2;
25     end
26
27     % Calcular tamaños (total y por subtrama)
28     [a,b] = size(rxGridLTE);
29     tamMax = a*b;
30     tamSf = a*u;
31
32     corr = 1;
33     rxGridLTEcorr = [];
34     % Recorrer cada trama y subtrama y realizar la corrección
35     for ii =1:nTramas
36         for jj=0:9
37
38             fin = corr+u-1;
39             if(fin<b)
40                 sfGrid = rxGridLTE(:,corr:fin);
41             else
42                 sfGrid = rxGridLTE(:,corr:end);
43             end
44             nSubframe_str = strcat('Subframe',num2str(jj));
45             indsAnt0 = indices.(nSubframe_str).ant0;
46             indsAnt0(indsAnt0>fin*a) = [];
47
48             % Obtenemos los índices de cada canal en cada subtrama
49             [mm,nn] = size(sfGrid);
50             powF = powerFactor_PRBs.(strcat('st',num2str(jj+1)));
51             powF = repmat(powF,12,1);
52             powF = reshape(powF,[mm,1]);
53             powF = repmat(powF,1,nn);
54             powF(indsAnt0) = powerFactor_pilots.(['st' num2str(jj+1)]);
55
56             rxGridLTEcorr = [rxGridLTEcorr sfGrid./powF];
57
58             % Añadir el offset del tamaño de una subtrama para actualizar
59             % los índices. (Los índices recogidos en la estructura indices
60             % no son absolutos, si no relativos para cada subtrama).
61             corr = corr + u;
62
63         end
64     end
65 end

```

6. Función: corrPow2.m

```

1 % Corrige en recepción las diferencias de potencia entre prbs y pilotos introducidas
2 % en transmisión para evaluar el comportamiento del amplificador.
3 % Función adaptada para su llamada desde la función de matlab que demodula y calcula
4 % las medidas de EVM. (Recibe como entrada sólo una subtrama)
5 %
6 % [rxGridLTE] = corrPow(enb, rxGridLTE,powerFactor_pilots,powerFactor_PRBs,indices,nTramas)
7 % Parámetros de ENTRADA:
8 % - rxGridLTE: grid de la señal de banda ancha recibido.
9 % - powerFactor_pilots: potencias relativas aplicadas a los pilotos
10 % - powerFactor_PRBs: potencias relativas aplicadas a los PRBs
11 % - indices: índices de cada uno de los canales que forman la señal
12 % - jj: número de subtrama en la que nos encontramos
13 % Parámetros de SALIDA:
14 % - rxGridLTE: grid de una subtrama la señal de banda ancha con las diferencias de

```

```

15 % potencia introducidas en transmisión corregidas.
16
17
18 function [rxGridLTEcorr] = corrPow2(rxGridLTE,powerFactor_pilots,powerFactor_PRBs,indices,jj)
19 % Calcular tamaño de la matriz-subtrama
20 [a,b] = size(rxGridLTE);
21
22 % Obtenemos los índices de cada canal en la subtrama
23 nSubframe_str = strcat('Subframe',num2str(jj));
24 indsAnt0 = indices.(nSubframe_str).ant0;
25
26 % Obtenemos los índices de cada canal en cada subtrama
27 powF = powerFactor_PRBs.(strcat('st',num2str(jj+1)));
28 powF = repmat(powF,12,1);
29 powF = reshape(powF,[a,1]);
30 powF = repmat(powF,1,b);
31 powF(indsAnt0) = powerFactor_pilots.(['st' num2str(jj+1)]);
32
33 rxGridLTEcorr = rxGridLTE./powF;
34 end

```

7. Función: generateNBIoT.m

```

1 % A partir de la información de posición, número de tramas y configuración
2 % del enlace, genera las formas de onda de los canales NB-IoT
3 %
4 [txWaveform, grid, gridMask, info, ngend] = generateNBIoT (position, pos,nFrames, ...
5 % RB, offsetInf, offsetSup)
6 % Parámetros de ENTRADA:
7 % - position: Modo de operación en banda o en banda de guarda
8 % - pos: Indica las subportadoras en las que se sitúa el canal de
9 % NB-IoT, y los PRBs en el caso de modo de operación en banda.
10 % - nFrames: Número de tramas a generar
11 % - RB: Número de bloques de recursos para los que está configurado el
12 % enlace
13 % - offsetInf: Permite controlar la posición de los canales de la
14 % banda de guarda inferior mediante un offset.
15 % - offsetSup: Permite controlar la posición de los canales de la
16 % banda de guarda superior mediante un offset.
17 % Parámetros de SALIDA:
18 % - txWaveform: Forma de onda correspondiente al canal generado
19 % - grid: Matriz de recursos de dos subtramas del canal
20 % - gridMask: Máscara que indica las posiciones del canal dentro de la
21 % matriz global.
22 % - info: Estructura con información acerca de la modulación
23 % - ngend: Objeto que recoge toda la información de los generadores de
24 % los canales NB-IoT
25 %
26 %
27 % NOTA: Se podría configurar cómo se llena la subtrama de NB-Iot, pero por
28 % simplicidad, se ha elegido una configuración fija, ya que la influencia
29 % que tendría en los resultados sería mínima.
30 %
31
32 function [txWaveform,grid,gridMask,info,ngend] = generateNBIoT(position, pos, nFrames, ...
33 % RB, offsetInf, offsetSup)
34
35 % Definimos tamaños de fft
36 NB_IoTnFFT = struct('RB15', 256, 'RB25', 512, 'RB50', 1024, 'RB75', 2048, ...
37 % 'RB100', 2048);
38
39 % Las tramas de iot se generan de dos en dos, si el numero de tramas a
40 % generar es impar, generar una más
41 nFramesAux = nFrames;
42 if (rem(nFrames,2)==1)
43 nFramesAux = nFrames+1;
44 end
45

```

```

46 % Definimos el grid y el objeto para generar la señal
47 grid = zeros(NB_IoTnFFT.(['RB' num2str(RB)]),nFramesAux*140);
48 gridMask = grid;
49 ngen = NBIoTDownlinkWaveformGenerator_V;
50
51 % Definimos el tipo de canal, si es in-band elegimos una 'Anchor' y el
52 % resto 'NonAnchor', si es guardband, todas 'NonAnchor'
53 if(strcmp(position,'GuardBand'))
54     ngen.Config.CarrierType = 'NonAnchor';
55 else
56     ngen.Config.CarrierType = 'Anchor';
57 end
58
59 % Definir todos los parámetros del objeto generador
60 ngen.Config.OperationMode = position;
61 ngen.Config.DownlinkBitmap = [1 1 1 1 1 1 1 1]; %Subframes used
62 ngen.Config.TotSubframes = nFramesAux*10;
63
64 % Configure the parameters of the first NPDCCH
65 npdcch1.Enable = 'On';
66 npdcch1.Power = 0;
67 npdcch1.NCCE = 1; % NPDCCH format 0 with NCCE 1
68 npdcch1.NRep = 6;
69 npdcch1.Rmax = 16;
70 npdcch1.RNTI = 0;
71 npdcch1.StartSubframe = 0;
72 npdcch1.DataBlkSize = 23;
73 npdcch1.DataSource = randi([0 1],npdcch1.DataBlkSize,1);
74
75 % Configure the parameters of the second NPDCCH
76 npdcch2.Enable = 'On';
77 npdcch2.Power = 0;
78 npdcch2.NCCE = 0; % NPDCCH format 0 with NCCE 0
79 npdcch2.NRep = 6;
80 npdcch2.Rmax = 16;
81 npdcch2.RNTI = 1;
82 npdcch2.StartSubframe = 0;
83 npdcch2.DataBlkSize = 23;
84 npdcch2.DataSource = randi([0 1],npdcch2.DataBlkSize,1);
85
86 % Configure the parameters of the first NPDSCH
87 npdsch1.Enable = 'On';
88 npdsch1.Power = 0;
89 npdsch1.NPDSCHDataType = 'BCCHNotSIB1NB';
90 npdsch1.NSF = 2;
91 npdsch1.NRep = 4;
92 npdsch1.Rmax = npdcch1.Rmax;
93 npdsch1.RNTI = 0;
94 npdsch1.StartSubframe = 10;
95 npdsch1.DataBlkSize = 616;
96 npdsch1.DataSource = randi([0 1],npdsch1.DataBlkSize,1);
97
98 % Configure the parameters of the second NPDSCH
99 npdsch2.Enable = 'On';
100 npdsch2.Power = 0;
101 npdsch2.NPDSCHDataType = 'NotBCCH';
102 npdsch2.NSF = 2;
103 npdsch2.NRep = 7;
104 npdsch2.Rmax = npdcch2.Rmax;
105 npdsch2.RNTI = 1;
106 npdsch2.StartSubframe = 20;
107 npdsch2.DataBlkSize = 616;
108 npdsch2.DataSource = randi([0 1],npdsch2.DataBlkSize,1);
109
110 for i=1:length(pos.subcarriers)
111     if(strcmp(ngen.Config.CarrierType,'Anchor'))
112         if(strcmp(ngen.Config.OperationMode,'Guardband'))
113             npdcch1.NRep = 6;
114             npdcch2.NRep = 6;
115             npdsch1.NRep = 4;
116             npdsch2.NRep = 6;

```

```

117         else
118             npdcch1.NRep = 6;
119             npdcch1.NRep = 6;
120             npdsch1.NRep = 4;
121             npdsch2.NRep = 3;
122         end
123     else
124         if(strcmp(ngen.Config.OperationMode,'Guardband'))
125             npdcch1.NRep = 10;
126             npdcch2.NRep = 10;
127             npdsch1.NRep = 5;
128             npdsch2.NRep = 8;
129         else
130             npdcch1.NRep = 10;
131             npdcch2.NRep = 10;
132             npdsch1.NRep = 5;
133             npdsch2.NRep = 7;
134         end
135     end
136 end
137
138
139 % Prepare NPDSCH and NPDCCCH structure vectors
140 ngen.Config.NPDCCCH = [npdcch1 npdcch2];
141 ngen.Config.NPDSCH = [npdsch1 npdsch2];
142
143 % Generar grid
144 [grid_aux,sftypes] = ngen.generateGrid;
145
146 % Ver si tenemos que aplicar el offset superior o el inferior
147 % dependiendo de en qué banda nos encontramos
148 if(pos.subcarriers(i)>size(grid_aux)/2)
149     offset = offsetSup;
150 else
151     offset = offsetInf;
152 end
153
154 % Colocar los grid generados en el lugar que corresponde en el grid
155 % global
156 grid(pos.subcarriers(i)-6+1+offset:pos.subcarriers(i)+6+offset,:) = grid_aux;
157 gridMask(pos.subcarriers(i)-6+1+offset:pos.subcarriers(i)+6+offset,:) = ...
158     ones(size(grid_aux));
159
160
161 % Guardar el objeto generador y las posiciones en la estructura
162 % ngend.
163 if(strcmp(position,'Inband-SamePCI'))
164     if(strcmp(ngen.Config.CarrierType,'Anchor'))
165         ngend.Anchor = ngen;
166         ngend.Anchor.Config.prb = pos.prb(1);
167     else
168         ngend.NonAnchor = ngen;
169         ngend.NonAnchor.Config.prb = pos.prb(2:end);
170     end
171 else
172     ngend.NonAnchor = ngen;
173 end
174
175
176 % Cambiar los canales que no son el primero a NonAnchor
177 ngen.Config.CarrierType = 'NonAnchor';
178 end
179
180 % Si el número de tramas a generar era impar, eliminar la última
181 if(rem(nFrames,2)==1)
182     grid = grid(:,1:end-140);
183     gridMask = gridMask(:,1:end-140);
184 end
185
186 % Modular y generar la señal en tiempo
187 enb = ngen.Config;

```

```

188 enb.CyclicPrefixUL = 'Normal';
189
190 if(~isfield(enb,'Windowing'))
191     enb.Windowing = 6;
192 end
193
194 if(strcmp(position,'Inband-SamePCI'))
195     aux = (NB_IoTnFFT.(['RB' num2str(RB)])- RB*12)/2+1;
196     grid = grid(aux:aux+RB*12-1,:);
197     gridMask = gridMask(aux:aux+RB*12-1,:);
198
199 end
200
201 [txWaveform,info] = lteOFDMModulate(enb,grid,6);
202 info.SubframeChannelTypes = sftypes;
203 end

```

8. Función: matrix_data_pseudoM.m

```

1
2 % Permite calcular la matriz necesaria para aplicar el modelo polinomial
3 % con memoria a partir de la señal de entrada y los parámetros del modelo.
4 %
5 % [ U ] = matrix_data_pseudoM_v(sigin,orden,memoria)
6 %   Parámetros de ENTRADA:
7 %       - sigin: señal de entrada
8 %       - orden: orden del modelo
9 %       - memoria: memoria del modelo
10 %   Parámetros de SALIDA:
11 %       - U: Señal de entrada en formato matricial.
12 %
13
14 function [ U ] = matrix_data_pseudoM(sigin,orden,memoria)
15     % Definimos los parametros
16     M = memoria;
17     N = orden;
18     long = length(sigin);
19     U = zeros(length(sigin),M*N); % Señal de entrada en forma matricial
20     aux = zeros(long,M); % Submatriz con los valores de x(n-m)
21
22     % Hacemos un padding al inicio de la señal de entrada para que poder
23     % tener memoria en las primeras muestras
24     auxsig = [zeros(M,1); sigin];
25
26     % Rellenamos los valores de x(n-m)
27     for ii=0:M-1
28         aux(:,ii+1) = auxsig(M-ii+1:M-ii+long);
29     end
30
31     % Calculamos los valores Xk y los ponemos en la matriz U
32     for ii=1:N
33         U(:,M*(ii-1)+1:M*(ii-1)+M) = aux.*abs(aux).^(ii-1);
34     end
35
36 end

```

9. Función: trainKDriver.m

```

1 % Calcula el factor de ganancia k del driver para poner al amplificador en
2 % el punto de trabajo que queremos
3 %
4 % [k] = trainKDriver (sigin)
5 %   Parámetros de ENTRADA:
6 %       - sigin: Señal de entrada de referencia al sistema no lineal que se
7 %       quiere modelar
8 %

```

```

9 % Parámetros de SALIDA:
10 % - k: factor de ganancia a aplicar a la señal generada si se ocupara
11 % todo el ancho de banda del sistema para que sea de la misma
12 % potencia que la señal de referencia
13 %
14
15 function [k] = trainKDriver(sigin)
16 % Definimos parámetros
17 k = struct('RB6',[],'RB15',[],'RB25',[],'RB50',[],'RB75',[],'RB100',[]);
18 RB = [6 15 25 50 75 100];
19 rc = {'R.4' 'R.5' 'R.6' 'R.7' 'R.8' 'R.9'};
20 Pin = potenciade(reshape(sigin,1,length(sigin)));
21
22 % Para cada ancho de banda, calculamos el factor k
23 for i=1:6
24     rmc = lteRMCDL(rc{i});
25     [stx,~,~]=lteRMCDLTool(rmc,[1;0;0;1]);
26     [~,ki] = ajustapotencia(reshape(stx,1,length(stx)),Pin);
27
28     k.(['RB' num2str(RB(i))]) = ki;
29
30 end
31 end

```

10. Función: trainPM.m

```

1 % Calcula el modelo del amplificador usando un modelo polinomial con
2 % memoria.
3 %
4 % [a,error,y_est] = trainMPM (sigin,sigout,orden,memoria,vis)
5 % Parámetros de ENTRADA:
6 % - sigin: Señal de entrada de entrenamiento
7 % - sigout: Señal de salida de entrenamiento
8 % - orden: Orden del modelo
9 % - memoria: Memoria del modelo (1 es sin memoria)
10 % - vis: Permite visualizar el resultado (1) o no (0)
11 % Parámetros de SALIDA:
12 % - a: Coeficientes del modelo
13 % - error: Error cometido en la estimación de la señal de salida
14 % - y_est: Señal de salida estimada con el modelo calculado
15
16 function [a,error,y_est] = trainMPM(sigin,sigout,orden,memoria,vis)
17 % Ponemos la señal de entrada en formato matricial
18 espacio = matrix_data_pseudoM_v(sigin,orden,memoria);
19
20 % Calculamos los coeficientes del modelo
21 a = pinv(espacio)*sigout;
22 y_est = espacio*a;
23
24 % Visualización del resultado del modelado
25 if(vis)
26     figure, scatter(abs(sigin),abs(sigout),'.');
27     hold on, scatter(abs(sigin),abs(y_est),'k','.');
28 end
29
30 % Calculamos el error en la estimación
31 error = 10*log10(mean(abs(sigout-y_est).^2)/mean(abs(sigout).^2));
32
33 end

```

11. Función: trainMPM.m

```

1 % Calcula el modelo predistorsionador del modelo polinomial con memoria.
2 %

```

```

3  % [b,error2,y_DPD,Gcas,u,v] = trainPDMPM (sigin, sigout, orden, memoria, FactorG, vis)
4  %   Parámetros de ENTRADA:
5  %       - sigin: Señal de entrada de entrenamiento
6  %       - sigout: Señal de salida de entrenamiento
7  %       - orden: Orden del modelo
8  %       - memoria: Memoria del modelo (1 es sin memoria)
9  %       - FactorG: Factor de ganancia
10 %       - vis: Permite visualizar el resultado del modelado (1) o no (0)
11 %   Parámetros de SALIDA:
12 %       - b: Coeficientes del modelo
13 %       - error2: Error cometido
14 %       - y_DPD: Señal de salida predistorsionada estimada
15 %       - Gcas: Ganancia obtenida
16 %       - u: Señal de entrada
17 %       - v: Señal de salida normalizada por la ganancia
18
19 function [b,error2,y_DPD,Gcas,u,v] = trainPDMPM(sigin,sigout,orden,memoria,FactorG,vis)
20 % Estimamos la ganancia promedio
21 Glina=mean(abs(sigout./sigin));
22 Glinp=mean(angle(sigout./sigin));
23 Glin=Glina*exp(j*Glinp);
24 Gcas=FactorG*Glin;
25
26 fprintf('Ganancia sin linealización: %0.2f dB \n', 20*log10(Glina));
27 fprintf('Ganancia con linealización: %0.2f dB \n', 20*log10(Gcas));
28 fprintf('Diferencia de ganancia: %0.2f dB \n', 20*log10(Gcas/Glina));
29
30 % Cambiamos las señales y calculamos el modelo
31 v=sigin;
32 u=sigout/Gcas;
33 [b,error2,y_DPD]=trainMPM_v(u,v,orden,memoria,vis);
34
35 end

```

12. Función: trainPDPM.m

```

1  % Calcula el modelo predistorsionador del modelo polinomial sin memoria.
2  %
3  % [b,error2,y_DPD,Gcas,u,v] = trainPDPM(sigin, sigout, orden, FactorG, vis)
4  %   Parámetros de ENTRADA:
5  %       - sigin: Señal de entrada de entrenamiento
6  %       - sigout: Señal de salida de entrenamiento
7  %       - orden: Orden del modelo
8  %       - FactorG: Factor de ganancia
9  %       - vis: Permite visualizar el resultado del modelado (1) o no (0)
10 %   Parámetros de SALIDA:
11 %       - b: Coeficientes del modelo
12 %       - error2: Error cometido
13 %       - y_DPD: Señal de salida predistorsionada estimada
14 %       - Gcas: Ganancia obtenida
15 %       - u: Señal de entrada
16 %       - v: Señal de salida normalizada por la ganancia
17
18 function [b,error2,y_DPD,Gcas,u,v] = trainPDPM( sigin,sigout, orden,FactorG, vis )
19 % Estimamos la ganancia promedio
20 Glina=mean(abs(sigout./sigin));
21 Glinp=mean(angle(sigout./sigin));
22 Glin=Glina*exp(j*Glinp);
23 Gcas=FactorG*Glin;
24
25 fprintf('Ganancia sin linealización: %0.2f dB \n', 20*log10(Glina));
26 fprintf('Ganancia con linealización: %0.2f dB \n', 20*log10(Gcas));
27 fprintf('Diferencia de ganancia: %0.2f dB \n', 20*log10(Gcas/Glina));
28
29 % Cambiamos las señales y calculamos el modelo
30 v=sigin;
31 u=sigout/Gcas;
32 [b,error2,y_DPD]=trainPM_v(u,v,orden,vis);

```

```

33
34 end

```

13. Función: trainPDMPM.m

```

1  % Calcula el modelo del amplificador usando un modelo polinomial con
2  % memoria.
3  %
4  % [a,error,y_est] = trainPM (sigin,sigout,orden,memoria,vis)
5  %   Parámetros de ENTRADA:
6  %       - sigin: Señal de entrada de entrenamiento
7  %       - sigout: Señal de salida de entrenamiento
8  %       - orden: Orden del modelo
9  %       - vis: Permite visualizar el resultado (1) o no (0)
10 %   Parámetros de SALIDA:
11 %       - a: Coeficientes del modelo
12 %       - error: Error cometido en la estimación de la señal de salida
13 %       - y_est: Señal de salida estimada con el modelo calculado
14
15
16 function [a, error,y_est ] = trainPM(sigin,sigout,orden,vis )
17     % Ponemos la señal de entrada en formato matricial. La matriz tendrá k
18     % columnas, y en cada una de ellas tendremos sigin.*abs(sigin).^k
19     ex = 0:orden-1;
20     X = sigin.*abs(sigin).^ex;
21
22     % Calculamos los coeficientes del modelo
23     a = pinv(X)*sigout;
24     y_est = X*a;
25
26     % Visualizamos el resultado del modelo
27     if (vis)
28         figure, scatter(abs(sigin),abs(sigout));
29         hold on, scatter(abs(sigin),abs(y_est),'k');
30     end
31
32     % Calculamos el error cometido
33     error = 10*log10(mean(abs(sigout-y_est).^2)/mean(abs(sigout).^2));
34
35 end

```

14. Función: DLSimulator.m

```

1  close all,
2  dlsimulator();
3
4  function dlsimulator ()
5
6      % Ajuste del path, declaracion de variables y de la interfaz
7      ...
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

```

400     % el simulador
401     ...

447 function menCan_Callback(src,event)
448     % Menu de ayuda acerca de los modelos de canal disponibles para simular
449     % una transmision real completa
450     ...

537 function menRepr_Callback(src,event)
538     % Menu con la informacion acerca de las representaciones y resultados
539     % ofrecidos al usuario
540     ...

612 function Modes_Callback(src,event)
613     % Menu que recoge la ayuda sobre los modos de funcionamiento del simulador
614     % y los objetivos de cada uno
615     ...

653 function NoLin_Callback(src,event)
654     % Menu que recoge la ayuda sobre el modelado de los elementos no lineales
655     ...

687 function CloseMen_Callback(source,fig)
688     % Funcion para cerrar las ventanas de los menus
689     ...

693 function Preferencias_Callback(src,event)
694     % Ventana que permite seleccionar el modo de trabajo del simulador
695     ...

731 function trainSignals_Callback(src,event)
732     % Ventana que permite cambiar las señales utilizadas para el modelado
733     % de los elementos no lineales
734     ...

800 % Definir configuracion por defecto del transmisor
801 ...

849 function nFrames_Callback(source,eventdata)
850     % Funcion que gestiona el cambio en el numero de tramas
851     ...

861 function nBlocks_Callback(source,eventdata)
862     % Funcion que gestiona cambios en la configuracion de ancho de banda del
863     % sistema
864     ...

887 function nBlocksUsed_Callback (source,eventdata)
888     % Funcion que gestiona los cambios en el numero de bloques usados, en el modo
889     % de configuracion no avanzado
890     ...

913 function nBlocksUsedAdv_Callback (source,eventdata)
914     % Funcion que gestiona cambios en el numero de bloques usados en el modo
915     % avanzado
916     ...

1017 function cpLength_Callback (source,eventdata)
1018     % Funcion que gestiona los cambios en el prefijo ciclico
1019     ...

1030 function Modulation_Callback(source,eventdata)
1031     % Funcion que permite los cambios en la modulacion utilizada para el canal de
1032     % datos
1033     ...

1042 function PilotsPow_Callback(source,eventdata)
1043     % Funcion que gestiona cambios en la potencia de las señales piloto, en el modo
1044     % de configuracion no avanzado

```

```

1045     ...

1057     function PilotsPowAdv_Callback(source,eventdata)
1058         % Funcion que gestiona cambios en la potencia de las señales piloto, en el modo
1059         % de configuracion avanzado
1060     ...

1150     function PRBsPowAdv_Callback(source,eventdata)
1151         % Funcion que permite ajustar la potencia de los PRBs transmitidos
1152     ...

1260     function NB_Iot_Callback(source,eventdata)
1261         % Funcion para generar e integrar las señales NB-IoT
1262     ...

1617     function NB_Iot_Del_Callback(source,eventdata)
1618         % Funcion para borrar las señales NB-IoT generadas
1619     ...

1624     function ModAmpli_Callback(source,eventdata,opt)
1625         % Funcion que gestiona la eleccion del modelo de amplificador
1626     ...

1638     function EditMem_Callback(source,eventdata)
1639         % Funcion que permite ajustar la memoria en el modelo de amplificador
1640         % con memoria
1641     ...

1646     function ModMem_Callback(source,eventdata,opt)
1647         % Funcion que gestiona los botones que permiten ajustar la memoria del
1648         % modelo de amplificador con memoria
1649     ...

1655     function Generate_Callback(source,eventdata)
1656         % Funcion que permite generar la señal 5G
1657     ...

2043     function TxResults_Callback(source,eventdata,opt)
2044         % Funcion que gestiona las graficas que se desea dibujar
2045     ...

2053     function TxResultsPlot_Callback(source,eventdata)
2054         % Funcion que realiza los plots seleccionados por el usuario
2055     ...

2392     function TimeDomExport_Callback(source,eventdata)
2393         % Funcion que permite exportar señales generadas
2394     ...

2541     function Channel_Callback(source,eventdata)
2542         % Funcion que gestiona la eleccion del modelo de canal
2543     ...

2798     function Demodulate_Callback(source,eventdata)
2799         % Funcion que realiza la demodulacion de la señal 5G recibida en
2800         % el receptor
2801     ...

2954     function Constellation_Callback(source,eventdata)
2955         % Funcion que calcula y representa la constelacion del canal de datos
2956     ...

3004     function EVM_Callback(source,eventdata)
3005         % Funcion que realiza todos los calculos pertinentes y muestra las graficas
3006         % de EVM
3007     ...

3148 end

```

15. Función: DL_5GNR_Simulator.m

```

1  close all,
2  DL5GNR_Simulator();
3
4  function DL5GNR_Simulator()
5
6      % Ajuste del path, declaracion de variables y de la interfaz
7      ...
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

```

680     % Funcion que permite elegir entre los dos modos de duplexado permitidos
681     ...

685     function ModAmpli_Callback(src,eventdata,opt)
686         % Funcion que permite que el usuario elija el modelo de amplificador deseado
687         ...

750     function ModChan_Callback(src,eventdata,opt)
751         % Función que gestiona los ajustes en el modelo de canal
752         ...

860     function Simulate_Callback(src,eventdata)
861         % Funcion que realiza la simulacion completa transmision-recepcion de la
862         % señal 5G-NR
863         ...

1221     function Plot_Callback(src,eventdata)
1222         % Funcion que permite realizar las representaciones seleccionadas por el
1223         % usuario
1224         ...

1512     function Export_Callback(src,eventdata)
1513         % Funcion que permite exportar señales generadas
1514         ...

1630
1631 end

```


Anexo D

Hojas de características

Se incluye la hoja de características del amplificador que se ha utilizado en la captura de señales reales y que posteriormente se ha modelado para obtener las medidas recogidas en este trabajo:

- Amplificador de potencia (PA): Modelo ZHL-4240+, Mini-Circuits

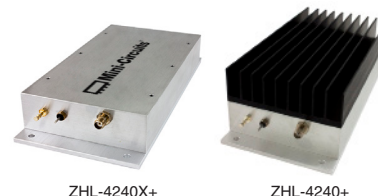
Coaxial Amplifier

ZHL-4240+

50Ω Medium High Power 600 to 4200 MHz

The Big Deal

- Wideband, 600 to 4200 MHz
- High gain, 39 dB
- Extremely flat gain, ± 1.3 dB
- High IP3, +38 dBm



CASE STYLE: U36

Product Overview

Mini-Circuits' ZHL-4240+ is a medium-power connectorized amplifier supporting a wide range of applications from 600 to 4200 MHz, such as test instrumentation, SatCom, and mobile communications systems, including those operating in the new telecom Band 71 allocation (617 to 698 MHz). This model provides +31 dBm output power at saturation and extremely flat gain (39 ± 1.3 dB) across its full bandwidth, making it ideal for systems where consistent performance across frequency is required. The amplifier operates on a 15V DC supply and comes housed in compact aluminum alloy case (7.00 x 3.25 x 2.13") with SMA connectors, built-in bracket for mounting, and an optional heat sink for efficient cooling.

Key Features

Feature	Advantages
Wideband, 600 to 4200 MHz	One amplifier supports a broad range of system and test lab applications. Extended bandwidth down to 600 MHz supports new telecom Band 71 allocation (617 to 698 MHz)
High gain, 39 dB	Reduces the number of gain stages, lowering component count and overall system cost.
Excellent gain flatness, ± 1.3 dB	Provides consistent performance across frequency, minimizing the need for external equalizing networks in wideband applications.
Medium output power, +31 dBm P3dB	Supports a wide range of power requirements.
High OIP3, +38 dBm	Provides highly linear performance with excellent sensitivity and two-tone spur-free dynamic range.

Notes

A. Performance and quality attributes and conditions not expressly stated in this specification document are intended to be excluded and do not form a part of this specification document.

B. Electrical specifications and performance data contained in this specification document are based on Mini-Circuit's applicable established test performance criteria and measurement instructions.

C. The parts covered by this specification document are subject to Mini-Circuits standard limited warranty and terms and conditions (collectively, "Standard Terms"); Purchasers of this part are entitled to the rights and benefits contained therein. For a full statement of the Standard Terms and the exclusive rights and remedies thereunder, please visit Mini-Circuits' website at www.minicircuits.com/MCLStore/terms.jsp



www.minicircuits.com P.O. Box 350166, Brooklyn, NY 11235-0003 (718) 934-4500 sales@minicircuits.com

Coaxial Amplifier

ZHL-4240+

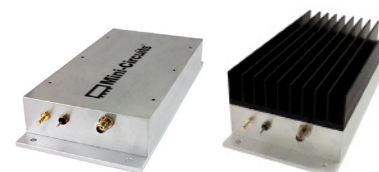
50Ω Medium High Power 600 to 4200 MHz

Features

- wideband, 600 to 4200 MHz
- high IP3, +38 dBm typ.
- high gain, 39 dB min.

Applications

- communication systems
- satellite dist./GPS/PCS
- instrumentation
- laboratory



ZHL-4240X+

ZHL-4240+

CASE STYLE: U36

Connectors	Model
SMA	ZHL-4240+
SMA	ZHL-4240X+

+RoHS Compliant

The +Suffix identifies RoHS Compliance. See our web site for RoHS Compliance methodologies and qualifications

Electrical Specifications at 25°C

Parameter	Condition (MHz)	ZHL-4240+ ▲ ZHL-4240X+			Units
		Min.	Typ.	Max.	
Frequency Range		600	—	4200	MHz
Gain	600-4200	39	42	47	dB
Gain Flatness	600-4200	—	±1.3	±1.8	dB
Output Power at 1dB compression*	600-4200	+28	+30	—	dBm
Output Power at 3dB compression**	600-4200	+29	+31	—	dBm
Noise Figure	600-4200	—	8.0	—	dB
Output third order intercept point	600-4200	—	+38	—	dBm
Input VSWR	600-4200	—	—	2.5	:1
Output VSWR	600-4200	—	—	2.5	:1
DC Supply Voltage		—	15	—	V
Supply Current		—	—	1.0	A

Open load is not recommended, potentially can cause damage.
With no load derate max. input power by 20 dB.

* +27 dBm at 3700-4200 MHz

** +28 dBm at 3700-4200 MHz

▲Heat sink not included. Alternative heat sinking and heat removal must be provided by the user to limit maximum base-plate temperature to 65°C, in order to ensure proper performance. For reference, this requires thermal resistance of user's external heat sink to be 1.3°C/W max.

Maximum Ratings

Parameter	Ratings
Operating Temperature	-20°C to 65°C
Storage Temperature	-55°C to 100°C
DC Voltage	+20V
Input RF Power (no damage)	-5 dBm

Permanent damage may occur if any of these limits are exceeded.

Notes

A. Performance and quality attributes and conditions not expressly stated in this specification document are intended to be excluded and do not form a part of this specification document.

B. Electrical specifications and performance data contained in this specification document are based on Mini-Circuit's applicable established test performance criteria and measurement instructions.

C. The parts covered by this specification document are subject to Mini-Circuits standard limited warranty and terms and conditions (collectively, "Standard Terms"); Purchasers of this part are entitled to the rights and benefits contained therein. For a full statement of the Standard Terms and the exclusive rights and remedies thereunder, please visit Mini-Circuits' website at www.minicircuits.com/MCLStore/terms.jsp

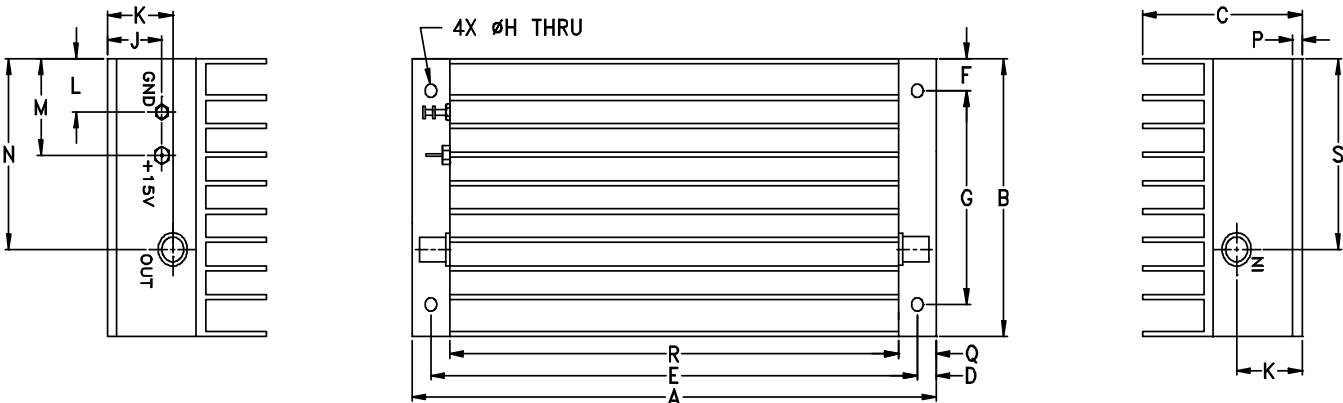


www.minicircuits.com P.O. Box 350166, Brooklyn, NY 11235-0003 (718) 934-4500 sales@minicircuits.com

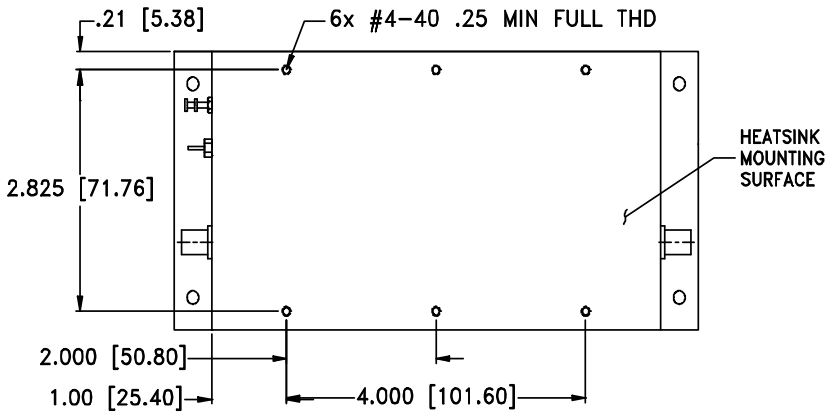
REV. A
M167682
ZHL-4240+
CH/CP/AM
180507

Page 2 of 4

Outline Drawing for models with heatsink



MOUNTING INFORMATION FOR MODELS WITHOUT HEATSINK



Outline Dimensions (^{inch}_{mm})

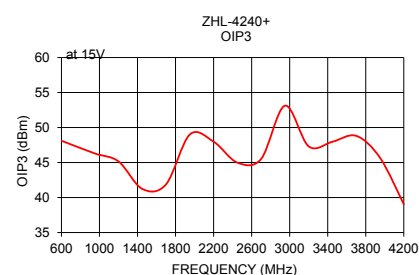
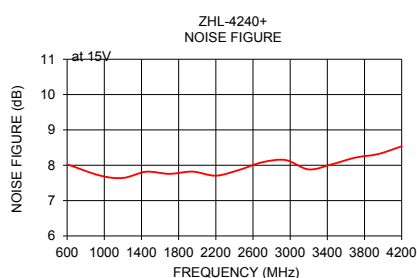
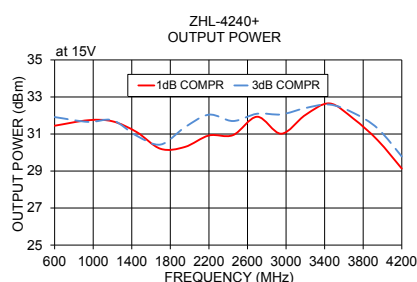
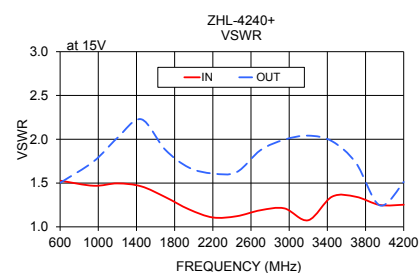
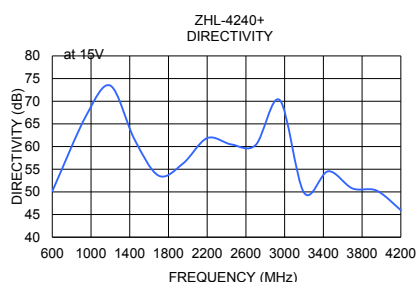
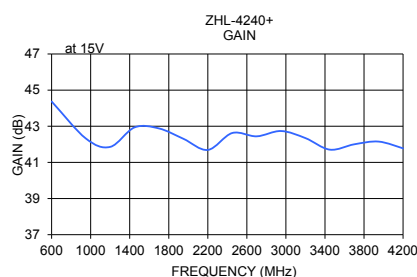
A	B	C	D	E	F	G	H	J	K	L	M	N	P	Q	R	S	wt
7.00	3.25	2.13	.25	6.500	.38	2.500	.156	.73	.88	.63	1.13	2.23	.125	.50	6.00	2.23	grams
177.80	82.55	54.10	6.35	165.10	9.65	63.50	3.96	18.54	22.35	16.00	28.70	56.64	3.18	12.70	152.40	56.64	900

*600 grams without heatsink

Notes
A. Performance and quality attributes and conditions not expressly stated in this specification document are intended to be excluded and do not form a part of this specification document.
B. Electrical specifications and performance data contained in this specification document are based on Mini-Circuits' applicable established test performance criteria and measurement instructions.
C. The parts covered by this specification document are subject to Mini-Circuits standard limited warranty and terms and conditions (collectively, "Standard Terms"); Purchasers of this part are entitled to the rights and benefits contained therein. For a full statement of the Standard Terms and the exclusive rights and remedies thereunder, please visit Mini-Circuits' website at www.minicircuits.com/MCLStore/terms.jsp



FREQUENCY (MHz)	GAIN (dB)	DIRECTIVITY (dB)	VSWR (:1)		POUT at 1 dB COMPR. (dBm)	POUT at 3 dB COMPR. (dBm)	NOISE FIGURE (dB)	IP3 (dBm)
	15V	15V	IN	OUT	15V	15V	15V	15V
600	44.37	50.13	1.53	1.50	31.45	31.91	8.03	48.11
950	42.33	66.82	1.47	1.75	31.74	31.65	7.71	46.30
1200	41.86	73.44	1.50	2.02	31.68	31.74	7.64	45.16
1450	42.94	61.55	1.46	2.23	31.11	30.91	7.82	41.23
1700	42.88	53.56	1.34	1.88	30.20	30.43	7.76	41.89
1950	42.31	56.25	1.20	1.68	30.30	31.36	7.82	49.01
2200	41.69	61.87	1.11	1.61	30.92	32.04	7.71	47.98
2450	42.63	60.47	1.12	1.62	30.94	31.70	7.87	44.99
2700	42.44	60.33	1.19	1.88	31.93	32.09	8.08	45.51
2950	42.74	70.24	1.21	2.00	31.01	32.06	8.15	53.12
3200	42.34	49.82	1.08	2.04	32.03	32.40	7.88	47.31
3450	41.71	54.55	1.34	1.98	32.65	32.59	8.03	47.97
3700	42.00	50.76	1.34	1.73	31.82	32.14	8.22	48.81
3950	42.15	50.28	1.25	1.24	30.66	31.30	8.32	45.68
4200	41.78	45.95	1.25	1.51	29.13	29.78	8.54	39.05
4200	41.78	45.95	1.25	1.51	29.13	29.78	8.54	39.05



Notes

- Performance and quality attributes and conditions not expressly stated in this specification document are intended to be excluded and do not form a part of this specification document.
- Electrical specifications and performance data contained in this specification document are based on Mini-Circuits' applicable established test performance criteria and measurement instructions.
- The parts covered by this specification document are subject to Mini-Circuits standard limited warranty and terms and conditions (collectively, "Standard Terms"); Purchasers of this part are entitled to the rights and benefits contained therein. For a full statement of the Standard Terms and the exclusive rights and remedies thereunder, please visit Mini-Circuits' website at www.minicircuits.com/MCLStore/terms.jsp



